TEXAS INSTRUMENTS

PCI4410 GHK/PDV

PC Card and OHCI Controller

Data Manual

NOTE **Designing with this device may require extensive support. Before incorporating this device into a design, customers should contact TI or an Authorized TI Distributor.**

2000 *PCIBus Solutions*

TEXAS INSTRUMENTS

Printed in U.S.A., 01/00 SCPS052

PCI4410 GHK/PDV Data Manual

PC Card and OHCI Controller

Literature Number: SCPS052
January 2000

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

Contents

List of Illustrations

List of Tables

1 Introduction

The Texas Instruments PCI4410 is an integrated single-socket PC Card controller and IEEE 1394 Open HCI host controller. This high-performance integrated solution provides the latest in both PC Card and IEEE 1394 technology.

1.1 Description

The PCI4410 is a dual-function PCI device compliant with *PCI Local Bus Specification 2.2*. Function 0 provides the independent PC Card socket controller compliant with the *1997 PC Card Standard*. The PCI4410 provides features that make it the best choice for bridging between the PCI bus and PC Cards, and supports either 16-bit or CardBus PC Cards in the socket, powered at 5 V or 3.3 V, as required.

All card signals are internally buffered to allow hot insertion and removal without external buffering. The PCI4410 is register compatible with the Intel™ 82365SL−DF and 82365SL ExCA controllers. The PCI4410 internal data path logic allows the host to access 8-, 16-, and 32-bit cards using full 32-bit PCI cycles for maximum performance. Independent buffering and a pipeline architecture provide an unsurpassed performance level with sustained bursting. The PCI4410 can be programmed to accept posted writes to improve bus utilization.

Function 1 of the PCI4410 is compatible with IEEE1394A and the latest 1394 open host controller interface (OHCI) specifications. The chip provides the IEEE1394 link function and is compatible with data rates of 100, 200, and 400 Mbits per second. Deep FIFOs are provided to buffer 1394 data and accommodate large host bus latencies. The PCI4410 provides physical write posting and a highly tuned physical data path for SBP-2 performance. Multiple cache line burst transfers, advanced internal arbitration, and bus holding buffers on the PHY/Link interface are other features that make the PCI4410 the best-in-class 1394 Open HCI solution.

The PCI4410 provides an internally buffered zoomed video (ZV) path. This reduces the design effort of PC board manufacturers to add a ZV-compatible solution and ensures compliance with the CardBus loading specifications.

Various implementation-specific functions and general-purpose inputs and outputs are provided through eight multifunction terminals. These terminals present a system with options in PC/PCI DMA, PCI $\overline{\text{LOCK}}$ and parallel interrupts, PC Card activity indicator LEDs, and other platform-specific signals. ACPI-compliant general-purpose events may be programmed and controlled through the multifunction terminals, and an ACPI-compliant programming interface is included for the general-purpose inputs and outputs.

The PCI4410 is compliant with the latest *PCI Bus Power Management Specification*, and provides several low-power modes which enable the host power system to further reduce power consumption. The *PC Card (CardBus) Controller* and *IEEE 1394 Host Controller Device Class Specifications* required for Microsoft OnNow™ power management are supported. Furthermore, an advanced complementary metal-oxide semiconductor (CMOS) process achieves low system power consumption.

Unused PCI4410 inputs must be pulled to a valid logic level using a 43-kΩ resistor.

1.2 Features

The PCI4410 supports the following features:

- Ability to wake from $D3_{hot}$ and $D3_{cold}$
- Fully compatible with the Intel 430TX (Mobile Triton II) chipset
- A 208-pin low-profile QFP (PDV) or 209-ball MICROSTAR BGA™ ball grid array (GHK) package

Intel is a trademark of Intel Corporation.
Microsoft OnNow is a trademark of Microsoft Corporation.
MicroStar BGA is a trademark of Texas Instruments Incorporated

- 3.3-V core logic with universal PCI interfaces compatible with 3.3-V and 5-V PCI signaling environments
- Mix-and-match 5-V/3.3-V 16-bit PC Cards and 3.3-V CardBus Cards
- Single PC Card or CardBus slot with hot insertion and removal
- Burst transfers to maximize data throughput on the PCI bus and the CardBus bus
- Parallel PCI interrupts, parallel ISA IRQ and parallel PCI interrupts, serial ISA IRQ with parallel PCI interrupts, and serial ISA IRQ and PCI interrupts
- Serial EEPROM interface for loading subsystem ID and subsystem vendor ID
- Pipelined architecture allows greater than 130M bps sustained throughput from CardBus-to-PCI and from PCI-to-CardBus
- Interface to parallel single-slot PC Card power interface switches like the TI™ TPS2211
- Up to five general-purpose I/Os
- Programmable output select for $\overline{\text{CLKRUN}}$
- Five PCI memory windows and two I/O windows available to the 16-bit PC Card socket
- Two I/O windows and two memory windows available to the CardBus socket
- Exchangeable Card Architecture (ExCA) compatible registers are mapped in memory and I/O space
- Intel 82365SL-DF and 82365SL register compatible
- Distributed DMA (DDMA) and PC/PCI DMA
- 16-Bit DMA on the PC Card socket
- Ring indicate, $\overline{\text{SUSPEND}}$, PCI $\overline{\text{CLKRUN}}$, and CardBus $\overline{\text{CLKRUN}}$
- Socket activity LED pins
- PCI bus lock ($\overline{\text{LOCK}}$)
- Advanced submicron, low-power CMOS technology
- Internal ring oscillator
- OHCI link function designed to *IEEE 1394 Open Host Controller Interface (OHCI) Specification*
- Implements PCI burst transfers and deep FIFOs to tolerate large host latency
- Supports physical write posting of up to 3 outstanding transactions
- OHCI link function is IEEE 1394-1995 compliant and compatible with Proposal 1394a
- Supports serial bus data rates of 100, 200, and 400 Mbits/second
- Provides bus-hold buffers on the PHY-Link I/F for low-cost single-capacitor isolation

1.3 Related Documents

- *Advanced Configuration and Power Interface (ACPI) Specification* (Revision 2.0)
- *PCI Bus Power Management Interface Specification* (Revision 1.1)
- *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges* (Revision 1.)
- *PCI Local Bus Specification* (Revision 2.2)
- *PCI Mobile Design Guide* (Revision 1.0)
- *PCI14xx Implemenation Guide for D3 Wake-Up*
- *1997 PC Card Standard*
- *PC 98/99*
- *Serialized IRQ Support for PCI Systems* (Revision 6)

1.4 Ordering Information

ORDERING NUMBER	NAME	VOLTAGE	PACKAGE
PCI4410	PC Card controller	3.3-V, 5-V tolerant I/Os	208-pin LQFP 209-ball PBGA

2 Terminal Descriptions

The PCI4410 is packaged in either a 209-ball GHK MICROSTAR BGA or a 208-terminal PDV package. The PCI4410 is a single-socket CardBus bridge with integrated OHCI link. Figure 2−1 is a terminal diagram of the PDV package with PCI-to-CardBus signal names. Figure 2−2 is a terminal diagram of the PDV package with PCI-to-PC Card signal names. Figure 2−3 is a terminal diagram of the GHK package.

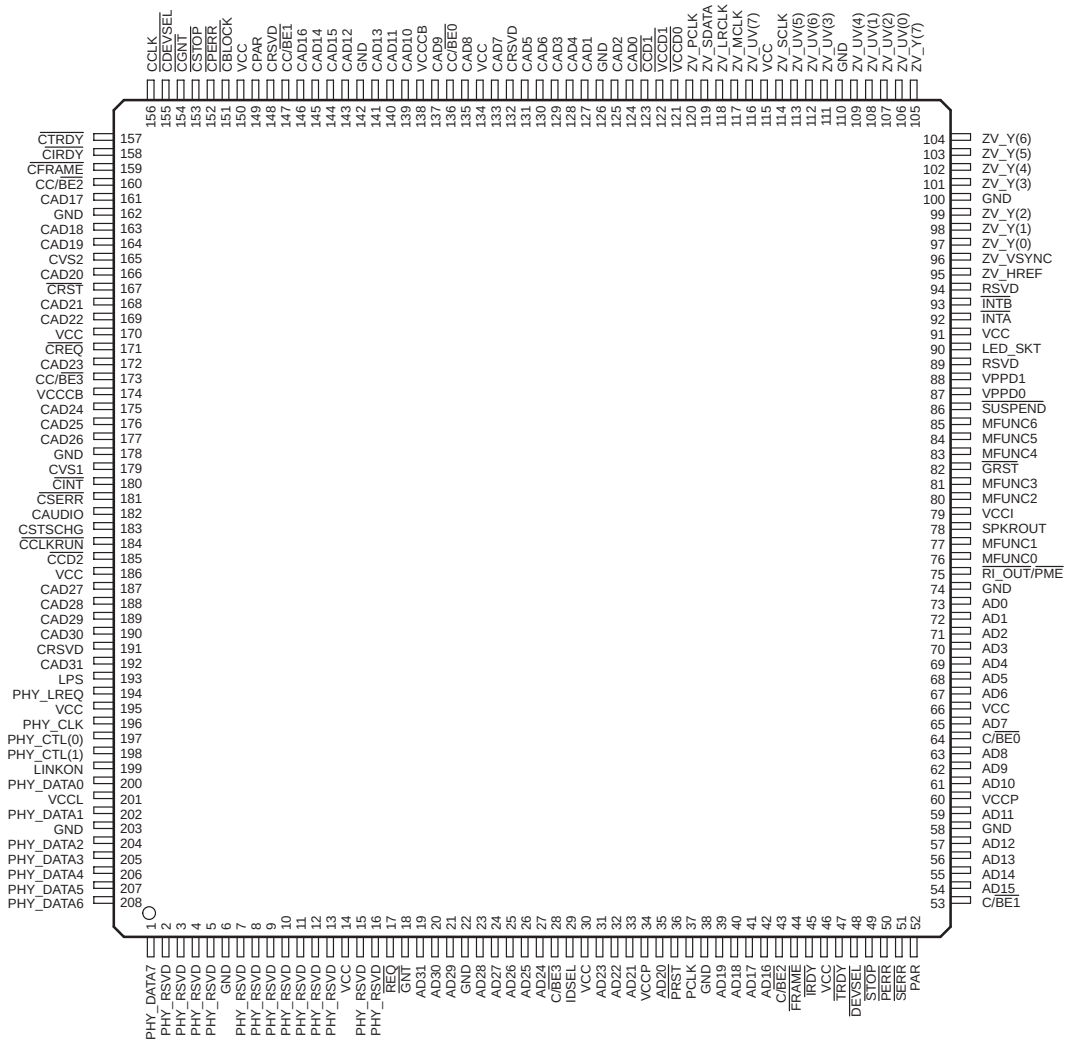


Figure 2−1. PCI-to-CardBus Terminal Diagram

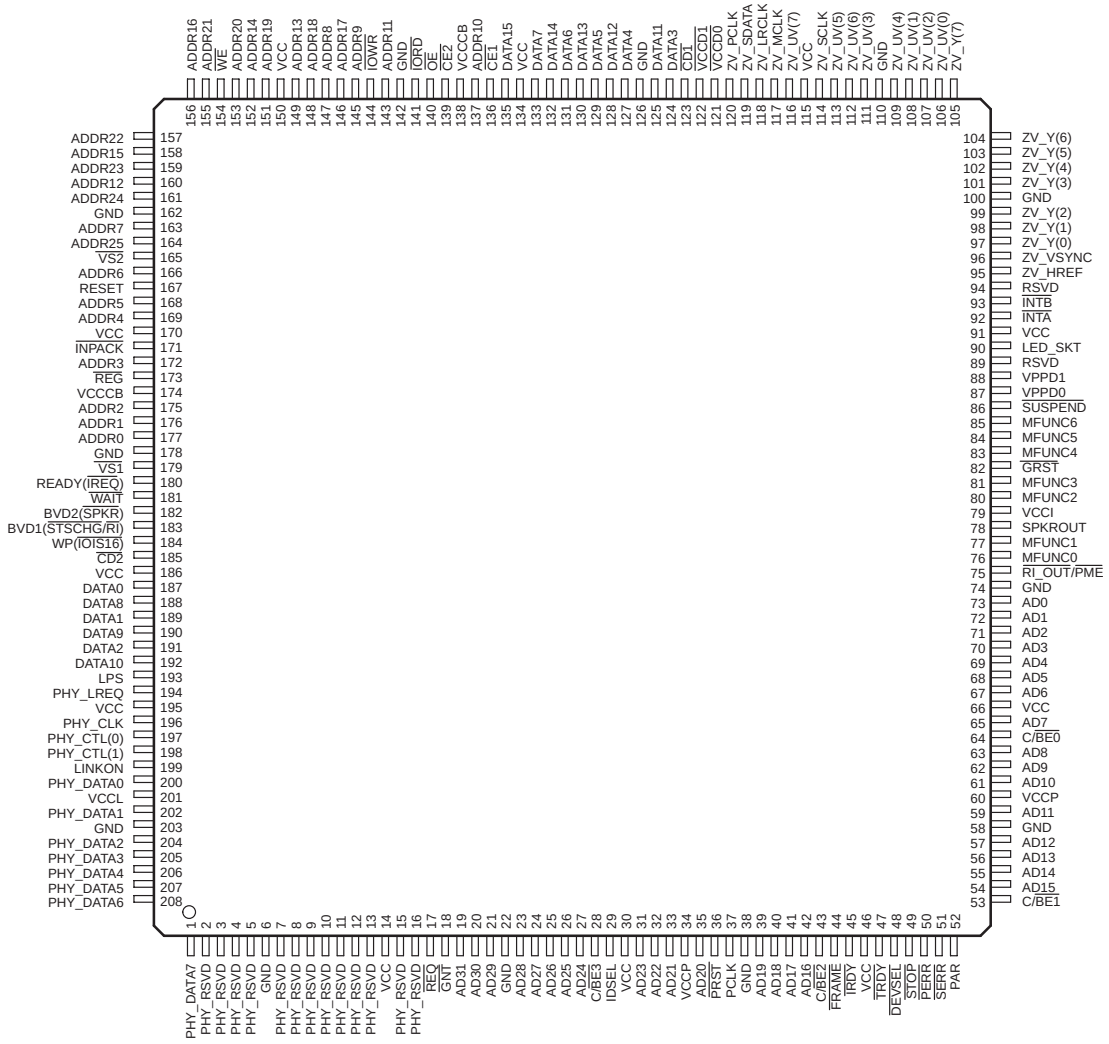


Figure 2−2. PCI-to-PC Card (16-Bit) Terminal Diagram

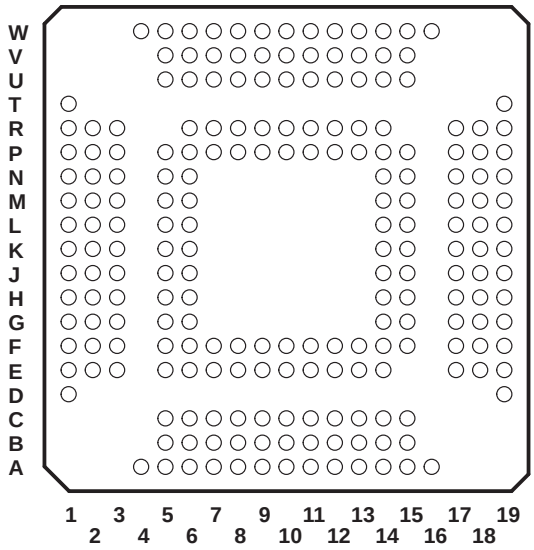


Figure 2−3. MICROSTAR BGA Ball Diagram

Table 2−1 shows the terminal assignments for the 208-terminal PDV CardBus and 16-bit PC Card signal names. Table 2−2 shows the terminal assignments for the 209-ball GHK CardBus and 16-bit PC Card signal names. Table 2−3 shows the CardBus PC Card signal names sorted alphabetically to the GHK/PDV terminal numbers. Table 2−4 shows the 16-bit PC Card signal names sorted alphabetically to the GHK/PDV terminal numbers.

Table 2−1. CardBus and 16-Bit PC Card Signal Names by PDV Terminal Number

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CARDBUS	16-BIT		CARDBUS	16-BIT		CARDBUS	16-BIT
1	PHY_DATA7	PHY_DATA7	44	$\overline{\text{FRAME}}$	$\overline{\text{FRAME}}$	87	VPPD0	VPPD0
2	PHY_RSVD	PHY_RSVD	45	$\overline{\text{IRDY}}$	$\overline{\text{IRDY}}$	88	VPPD1	VPPD1
3	PHY_RSVD	PHY_RSVD	46	V_{CC}	V_{CC}	89	RSVD	RSVD
4	PHY_RSVD	PHY_RSVD	47	$\overline{\text{TRDY}}$	$\overline{\text{TRDY}}$	90	LED_SKT	LED_SKT
5	PHY_RSVD	PHY_RSVD	48	$\overline{\text{DEVSEL}}$	$\overline{\text{DEVSEL}}$	91	V_{CC}	V_{CC}
6	GND	GND	49	$\overline{\text{STOP}}$	$\overline{\text{STOP}}$	92	$\overline{\text{INTA}}$	$\overline{\text{INTA}}$
7	PHY_RSVD	PHY_RSVD	50	$\overline{\text{PERR}}$	$\overline{\text{PERR}}$	93	$\overline{\text{INTB}}$	$\overline{\text{INTB}}$
8	PHY_RSVD	PHY_RSVD	51	$\overline{\text{SERR}}$	$\overline{\text{SERR}}$	94	RSVD	RSVD
9	PHY_RSVD	PHY_RSVD	52	PAR	PAR	95	ZV_HREF	ZV_HREF
10	PHY_RSVD	PHY_RSVD	53	C/$\overline{\text{BE1}}$	C/$\overline{\text{BE1}}$	96	ZV_VSYNC	ZV_VSYNC
11	PHY_RSVD	PHY_RSVD	54	AD15	AD15	97	ZV_Y(0)	ZV_Y(0)
12	PHY_RSVD	PHY_RSVD	55	AD14	AD14	98	ZV_Y(1)	ZV_Y(1)
13	PHY_RSVD	PHY_RSVD	56	AD13	AD13	99	ZV_Y(2)	ZV_Y(2)
14	V_{CC}	V_{CC}	57	AD12	AD12	100	GND	GND
15	PHY_RSVD	PHY_RSVD	58	GND	GND	101	ZV_Y(3)	ZV_Y(3)
16	PHY_RSVD	PHY_RSVD	59	AD11	AD11	102	ZV_Y(4)	ZV_Y(4)
17	$\overline{\text{REQ}}$	$\overline{\text{REQ}}$	60	V_{CCP}	V_{CCP}	103	ZV_Y(5)	ZV_Y(5)
18	$\overline{\text{GNT}}$	$\overline{\text{GNT}}$	61	AD10	AD10	104	ZV_Y(6)	ZV_Y(6)
19	AD31	AD31	62	AD9	AD9	105	ZV_Y(7)	ZV_Y(7)
20	AD30	AD30	63	AD8	AD8	106	ZV_UV(0)	ZV_UV(0)
21	AD29	AD29	64	C/$\overline{\text{BE0}}$	C/$\overline{\text{BE0}}$	107	ZV_UV(2)	ZV_UV(2)
22	GND	GND	65	AD7	AD7	108	ZV_UV(1)	ZV_UV(1)
23	AD28	AD28	66	V_{CC}	V_{CC}	109	ZV_UV(4)	ZV_UV(4)
24	AD27	AD27	67	AD6	AD6	110	GND	GND
25	AD26	AD26	68	AD5	AD5	111	ZV_UV(3)	ZV_UV(3)
26	AD25	AD25	69	AD4	AD4	112	ZV_UV(6)	ZV_UV(6)
27	AD24	AD24	70	AD3	AD3	113	ZV_UV(5)	ZV_UV(5)
28	C/$\overline{\text{BE3}}$	C/$\overline{\text{BE3}}$	71	AD2	AD2	114	ZV_SCLK	ZV_SCLK
29	IDSEL	IDSEL	72	AD1	AD1	115	V_{CC}	V_{CC}
30	V_{CC}	V_{CC}	73	AD0	AD0	116	ZV_UV(7)	ZV_UV(7)
31	AD23	AD23	74	GND	GND	117	ZV_MCLK	ZV_MCLK
32	AD22	AD22	75	$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	118	ZV_LRCLK	ZV_LRCLK
33	AD21	AD21	76	MFUNC0	MFUNC0	119	ZV_SDATA	ZV_SDATA
34	V_{CCP}	V_{CCP}	77	MFUNC1	MFUNC1	120	ZV_PCLK	ZV_PCLK
35	AD20	AD20	78	SPKROUT	SPKROUT	121	$\overline{\text{VCCD0}}$	$\overline{\text{VCCD0}}$
36	$\overline{\text{PRST}}$	$\overline{\text{PRST}}$	79	V_{CCI}	V_{CCI}	122	$\overline{\text{VCCD1}}$	$\overline{\text{VCCD1}}$
37	PCLK	PCLK	80	MFUNC2	MFUNC2	123	$\overline{\text{CCD1}}$	$\overline{\text{CD1}}$
38	GND	GND	81	MFUNC3	MFUNC3	124	CAD0	DATA3
39	AD19	AD19	82	$\overline{\text{GRST}}$	$\overline{\text{GRST}}$	125	CAD2	DATA11
40	AD18	AD18	83	MFUNC4	MFUNC4	126	GND	GND
41	AD17	AD17	84	MFUNC5	MFUNC5	127	CAD1	DATA4
42	AD16	AD16	85	MFUNC6	MFUNC6	128	CAD4	DATA12
43	C/$\overline{\text{BE2}}$	C/$\overline{\text{BE2}}$	86	$\overline{\text{SUSPEND}}$	$\overline{\text{SUSPEND}}$	129	CAD3	DATA5

Table 2−1. CardBus and 16-Bit PC Card Signal Names by PDV Terminal Number (Continued)

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CARDBUS	16-BIT		CARDBUS	16-BIT		CARDBUS	16-BIT
130	CAD6	DATA13	157	$\overline{\text{CTRDY}}$	ADDR22	184	$\overline{\text{CCLKRUN}}$	WP($\overline{\text{IOIS16}}$)
131	CAD5	DATA6	158	$\overline{\text{CIRDY}}$	ADDR15	185	$\overline{\text{CCD2}}$	$\overline{\text{CD2}}$
132	CRSVD	DATA14	159	$\overline{\text{CFRAME}}$	ADDR23	186	V_{CC}	V_{CC}
133	CAD7	DATA7	160	CC/$\overline{\text{BE2}}$	ADDR12	187	CAD27	DATA0
134	V_{CC}	V_{CC}	161	CAD17	ADDR24	188	CAD28	DATA8
135	CAD8	DATA15	162	GND	GND	189	CAD29	DATA1
136	CC/$\overline{\text{BE0}}$	$\overline{\text{CE1}}$	163	CAD18	ADDR7	190	CAD30	DATA9
137	CAD9	ADDR10	164	CAD19	ADDR25	191	CRSVD	DATA2
138	V_{CCCB}	V_{CCCB}	165	CVS2	$\overline{\text{VS2}}$	192	CAD31	DATA10
139	CAD10	$\overline{\text{CE2}}$	166	CAD20	ADDR6	193	LPS	LPS
140	CAD11	$\overline{\text{OE}}$	167	$\overline{\text{CRST}}$	RESET	194	PHY_LREQ	PHY_LREQ
141	CAD13	$\overline{\text{IORD}}$	168	CAD21	ADDR5	195	V_{CC}	V_{CC}
142	GND	GND	169	CAD22	ADDR4	196	PHY_CLK	PHY_CLK
143	CAD12	ADDR11	170	V_{CC}	V_{CC}	197	PHY_CTL(0)	PHY_CTL(0)
144	CAD15	$\overline{\text{IOWR}}$	171	$\overline{\text{CREQ}}$	$\overline{\text{INPACK}}$	198	PHY_CTL(1)	PHY_CTL(1)
145	CAD14	ADDR9	172	CAD23	ADDR3	199	LINKON	LINKON
146	CAD16	ADDR17	173	CC/$\overline{\text{BE3}}$	$\overline{\text{REG}}$	200	PHY_DATA0	PHY_DATA0
147	CC/$\overline{\text{BE1}}$	ADDR8	174	V_{CCCB}	V_{CCCB}	201	V_{CCL}	V_{CCL}
148	CRSVD	ADDR18	175	CAD24	ADDR2	202	PHY_DATA1	PHY_DATA1
149	CPAR	ADDR13	176	CAD25	ADDR1	203	GND	GND
150	V_{CC}	V_{CC}	177	CAD26	ADDR0	204	PHY_DATA2	PHY_DATA2
151	$\overline{\text{CBLOCK}}$	ADDR19	178	GND	GND	205	PHY_DATA3	PHY_DATA3
152	$\overline{\text{CPERR}}$	ADDR14	179	CVS1	$\overline{\text{VS1}}$	206	PHY_DATA4	PHY_DATA4
153	$\overline{\text{CSTOP}}$	ADDR20	180	$\overline{\text{CINT}}$	READY($\overline{\text{IREQ}}$)	207	PHY_DATA5	PHY_DATA5
154	$\overline{\text{CGNT}}$	$\overline{\text{WE}}$	181	$\overline{\text{CSERR}}$	$\overline{\text{WAIT}}$	208	PHY_DATA6	PHY_DATA6
155	$\overline{\text{CDEVSEL}}$	ADDR21	182	CAUDIO	BVD2($\overline{\text{SPKR}}$)			
156	CCLK	ADDR16	183	CSTSCHG	BVD1 ($\overline{\text{STSCHG}}$/$\overline{\text{RI}}$)			

Table 2−2. CardBus and 16-Bit PC Card Signal Names by GHK Terminal Number

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CARDBUS	16-BIT		CARDBUS	16-BIT		CARDBUS	16-BIT
A4	PHY_DATA6	PHY_DATA6	E8	PHY_LREQ	PHY_LREQ	H14	CAD13	$\overline{\text{IORD}}$
A5	GND	GND	E9	CAD29	DATA1	H15	GND	GND
A6	LINKON	LINKON	E10	CSTSCHG	BVD1 ($\overline{\text{STSCHG}}$/$\overline{\text{RI}}$)	H17	CAD11	$\overline{\text{OE}}$
A7	V_{CC}	V_{CC}	E11	GND	GND	H18	CAD10	$\overline{\text{CE2}}$
A8	CAD30	DATA9	E12	$\overline{\text{CREQ}}$	$\overline{\text{INPACK}}$	H19	V_{CCCB}	V_{CCCB}
A9	$\overline{\text{CCD2}}$	$\overline{\text{CD2}}$	E13	CVS2	$\overline{\text{VS2}}$	J1	AD31	AD31
A10	$\overline{\text{CINT}}$	READY($\overline{\text{IREQ}}$)	E14	$\overline{\text{CFRAME}}$	ADDR23	J2	AD30	AD30
A11	CAD24	ADDR2	E17	$\overline{\text{CDEVSEL}}$	ADDR21	J3	AD29	AD29
A12	V_{CCCB}	V_{CCCB}	E18	$\overline{\text{CSTOP}}$	ADDR20	J5	GND	GND
A13	V_{CC}	V_{CC}	E19	$\overline{\text{CBLOCK}}$	ADDR19	J6	AD28	AD28
A14	CAD20	ADDR6	F1	PHY_RSVD	PHY_RSVD	J14	CC/$\overline{\text{BE0}}$	$\overline{\text{CE1}}$
A15	GND	GND	F2	PHY_RSVD	PHY_RSVD	J15	CAD9	ADDR10
A16	$\overline{\text{CTRDY}}$	ADDR22	F3	PHY_RSVD	PHY_RSVD	J17	CAD8	DATA15
B5	PHY_DATA3	PHY_DATA3	F5	PHY_RSVD	PHY_RSVD	J18	V_{CC}	V_{CC}
B6	PHY_DATA0	PHY_DATA0	F6	PHY_DATA2	PHY_DATA2	J19	CAD7	DATA7
B7	PHY_CLK	PHY_CLK	F7	PHY_CTL(1)	PHY_CTL(1)	K1	AD27	AD27
B8	CRSVD	DATA2	F8	LPS	LPS	K2	AD26	AD26
B9	V_{CC}	V_{CC}	F9	CAD28	DATA8	K3	AD25	AD25
B10	$\overline{\text{CSERR}}$	$\overline{\text{WAIT}}$	F10	$\overline{\text{CCLKRUN}}$	WP($\overline{\text{IOIS16}}$)	K5	AD24	AD24
B11	CAD25	ADDR1	F11	CVS1	$\overline{\text{VS1}}$	K6	C/$\overline{\text{BE3}}$	C/$\overline{\text{BE3}}$
B12	CC/$\overline{\text{BE3}}$	$\overline{\text{REG}}$	F12	$\overline{\text{CRST}}$	RESET	K14	CRSVD	DATA14
B13	CAD22	ADDR4	F13	CC/$\overline{\text{BE2}}$	ADDR12	K15	CAD5	DATA6
B14	CAD19	ADDR25	F14	$\overline{\text{CPERR}}$	ADDR14	K17	CAD6	DATA13
B15	CAD17	ADDR24	F15	$\overline{\text{CGNT}}$	$\overline{\text{WE}}$	K18	CAD3	DATA5
C5	PHY_DATA5	PHY_DATA5	F17	V_{CC}	V_{CC}	K19	CAD4	DATA12
C6	PHY_DATA1	PHY_DATA1	F18	CRSVD	ADDR18	L1	IDSEL	IDSEL
C7	PHY_CTL(0)	PHY_CTL(0)	F19	CC/$\overline{\text{BE1}}$	ADDR8	L2	V_{CC}	V_{CC}
C8	CAD31	DATA10	G1	V_{CC}	V_{CC}	L3	AD23	AD23
C9	CAD27	DATA0	G2	PHY_RSVD	PHY_RSVD	L5	AD21	AD21
C10	CAUDIO	BVD2($\overline{\text{SPKR}}$)	G3	PHY_RSVD	PHY_RSVD	L6	AD22	AD22
C11	CAD26	ADDR0	G5	PHY_RSVD	PHY_RSVD	L14	CAD1	DATA4
C12	CAD23	ADDR3	G6	PHY_RSVD	PHY_RSVD	L15	GND	GND
C13	CAD21	ADDR5	G14	CAD16	ADDR17	L17	CAD2	DATA11
C14	CAD18	ADDR7	G15	CPAR	ADDR13	L18	CAD0	DATA3
C15	$\overline{\text{CIRDY}}$	ADDR15	G17	CAD14	ADDR9	L19	$\overline{\text{CCD1}}$	$\overline{\text{CD1}}$
D1	PHY_DATA7	PHY_DATA7	G18	CAD15	$\overline{\text{IOWR}}$	M1	V_{CCP}	V_{CCP}
D19	CCLK	ADDR16	G19	CAD12	ADDR11	M2	AD20	AD20
E1	GND	GND	H1	$\overline{\text{GNT}}$	$\overline{\text{GNT}}$	M3	$\overline{\text{PRST}}$	$\overline{\text{PRST}}$
E2	PHY_RSVD	PHY_RSVD	H2	$\overline{\text{REQ}}$	$\overline{\text{REQ}}$	M5	GND	GND
E3	PHY_RSVD	PHY_RSVD	H3	PHY_RSVD	PHY_RSVD	M6	PCLK	PCLK
E6	PHY_DATA4	PHY_DATA4	H5	PHY_RSVD	PHY_RSVD	M14	V_{CC}	V_{CC}
E7	V_{CCL}	V_{CCL}	H6	PHY_RSVD	PHY_RSVD	M15	ZV_SDATA	ZV_SDATA

Table 2−2. CardBus and 16-Bit PC Card Signal Names by GHK Terminal Number (Continued)

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CARDBUS	16-BIT		CARDBUS	16-BIT		CARDBUS	16-BIT
M17	ZV_PCLK	ZV_PCLK	P18	ZV_UV(6)	ZV_UV(6)	U14	ZV_Y(1)	ZV_Y(1)
M18	$\overline{\text{VCCD0}}$	$\overline{\text{VCCD0}}$	P19	ZV_SCLK	ZV_SCLK	U15	ZV_Y(5)	ZV_Y(5)
M19	$\overline{\text{VCCD1}}$	$\overline{\text{VCCD1}}$	R1	$\overline{\text{TRDY}}$	$\overline{\text{TRDY}}$	V5	AD12	AD12
N1	AD19	AD19	R2	$\overline{\text{STOP}}$	$\overline{\text{STOP}}$	V6	V_{CCP}	V_{CCP}
N2	AD18	AD18	R3	$\overline{\text{SERR}}$	$\overline{\text{SERR}}$	V7	AD7	AD7
N3	AD17	AD17	R6	AD14	AD14	V8	AD4	AD4
N5	$\overline{\text{IRDY}}$	$\overline{\text{IRDY}}$	R7	AD10	AD10	V9	AD1	AD1
N6	AD16	AD16	R8	AD6	AD6	V10	MFUNC1	MFUNC1
N14	ZV_UV(1)	ZV_UV(1)	R9	GND	GND	V11	$\overline{\text{GRST}}$	$\overline{\text{GRST}}$
N15	ZV_UV(5)	ZV_UV(5)	R10	V_{CCI}	V_{CCI}	V12	VPPD0	VPPD0
N17	ZV_UV(7)	ZV_UV(7)	R11	MFUNC6	MFUNC6	V13	$\overline{\text{INTA}}$	$\overline{\text{INTA}}$
N18	ZV_MCLK	ZV_MCLK	R12	LED_SKT	LED_SKT	V14	ZV_VSYNC	ZV_VSYNC
N19	ZV_LRCLK	ZV_LRCLK	R13	ZV_Y(0)	ZV_Y(0)	V15	ZV_Y(3)	ZV_Y(3)
P1	C/$\overline{\text{BE2}}$	C/$\overline{\text{BE2}}$	R14	ZV_Y(4)	ZV_Y(4)	W4	C/$\overline{\text{BE1}}$	C/$\overline{\text{BE1}}$
P2	$\overline{\text{FRAME}}$	$\overline{\text{FRAME}}$	R17	ZV_UV(0)	ZV_UV(0)	W5	GND	GND
P3	V_{CC}	V_{CC}	R18	ZV_UV(4)	ZV_UV(4)	W6	AD9	AD9
P5	$\overline{\text{PERR}}$	$\overline{\text{PERR}}$	R19	GND	GND	W7	V_{CC}	V_{CC}
P6	$\overline{\text{DEVSEL}}$	$\overline{\text{DEVSEL}}$	T1	PAR	PAR	W8	AD3	AD3
P7	AD13	AD13	T19	ZV_Y(7)	ZV_Y(7)	W9	AD2	AD2
P8	AD8	AD8	U5	AD15	AD15	W10	MFUNC0	MFUNC0
P9	$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	U6	AD11	AD11	W11	MFUNC3	MFUNC3
P10	MFUNC2	MFUNC2	U7	C/$\overline{\text{BE0}}$	C/$\overline{\text{BE0}}$	W12	$\overline{\text{SUSPEND}}$	$\overline{\text{SUSPEND}}$
P11	MFUNC5	MFUNC5	U8	AD5	AD5	W13	V_{CC}	V_{CC}
P12	RSVD	RSVD	U9	AD0	AD0	W14	ZV_HREF	ZV_HREF
P13	RSVD	RSVD	U10	SPKROUT	SPKROUT	W15	ZV_Y(2)	ZV_Y(2)
P14	GND	GND	U11	MFUNC4	MFUNC4	W16	ZV_Y(6)	ZV_Y(6)
P15	ZV_UV(2)	ZV_UV(2)	U12	VPPD1	VPPD1			
P17	ZV_UV(3)	ZV_UV(3)	U13	$\overline{\text{INTB}}$	$\overline{\text{INTB}}$			

Table 2−3. CardBus PC Card Signal Names Sorted Alphabetically to GHK/PDV Terminal Number

SIGNAL NAME	TERM. NO.		SIGNAL NAME	TERM. NO.		SIGNAL NAME	TERM. NO.		SIGNAL NAME	TERM. NO.	
	PDV	GHK		PDV	GHK		PDV	GHK		PDV	GHK
AD0	73	U9	CAD11	140	H17	$\overline{\text{CRST}}$	167	F12	PHY_CLK	196	B7
AD1	72	V9	CAD12	143	G19	CRSVD	132	K14	PHY_CTL(0)	197	C7
AD2	71	W9	CAD13	141	H14	CRSVD	148	F18	PHY_CTL(1)	198	F7
AD3	70	W8	CAD14	145	G17	CRSVD	191	B8	PHY_DATA0	200	B6
AD4	69	V8	CAD15	144	G18	$\overline{\text{CSERR}}$	181	B10	PHY_DATA1	202	C6
AD5	68	U8	CAD16	146	G14	$\overline{\text{CSTOP}}$	153	E18	PHY_DATA2	204	F6
AD6	67	R8	CAD17	161	B15	CSTSCHG	183	E10	PHY_DATA3	205	B5
AD7	65	V7	CAD18	163	C14	$\overline{\text{CTRDY}}$	157	A16	PHY_DATA4	206	E6
AD8	63	P8	CAD19	164	B14	CVS1	179	F11	PHY_DATA5	207	C5
AD9	62	W6	CAD20	166	A14	CVS2	165	E13	PHY_DATA6	208	A4
AD10	61	R7	CAD21	168	C13	$\overline{\text{DEVSEL}}$	48	P6	PHY_DATA7	1	D1
AD11	59	U6	CAD22	169	B13	$\overline{\text{FRAME}}$	44	P2	PHY_LREQ	194	E8
AD12	57	V5	CAD23	172	C12	GND	6	E1	PHY_RSVD	2	E3
AD13	56	P7	CAD24	175	A11	GND	22	J5	PHY_RSVD	3	F5
AD14	55	R6	CAD25	176	B11	GND	38	M5	PHY_RSVD	4	G6
AD15	54	U5	CAD26	177	C11	GND	58	W5	PHY_RSVD	5	E2
AD16	42	N6	CAD27	187	C9	GND	74	R9	PHY_RSVD	7	F3
AD17	41	N3	CAD28	188	F9	GND	100	P14	PHY_RSVD	8	F2
AD18	40	N2	CAD29	189	E9	GND	110	R19	PHY_RSVD	9	G5
AD19	39	N1	CAD30	190	A8	GND	126	L15	PHY_RSVD	10	F1
AD20	35	M2	CAD31	192	C8	GND	142	H15	PHY_RSVD	11	H6
AD21	33	L5	CAUDIO	182	C10	GND	162	A15	PHY_RSVD	12	G3
AD22	32	L6	C/$\overline{\text{BE0}}$	64	U7	GND	178	E11	PHY_RSVD	13	G2
AD23	31	L3	C/$\overline{\text{BE1}}$	53	W4	GND	203	A5	PHY_RSVD	15	H5
AD24	27	K5	C/$\overline{\text{BE2}}$	43	P1	$\overline{\text{GNT}}$	18	H1	PHY_RSVD	16	H3
AD25	26	K3	C/$\overline{\text{BE3}}$	28	K6	$\overline{\text{GRST}}$	82	V11	$\overline{\text{PRST}}$	36	M3
AD26	25	K2	$\overline{\text{CBLOCK}}$	151	E19	IDSEL	29	L1	$\overline{\text{REQ}}$	17	H2
AD27	24	K1	CC/$\overline{\text{BE0}}$	136	J14	$\overline{\text{INTA}}$	92	V13	$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	75	P9
AD28	23	J6	CC/$\overline{\text{BE1}}$	147	F19	$\overline{\text{INTB}}$	93	U13	RSVD	89	P12
AD29	21	J3	CC/$\overline{\text{BE2}}$	160	F13	$\overline{\text{IRDY}}$	45	N5	RSVD	94	P13
AD30	20	J2	CC/$\overline{\text{BE3}}$	173	B12	LED_SKT	90	R12	$\overline{\text{SERR}}$	51	R3
AD31	19	J1	$\overline{\text{CCD1}}$	123	L19	LINKON	199	A6	SPKROUT	78	U10
CAD0	124	L18	$\overline{\text{CCD2}}$	185	A9	LPS	193	F8	$\overline{\text{STOP}}$	49	R2
CAD1	127	L14	CCLK	156	D19	MFUNC0	76	W10	$\overline{\text{SUSPEND}}$	86	W12
CAD2	125	L17	$\overline{\text{CCLKRUN}}$	184	F10	MFUNC1	77	V10	$\overline{\text{TRDY}}$	47	R1
CAD3	129	K18	$\overline{\text{CDEVSEL}}$	155	E17	MFUNC2	80	P10	V_{CC}	14	G1
CAD4	128	K19	$\overline{\text{CFRAME}}$	159	E14	MFUNC3	81	W11	V_{CC}	30	L2
CAD5	131	K15	$\overline{\text{CGNT}}$	154	F15	MFUNC4	83	U11	V_{CC}	46	P3
CAD6	130	K17	$\overline{\text{CINT}}$	180	A10	MFUNC5	84	P11	V_{CC}	66	W7
CAD7	133	J19	$\overline{\text{CIRDY}}$	158	C15	MFUNC6	85	R11	V_{CC}	91	W13
CAD8	135	J17	CPAR	149	G15	PAR	52	T1	V_{CC}	115	M14
CAD9	137	J15	$\overline{\text{CPERR}}$	152	F14	PCLK	37	M6	V_{CC}	134	J18
CAD10	139	H18	$\overline{\text{CREQ}}$	171	E12	$\overline{\text{PERR}}$	50	P5	V_{CC}	150	F17

Table 2−3. CardBus PC Card Signal Names Sorted Alphabetically to GHK/PDV Terminal Number (Continued)

SIGNAL NAME	TERM. NO.		SIGNAL NAME	TERM. NO.		SIGNAL NAME	TERM. NO.		SIGNAL NAME	TERM. NO.	
	PDV	GHK		PDV	GHK		PDV	GHK		PDV	GHK
V_{CC}	170	A13	V_{CCP}	34	M1	ZV_SDATA	119	M15	ZV_VSYNC	96	V14
V_{CC}	186	B9	V_{CCP}	60	V6	ZV_UV(0)	106	R17	ZV_Y(0)	97	R13
V_{CC}	195	A7	VPPD0	87	V12	ZV_UV(1)	108	N14	ZV_Y(1)	98	U14
V_{CCCB}	138	H19	VPPD1	88	U12	ZV_UV(2)	107	P15	ZV_Y(2)	99	W15
V_{CCCB}	174	A12	ZV_HREF	95	W14	ZV_UV(3)	111	P17	ZV_Y(3)	101	V15
$\overline{VCCD0}$	121	M18	ZV_LRCLK	118	N19	ZV_UV(4)	109	R18	ZV_Y(4)	102	R14
$\overline{VCCD1}$	122	M19	ZV_MCLK	117	N18	ZV_UV(5)	113	N15	ZV_Y(5)	103	U15
V_{CCI}	79	R10	ZV_PCLK	120	M17	ZV_UV(6)	112	P18	ZV_Y(6)	104	W16
V_{CCL}	201	E7	ZV_SCLK	114	P19	ZV_UV(7)	116	N17	ZV_Y(7)	105	T19

Table 2−4. 16-Bit PC Card Signal Names Sorted Alphabetically to GHK/PDV Terminal Number

SIGNAL NAME	TERM. NO.		SIGNAL NAME	TERM. NO.		SIGNAL NAME	TERM. NO.		SIGNAL NAME	TERM. NO.	
	PDV	GHK		PDV	GHK		PDV	GHK		PDV	GHK
AD0	73	U9	ADDR10	137	J15	$\overline{\text{DEVSEL}}$	48	P6	PHY_DATA2	204	F6
AD1	72	V9	ADDR11	143	G19	$\overline{\text{FRAME}}$	44	P2	PHY_DATA3	205	B5
AD2	71	W9	ADDR12	160	F13	GND	6	E1	PHY_DATA4	206	E6
AD3	70	W8	ADDR13	149	G15	GND	22	J5	PHY_DATA5	207	C5
AD4	69	V8	ADDR14	152	F14	GND	38	M5	PHY_DATA6	208	A4
AD5	68	U8	ADDR15	158	C15	GND	58	W5	PHY_DATA7	1	D1
AD6	67	R8	ADDR16	156	D19	GND	74	R9	PHY_LREQ	194	E8
AD7	65	V7	ADDR17	146	G14	GND	100	P14	PHY_RSVD	2	E3
AD8	63	P8	ADDR18	148	F18	GND	110	R19	PHY_RSVD	3	F5
AD9	62	W6	ADDR19	151	E19	GND	126	L15	PHY_RSVD	4	G6
AD10	61	R7	ADDR20	153	E18	GND	142	H15	PHY_RSVD	5	E2
AD11	59	U6	ADDR21	155	E17	GND	162	A15	PHY_RSVD	7	F3
AD12	57	V5	ADDR22	157	A16	GND	178	E11	PHY_RSVD	8	F2
AD13	56	P7	ADDR23	159	E14	GND	203	A5	PHY_RSVD	9	G5
AD14	55	R6	ADDR24	161	B15	$\overline{\text{GNT}}$	18	H1	PHY_RSVD	10	F1
AD15	54	U5	ADDR25	164	B14	$\overline{\text{GRST}}$	82	V11	PHY_RSVD	11	H6
AD16	42	N6	BVD1 ($\overline{\text{STSCHG}}$/$\overline{\text{RI}}$)	183	E10	IDSEL	29	L1	PHY_RSVD	12	G3
AD17	41	N3	BVD2($\overline{\text{SPKR}}$)	182	C10	$\overline{\text{INPACK}}$	171	E12	PHY_RSVD	13	G2
AD18	40	N2	C/$\overline{\text{BE0}}$	64	U7	$\overline{\text{INTA}}$	92	V13	PHY_RSVD	15	H5
AD19	39	N1	C/$\overline{\text{BE1}}$	53	W4	$\overline{\text{INTB}}$	93	U13	PHY_RSVD	16	H3
AD20	35	M2	C/$\overline{\text{BE2}}$	43	P1	$\overline{\text{IRDY}}$	45	N5	$\overline{\text{PRST}}$	36	M3
AD21	33	L5	C/$\overline{\text{BE3}}$	28	K6	$\overline{\text{IORD}}$	141	H14	READY($\overline{\text{IREQ}}$)	180	A10
AD22	32	L6	$\overline{\text{CD1}}$	123	L19	$\overline{\text{IOWR}}$	144	G18	$\overline{\text{REG}}$	173	B12
AD23	31	L3	$\overline{\text{CD2}}$	185	A9	LED_SKT	90	R12	$\overline{\text{REQ}}$	17	H2
AD24	27	K5	$\overline{\text{CE1}}$	136	J14	LINKON	199	A6	RESET	167	F12
AD25	26	K3	$\overline{\text{CE2}}$	139	H18	LPS	193	F8	$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	75	P9
AD26	25	K2	DATA0	187	C9	MFUNC0	76	W10	RSVD	89	P12
AD27	24	K1	DATA1	189	E9	MFUNC1	77	V10	RSVD	94	P13
AD28	23	J6	DATA2	191	B8	MFUNC2	80	P10	$\overline{\text{SERR}}$	51	R3
AD29	21	J3	DATA3	124	L18	MFUNC3	81	W11	SPKROUT	78	U10
AD30	20	J2	DATA4	127	L14	MFUNC4	83	U11	$\overline{\text{STOP}}$	49	R2
AD31	19	J1	DATA5	129	K18	MFUNC5	84	P11	$\overline{\text{SUSPEND}}$	86	W12
ADDR0	177	C11	DATA6	131	K15	MFUNC6	85	R11	$\overline{\text{TRDY}}$	47	R1
ADDR1	176	B11	DATA7	133	J19	$\overline{\text{OE}}$	140	H17	V_{CC}	14	G1
ADDR2	175	A11	DATA8	188	F9	PAR	52	T1	V_{CC}	30	L2
ADDR3	172	C12	DATA9	190	A8	PCLK	37	M6	V_{CC}	46	P3
ADDR4	169	B13	DATA10	192	C8	$\overline{\text{PERR}}$	50	P5	V_{CC}	66	W7
ADDR5	168	C13	DATA11	125	L17	PHY_CLK	196	B7	V_{CC}	91	W13
ADDR6	166	A14	DATA12	128	K19	PHY_CTL(0)	197	C7	V_{CC}	115	M14
ADDR7	163	C14	DATA13	130	K17	PHY_CTL(1)	198	F7	V_{CC}	134	J18
ADDR8	147	F19	DATA14	132	K14	PHY_DATA0	200	B6	V_{CC}	150	F17
ADDR9	145	G17	DATA15	135	J17	PHY_DATA1	202	C6	V_{CC}	170	A13

Table 2−4. 16-Bit PC Card Signal Names Sorted Alphabetically to GHK/PDV Terminal Number (Continued)

SIGNAL NAME	TERM. NO.		SIGNAL NAME	TERM. NO.		SIGNAL NAME	TERM. NO.		SIGNAL NAME	TERM. NO.	
	PDV	GHK		PDV	GHK		PDV	GHK		PDV	GHK
V_{CC}	186	B9	VPPD0	87	V12	ZV_PCLK	120	M17	ZV_UV(7)	116	N17
V_{CC}	195	A7	VPPD1	88	U12	ZV_SCLK	114	P19	ZV_VSYNC	96	V14
V_{CCCB}	138	H19	$\overline{VS1}$	179	F11	ZV_SDATA	119	M15	ZV_Y(0)	97	R13
V_{CCCB}	174	A12	$\overline{VS2}$	165	E13	ZV_UV(0)	106	R17	ZV_Y(1)	98	U14
$\overline{VCCD0}$	121	M18	$\overline{WAIT}$	181	B10	ZV_UV(1)	108	N14	ZV_Y(2)	99	W15
$\overline{VCCD1}$	122	M19	$\overline{WE}$	154	F15	ZV_UV(2)	107	P15	ZV_Y(3)	101	V15
V_{CCI}	79	R10	WP($\overline{IOIS16}$)	184	F10	ZV_UV(3)	111	P17	ZV_Y(4)	102	R14
V_{CCL}	201	E7	ZV_HREF	95	W14	ZV_UV(4)	109	R18	ZV_Y(5)	103	U15
V_{CCP}	34	M1	ZV_LRCLK	118	N19	ZV_UV(5)	113	N15	ZV_Y(6)	104	W16
V_{CCP}	60	V6	ZV_MCLK	117	N18	ZV_UV(6)	112	P18	ZV_Y(7)	105	T19

The terminals are grouped in tables by functionality, such as PCI system function and power-supply function (see Table 2−5 through Table 2−17). The terminal numbers are also listed for convenient reference.

Table 2−5. Power Supply Terminals

TERMINAL			DESCRIPTION
NAME	NUMBER		
	PDV	GHK	
GND	6, 22, 38, 58, 74, 100, 126, 142, 162, 178, 203	A5, A15, E1, E11, H15, J5, L15, M5, P14, R9, W5	Device ground terminals
V_{CC}	14, 30, 46, 66, 91, 115, 134, 150, 170, 186, 195	A7, A13, B9, F17, G1, J18, L2, M14, P3, W7, W13	Power supply terminal for core logic (3.3 V)
V_{CCCB}	138, 174	A12, H19	Clamp voltage for PC Card interface. Matches card signaling environment, 5 V or 3.3 V.
V_{CCI}	79	R10	Clamp voltage for miscellaneous I/O signals (MFUNC, $\overline{\text{GRST}}$, and $\overline{\text{SUSPEND}}$)
V_{CCL}	201	E7	Clamp voltage for 1394 link function
V_{CCP}	34, 60	M1, V6	Clamp voltage for PCI interface, ZV interface, SPKROUT, $\overline{\text{INTA}}$, $\overline{\text{INTB}}$ LED_SKT, $\overline{\text{VCCD0}}$, $\overline{\text{VCCD1}}$, VPPD0, VPPD1

Table 2−6. PC Card Power Switch Terminals

TERMINAL			I/O	DESCRIPTION
NAME	NUMBER			
	PDV	GHK		
$\overline{\text{VCCD0}}$ $\overline{\text{VCCD1}}$	121 122	M18 M19	O	Logic controls to the TPS2211 PC Card power interface switch to control AVCC
VPPD0 VPPD1	87 88	V12 U12	O	Logic controls to the TPS2211 PC Card power interface switch to control AVPP

Table 2−7. PCI System Terminals

TERMINAL			I/O	DESCRIPTION
NAME	NUMBER			
	PDV	GHK		
$\overline{\text{GRST}}$	82	V11	I	Global reset. When the global reset is asserted, the $\overline{\text{GRST}}$ signal causes the PCI4410 to place all output buffers in a high-impedance state and reset all internal registers. When $\overline{\text{GRST}}$ is asserted, the device is completely in its default state. For systems that require wake-up from D3, $\overline{\text{GRST}}$ will normally be asserted only during initial boot. $\overline{\text{PRST}}$ should be asserted following initial boot so that PME context is retained when transitioning from D3 to D0. For systems that do not require wake-up from D3, $\overline{\text{GRST}}$ should be tied to $\overline{\text{PRST}}$. When the $\overline{\text{SUSPEND}}$ mode is enabled, the device is protected from the $\overline{\text{GRST}}$, and the internal registers are preserved. All outputs are placed in a high-impedance state, but the contents of the registers are preserved.
PCLK	37	M6	I	PCI bus clock. PCLK provides timing for all transactions on the PCI bus. All PCI signals are sampled at the rising edge of PCLK.
$\overline{\text{PRST}}$	36	M3	I	PCI bus reset. When the PCI bus reset is asserted, $\overline{\text{PRST}}$ causes the PCI4410 to place all output buffers in a high-impedance state and reset internal registers. When $\overline{\text{PRST}}$ is asserted, the device is completely nonfunctional. After $\overline{\text{PRST}}$ is deasserted, the PCI4410 is in a default state. When $\overline{\text{SUSPEND}}$ and $\overline{\text{PRST}}$ are asserted, the device is protected from $\overline{\text{PRST}}$ clearing the internal registers. All outputs are placed in a high-impedance state, but the contents of the registers are preserved.

Table 2−8. PCI Address and Data Terminals

TERMINAL NAME	TERMINAL NUMBER PDV	TERMINAL NUMBER GHK	I/O	DESCRIPTION
AD31	19	J1	I/O	PCI address/data bus. These signals make up the multiplexed PCI address and data bus on the primary interface. During the address phase of a primary bus PCI cycle, AD31−AD0 contain a 32-bit address or other destination information. During the data phase, AD31−AD0 contain data.
AD30	20	J2		
AD29	21	J3		
AD28	23	J6		
AD27	24	K1		
AD26	25	K2		
AD25	26	K3		
AD24	27	K5		
AD23	31	L3		
AD22	32	L6		
AD21	33	L5		
AD20	35	M2		
AD19	39	N1		
AD18	40	N2		
AD17	41	N3		
AD16	42	N6		
AD15	54	U5		
AD14	55	R6		
AD13	56	P7		
AD12	57	V5		
AD11	59	U6		
AD10	61	R7		
AD9	62	W6		
AD8	63	P8		
AD7	65	V7		
AD6	67	R8		
AD5	68	U8		
AD4	69	V8		
AD3	70	W8		
AD2	71	W9		
AD1	72	V9		
AD0	73	U9		
C/$\overline{BE3}$	28	K6	I/O	PCI bus commands and byte enables. These signals are multiplexed on the same PCI terminals. During the address phase of a primary bus PCI cycle, C/$\overline{BE3}$−C/$\overline{BE0}$ define the bus command. During the data phase, this 4-bit bus is used as byte enables. The byte enables determine which byte paths of the full 32-bit data bus carry meaningful data. C/$\overline{BE0}$ applies to byte 0 (AD7−AD0), C/$\overline{BE1}$ applies to byte 1 (AD15−AD8), C/$\overline{BE2}$ applies to byte 2 (AD23−AD16), and C/$\overline{BE3}$ applies to byte 3 (AD31−AD24).
C/$\overline{BE2}$	43	P1		
C/$\overline{BE1}$	53	W4		
C/$\overline{BE0}$	64	U7		
PAR	52	T1	I/O	PCI bus parity. In all PCI bus read and write cycles, the PCI4410 calculates even parity across the AD31−AD0 and C/$\overline{BE3}$−C/$\overline{BE0}$ buses. As an initiator during PCI cycles, the PCI4410 outputs this parity indicator with a one-PCLK delay. As a target during PCI cycles, the calculated parity is compared to the initiator's parity indicator. A compare error results in the assertion of a parity error ($\overline{PERR}$).

Table 2−9. PCI Interface Control Terminals

TERMINAL NAME	TERMINAL NUMBER PDV	TERMINAL NUMBER GHK	I/O	DESCRIPTION
$\overline{\text{DEVSEL}}$	48	P6	I/O	PCI device select. The PCI4410 asserts $\overline{\text{DEVSEL}}$ to claim a PCI cycle as the target device. As a PCI initiator on the bus, the PCI4410 monitors $\overline{\text{DEVSEL}}$ until a target responds. If no target responds before timeout occurs, then the PCI4410 terminates the cycle with an initiator abort.
$\overline{\text{FRAME}}$	44	P2	I/O	PCI cycle frame. $\overline{\text{FRAME}}$ is driven by the initiator of a bus cycle. $\overline{\text{FRAME}}$ is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When $\overline{\text{FRAME}}$ is deasserted, the PCI bus transaction is in the final data phase.
$\overline{\text{GNT}}$	18	H1	I	PCI bus grant. $\overline{\text{GNT}}$ is driven by the PCI bus arbiter to grant the PCI4410 access to the PCI bus after the current data transaction has completed. $\overline{\text{GNT}}$ may or may not follow a PCI bus request, depending on the PCI bus parking algorithm.
IDSEL	29	L1	I	Initialization device select. IDSEL selects the PCI4410 during configuration space accesses. IDSEL can be connected to one of the upper 24 PCI address lines on the PCI bus.
$\overline{\text{IRDY}}$	45	N5	I/O	PCI initiator ready. $\overline{\text{IRDY}}$ indicates the PCI bus initiator's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCLK where both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. Until $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are both sampled asserted, wait states are inserted.
$\overline{\text{PERR}}$	50	P5	I/O	PCI parity error indicator. $\overline{\text{PERR}}$ is driven by a PCI device to indicate that calculated parity does not match PAR when $\overline{\text{PERR}}$ is enabled through bit 6 of the command register (see Section 4.4).
$\overline{\text{REQ}}$	17	H2	O	PCI bus request. $\overline{\text{REQ}}$ is asserted by the PCI4410 to request access to the PCI bus as an initiator.
$\overline{\text{SERR}}$	51	R3	O	PCI system error. $\overline{\text{SERR}}$ is an output that is pulsed from the PCI4410 when enabled through bit 8 of the command register (see Section 4.4) indicating a system error has occurred. The PCI4410 need not be the target of the PCI cycle to assert this signal. When $\overline{\text{SERR}}$ is enabled in the command register, this signal also pulses, indicating that an address parity error has occurred on a CardBus interface.
$\overline{\text{STOP}}$	49	R2	I/O	PCI cycle stop signal. $\overline{\text{STOP}}$ is driven by a PCI target to request the initiator to stop the current PCI bus transaction. $\overline{\text{STOP}}$ is used for target disconnects and is commonly asserted by target devices that do not support burst data transfers.
$\overline{\text{TRDY}}$	47	R1	I/O	PCI target ready. $\overline{\text{TRDY}}$ indicates the primary bus target's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCLK when both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. Until both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted, wait states are inserted.

Table 2−10. Multifunction and Miscellaneous Terminals

TERMINAL			I/O	DESCRIPTION
NAME	NUMBER			
	PDV	GHK		
$\overline{\text{INTA}}$	92	V13	O	Parallel PCI interrupt. $\overline{\text{INTA}}$
$\overline{\text{INTB}}$	93	U13	O	Parallel PCI interrupt. $\overline{\text{INTB}}$
LED_SKT	90	R12	O	PC Card socket activity LED indicator. LED_SKT provides an output indicating PC Card socket activity.
MFUNC0	76	W10	I/O	Multifunction terminal 0. MFUNC0 can be configured as parallel PCI interrupt $\overline{\text{INTA}}$, GPI0, GPO0, socket activity LED output, ZV switching outputs, CardBus audio PWM, $\overline{\text{GPE}}$, or a parallel IRQ. See Section 4.32, *Multifunction Routing Register*, for configuration details.
MFUNC1	77	V10	I/O	Multifunction terminal 1. MFUNC1 can be configured as GPI1, GPO1, socket activity LED output, ZV switching outputs, CardBus audio PWM, $\overline{\text{GPE}}$, or a parallel IRQ. See Section 4.32, *Multifunction Routing Register*, for configuration details. Serial data (SDA). When VCCD0 and $\overline{\text{VCCD1}}$ are high after a PCI reset, the MFUNC1 terminal provides the SDA signaling for the serial bus interface. The two-terminal serial interface loads the subsystem identification and other register defaults from an EEPROM after a PCI reset. See Section 3.6.1, *Serial Bus Interface Implementation*, for details on other serial bus applications.
MFUNC2	80	P10	I/O	Multifunction terminal 2. MFUNC2 can be configured as PC/PCI DMA request, GPI2, GPO2, ZV switching outputs, CardBus audio PWM, $\overline{\text{GPE}}$, $\overline{\text{RI\_OUT}}$, or a parallel IRQ. See Section 4.32, *Multifunction Routing Register*, for configuration details.
MFUNC3	81	W11	I/O	Multifunction terminal 3. MFUNC3 can be configured as a parallel IRQ or the serialized interrupt signal IRQSER. See Section 4.32, *Multifunction Routing Register*, for configuration details.
MFUNC4	83	U11	I/O	Multifunction terminal 4. MFUNC4 can be configured as PCI $\overline{\text{LOCK}}$, GPI3, GPO3, socket activity LED output, ZV switching outputs, CardBus audio PWM, $\overline{\text{GPE}}$, $\overline{\text{RI\_OUT}}$, or a parallel IRQ. See Section 4.32, *Multifunction Routing Register*, for configuration details. Serial clock (SCL). When $\overline{\text{VCCD0}}$ and $\overline{\text{VCCD1}}$ are high after a PCI reset, the MFUNC4 terminal provides the SCL signaling for the serial bus interface. The two-terminal serial interface loads the subsystem identification and other register defaults from an EEPROM after a PCI reset. See Section 3.6.1, *Serial Bus Interface Implementation*, for details on other serial bus applications.
MFUNC5	84	P11	I/O	Multifunction terminal 5. MFUNC5 can be configured as PC/PCI DMA grant, GPI4, GPO4, socket activity LED output, ZV switching outputs, CardBus audio PWM, $\overline{\text{GPE}}$, or a parallel IRQ. See Section 4.32, *Multifunction Routing Register*, for configuration details.
MFUNC6	85	R11	I/O	Multifunction terminal 6. MFUNC6 can be configured as a PCI $\overline{\text{CLKRUN}}$ or a parallel IRQ. See Section 4.32, *Multifunction Routing Register*, for configuration details.
$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	75	P9	O	Ring indicate out and power management event output. Terminal provides an output for ring-indicate or $\overline{\text{PME}}$ signals.
SPKROUT	78	U10	O	Speaker output. SPKROUT is the output to the host system that can carry $\overline{\text{SPKR}}$ or CAUDIO through the PCI4410 from the PC Card interface. SPKROUT is driven as the exclusive-OR combination of card $\overline{\text{SPKR}}$//CAUDIO inputs.
$\overline{\text{SUSPEND}}$	86	W12	I	Suspend. $\overline{\text{SUSPEND}}$ protects the internal registers from clearing when the $\overline{\text{GRST}}$ or $\overline{\text{PRST}}$ signal is asserted. See Section 3.8.4, *Suspend Mode*, for details.

Table 2−11. 16-Bit PC Card Address and Data Terminals

TERMINAL			I/O	DESCRIPTION
NAME	NUMBER			
	PDV	GHK		
ADDR25	164	B14	O	PC Card address. 16-bit PC Card address lines. ADDR25 is the most significant bit.
ADDR24	161	B15		
ADDR23	159	E14		
ADDR22	157	A16		
ADDR21	155	E17		
ADDR20	153	E18		
ADDR19	151	E19		
ADDR18	148	F18		
ADDR17	146	G14		
ADDR16	156	D19		
ADDR15	158	C15		
ADDR14	152	F14		
ADDR13	149	G15		
ADDR12	160	F13		
ADDR11	143	G19		
ADDR10	137	J15		
ADDR9	145	G17		
ADDR8	147	F19		
ADDR7	163	C14		
ADDR6	166	A14		
ADDR5	168	C13		
ADDR4	169	B13		
ADDR3	172	C12		
ADDR2	175	A11		
ADDR1	176	B11		
ADDR0	177	C11		
DATA15	135	J17	I/O	PC Card data. 16-bit PC Card data lines. DATA15 is the most significant bit.
DATA14	132	K14		
DATA13	130	K17		
DATA12	128	K19		
DATA11	189	L17		
DATA10	192	C8		
DATA9	190	A8		
DATA8	188	F9		
DATA7	133	J19		
DATA6	131	K15		
DATA5	129	K18		
DATA4	127	L14		
DATA3	124	L18		
DATA2	191	B8		
DATA1	189	E9		
DATA0	187	C9		

Table 2−12. 16-Bit PC Card Interface Control Terminals

TERMINAL			I/O	DESCRIPTION
NAME	NUMBER			
	PDV	GHK		
BVD1 (STSCHG/RI)	183	E10	I	Battery voltage detect 1. BVD1 is generated by 16-bit memory PC Cards that include batteries. BVD1 is used with BVD2 as an indication of the condition of the batteries on a memory PC Card. Both BVD1 and BVD2 are high when the battery is good. When BVD2 is low and BVD1 is high, the battery is weak and should be replaced. When BVD1 is low, the battery is no longer serviceable and the data in the memory PC Card is lost. See Section 5.6, *ExCA Card Status-Change-Interrupt Configuration Register*, for enable bits. See Section 5.5, *ExCA Card Status-Change Register*, and Section 5.2, *ExCA Interface Status Register*, for the status bits for this signal. Status change. STSCHG is used to alert the system to a change in the READY, write protect, or battery voltage dead condition of a 16-bit I/O PC Card. Ring indicate. RI is used by 16-bit modem cards to indicate a ring detection.
BVD2 (SPKR)	182	C10	I	Battery voltage detect 2. BVD2 is generated by 16-bit memory PC Cards that include batteries. BVD2 is used with BVD1 as an indication of the condition of the batteries on a memory PC Card. Both BVD1 and BVD2 are high when the battery is good. When BVD2 is low and BVD1 is high, the battery is weak and should be replaced. When BVD1 is low, the battery is no longer serviceable and the data in the memory PC Card is lost. See Section 5.6, *ExCA Card Status-Change-Interrupt Configuration Register*, for enable bits. See Section 5.5, *ExCA Card Status-Change Register*, and Section 5.2, *ExCA Interface Status Register*, for the status bits for this signal. Speaker. SPKR is an optional binary audio signal available only when the card and socket have been configured for the 16-bit I/O interface. The audio signals from cards A and B are combined by the PCI4410 and are output on SPKROUT. DMA request. BVD2 can be used as the DMA request signal during DMA operations to a 16-bit PC Card that supports DMA. The PC Card asserts BVD2 to indicate a request for a DMA operation.
CD1 CD2	123 185	L19 A9	I	Card detect 1 and Card detect 2. CD1 and CD2 are internally connected to ground on the PC Card. When a PC Card is inserted into a socket, CD1 and CD2 are pulled low. For signal status, see Section 5.2, *ExCA Interface Status Register*.
CE1 CE2	136 139	J14 H18	O	Card enable 1 and card enable 2. CE1 and CE2 enable even- and odd-numbered address bytes. CE1 enables even-numbered address bytes, and CE2 enables odd-numbered address bytes.
INPACK	171	E12	I	Input acknowledge. INPACK is asserted by the PC Card when it can respond to an I/O read cycle at the current address. DMA request. INPACK can be used as the DMA request signal during DMA operations from a 16-bit PC Card that supports DMA. If it is used as a strobe, then the PC Card asserts this signal to indicate a request for a DMA operation.
IORD	141	H14	O	I/O read. IORD is asserted by the PCI4410 to enable 16-bit I/O PC Card data output during host I/O read cycles. DMA write. IORD is used as the DMA write strobe during DMA operations from a 16-bit PC Card that supports DMA. The PCI4410 asserts IORD during DMA transfers from the PC Card to host memory.
IOWR	144	G18	O	I/O write. IOWR is driven low by the PCI4410 to strobe write data into 16-bit I/O PC Cards during host I/O write cycles. DMA read. IOWR is used as the DMA write strobe during DMA operations from a 16-bit PC Card that supports DMA. The PCI4410 asserts IOWR during transfers from host memory to the PC Card.
OE	140	H17	O	Output enable. OE is driven low by the PCI4410 to enable 16-bit memory PC Card data output during host memory read cycles. DMA terminal count. OE is used as terminal count (TC) during DMA operations to a 16-bit PC Card that supports DMA. The PCI4410 asserts OE to indicate TC for a DMA write operation.

Table 2−12. 16-Bit PC Card Interface Control Terminals (Continued)

TERMINAL			I/O	DESCRIPTION
NAME	NUMBER			
	PDV	GHK		
READY (IREQ)	180	A10	I	Ready. The ready function is provided by READY when the 16-bit PC Card and the host socket are configured for the memory-only interface. READY is driven low by the 16-bit memory PC Cards to indicate that the memory card circuits are busy processing a previous write command. READY is driven high when the 16-bit memory PC Card is ready to accept a new data transfer command. Interrupt request. IREQ is asserted by a 16-bit I/O PC Card to indicate to the host that a device on the 16-bit I /O PC Card requires service by the host software. IREQ is high (deasserted) when no interrupt is requested.
REG	173	B12	O	Attribute memory select. REG remains high for all common memory accesses. When REG is asserted, access is limited to attribute memory (OE or WE active) and to the I/O space (IORD or IOWR active). Attribute memory is a separately accessed section of card memory and is generally used to record card capacity and other configuration and attribute information. DMA acknowledge. REG is used as a DMA acknowledge (DACK) during DMA operations to a 16-bit PC Card that supports DMA. The PCI4410 asserts REG to indicate a DMA operation. REG is used in conjunction with the DMA read (IOWR) or DMA write (IORD) strobes to transfer data.
RESET	167	F12	O	PC Card reset. RESET forces a hard reset to a 16-bit PC Card.
WAIT	181	B10	I	Bus cycle wait. WAIT is driven by a 16-bit PC Card to extend the completion of the memory or I/O cycle in progress.
WE	154	F15	O	Write enable. WE is used to strobe memory write data into 16-bit memory PC Cards. WE is also used for memory PC Cards that employ programmable memory technologies. DMA terminal count. WE is used as TC during DMA operations to a 16-bit PC Card that supports DMA. The PCI4410 asserts WE to indicate TC for a DMA read operation.
WP (IOIS16)	184	F10	I	Write protect. WP applies to 16-bit memory PC Cards. WP reflects the status of the write-protect switch on 16-bit memory PC Cards. For 16-bit I/O PC cards, WP is used for the 16-bit port (IOIS16) function. I/O is 16 bits. IOIS16 applies to 16-bit I/O PC Cards. IOIS16 is asserted by the 16-bit PC Card when the address on the bus corresponds to an address to which the 16-bit PC Card responds, and the I/O port that is addressed is capable of 16-bit accesses. DMA request. WP can be used as the DMA request signal during DMA operations to a 16-bit PC Card that supports DMA. If used, then the PC Card asserts WP to indicate a request for a DMA operation.
VS1 VS2	179 165	F11 E13	I/O	Voltage sense 1 and voltage sense 2. VS1 and VS2, when used in conjunction with each other, determine the operating voltage of the PC Card.

Table 2−13. CardBus PC Card Interface System Terminals

TERMINAL			I/O	DESCRIPTION
NAME	NUMBER			
	PDV	GHK		
CCLK	156	D19	O	CardBus clock. CCLK provides synchronous timing for all transactions on the CardBus interface. All signals except CRST, CCLKRUN, CINT, CSTSCHG, CAUDIO, CCD2, CCD1, CVS2, and CVS1 are sampled on the rising edge of CCLK, and all timing parameters are defined with the rising edge of this signal. CCLK operates at the PCI bus clock frequency, but it can be stopped in the low state or slowed down for power savings.
CCLKRUN	184	F10	I/O	CardBus clock run. CCLKRUN is used by a CardBus PC Card to request an increase in the CCLK frequency, and by the PCI4410 to indicate that the CCLK frequency is going to be decreased.
CRST	167	F12	O	CardBus reset. CRST brings CardBus PC Card-specific registers, sequencers, and signals to a known state. When CRST is asserted, all CardBus PC Card signals are placed in a high-impedance state, and the PCI4410 drives these signals to a valid logic level. Assertion can be asynchronous to CCLK, but deassertion must be synchronous to CCLK.

Table 2−14. CardBus PC Card Address and Data Terminals

TERMINAL NAME	TERMINAL NUMBER PDV	TERMINAL NUMBER GHK	I/O	DESCRIPTION
CAD31	192	C8	I/O	CardBus address and data. These signals make up the multiplexed CardBus address and data bus on the CardBus interface. During the address phase of a CardBus cycle, CAD31−CAD0 contain a 32-bit address. During the data phase of a CardBus cycle, CAD31−CAD0 contain data. CAD31 is the most significant bit.
CAD30	190	A8		
CAD29	189	E9		
CAD28	188	F9		
CAD27	187	C9		
CAD26	177	C11		
CAD25	176	B11		
CAD24	175	A11		
CAD23	172	C12		
CAD22	169	B13		
CAD21	168	C13		
CAD20	166	A14		
CAD19	164	B14		
CAD18	163	C14		
CAD17	161	B15		
CAD16	146	G14		
CAD15	144	G18		
CAD14	145	G17		
CAD13	141	H14		
CAD12	143	G19		
CAD11	140	H17		
CAD10	139	H18		
CAD9	137	J15		
CAD8	135	J17		
CAD7	133	J19		
CAD6	130	K17		
CAD5	131	K15		
CAD4	128	K19		
CAD3	129	K18		
CAD2	125	L17		
CAD1	127	L14		
CAD0	124	L18		
CC/$\overline{\text{BE3}}$	173	B12	I/O	CardBus bus commands and byte enables. CC/$\overline{\text{BE3}}$−CC/$\overline{\text{BE0}}$ are multiplexed on the same CardBus terminals. During the address phase of a CardBus cycle, CC/$\overline{\text{BE3}}$−CC/$\overline{\text{BE0}}$ define the bus command. During the data phase, this 4-bit bus is used as byte enables. The byte enables determine which byte paths of the full 32-bit data bus carry meaningful data. CC/$\overline{\text{BE0}}$ applies to byte 0 (CAD7−CAD0), CC/$\overline{\text{BE1}}$ applies to byte 1 (CAD15−CAD8), CC/$\overline{\text{BE2}}$ applies to byte 2 (CAD23−CAD16), and CC/$\overline{\text{BE3}}$ applies to byte 3 (CAD31−CAD24).
CC/$\overline{\text{BE2}}$	160	F13		
CC/$\overline{\text{BE1}}$	147	F19		
CC/$\overline{\text{BE0}}$	136	J14		
CPAR	149	G15	I/O	CardBus parity. In all CardBus read and write cycles, the PCI4410 calculates even parity across the CAD and CC/$\overline{\text{BE}}$ buses. As an initiator during CardBus cycles, the PCI4410 outputs CPAR with a one-CCLK delay. As a target during CardBus cycles, the calculated parity is compared to the initiator's parity indicator; a compare error results in a parity error assertion.

Table 2−15. CardBus PC Card Interface Control Terminals

TERMINAL			I/O	DESCRIPTION
NAME	NUMBER			
	PDV	GHK		
CAUDIO	182	C10	I	CardBus audio. CAUDIO is a digital input signal from a PC Card to the system speaker. The PCI4410 supports the binary audio mode and outputs a binary signal from the card to SPKROUT.
$\overline{\text{CBLOCK}}$	151	E19	I/O	CardBus lock. $\overline{\text{CBLOCK}}$ is used to gain exclusive access to a target.
$\overline{\text{CCD1}}$ $\overline{\text{CCD2}}$	123 185	L19 A9	I	CardBus detect 1 and CardBus detect 2. $\overline{\text{CCD1}}$ and $\overline{\text{CCD2}}$ are used in conjunction with CVS1 and CVS2 to identify card insertion and interrogate cards to determine the operating voltage and card type.
$\overline{\text{CDEVSEL}}$	155	E17	I/O	CardBus device select. The PCI4410 asserts $\overline{\text{CDEVSEL}}$ to claim a CardBus cycle as the target device. As a CardBus initiator on the bus, the PCI4410 monitors $\overline{\text{CDEVSEL}}$ until a target responds. If no target responds before timeout occurs, then the PCI4410 terminates the cycle with an initiator abort.
$\overline{\text{CFRAME}}$	159	E14	I/O	CardBus cycle frame. $\overline{\text{CFRAME}}$ is driven by the initiator of a CardBus bus cycle. $\overline{\text{CFRAME}}$ is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When $\overline{\text{CFRAME}}$ is deasserted, the CardBus bus transaction is in the final data phase.
$\overline{\text{CGNT}}$	154	F15	O	CardBus bus grant. $\overline{\text{CGNT}}$ is driven by the PCI4410 to grant a CardBus PC Card access to the CardBus bus after the current data transaction has been completed.
$\overline{\text{CINT}}$	180	A10	I	CardBus interrupt. $\overline{\text{CINT}}$ is asserted low by a CardBus PC Card to request interrupt servicing from the host.
$\overline{\text{CIRDY}}$	158	C15	I/O	CardBus initiator ready. $\overline{\text{CIRDY}}$ indicates the CardBus initiator's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of CCLK when both $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are asserted. Until $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are both sampled asserted, wait states are inserted.
$\overline{\text{CPERR}}$	152	F14	I/O	CardBus parity error. $\overline{\text{CPERR}}$ reports parity errors during CardBus transactions, except during special cycles. It is driven low by a target two clocks following that data when a parity error is detected.
$\overline{\text{CREQ}}$	171	E12	I	CardBus request. $\overline{\text{CREQ}}$ indicates to the arbiter that the CardBus PC Card desires use of the CardBus bus as an initiator.
$\overline{\text{CSERR}}$	181	B10	I	CardBus system error. $\overline{\text{CSERR}}$ reports address parity errors and other system errors that could lead to catastrophic results. $\overline{\text{CSERR}}$ is driven by the card synchronous to CCLK, but deasserted by a weak pullup, and may take several CCLK periods. The PCI4410 can report $\overline{\text{CSERR}}$ to the system by assertion of $\overline{\text{SERR}}$ on the PCI interface.
$\overline{\text{CSTOP}}$	153	E18	I/O	CardBus stop. $\overline{\text{CSTOP}}$ is driven by a CardBus target to request the initiator to stop the current CardBus transaction. $\overline{\text{CSTOP}}$ is used for target disconnects, and is commonly asserted by target devices that do not support burst data transfers.
CSTSCHG	183	E10	I	CardBus status change. CSTSCHG alerts the system to a change in the card's status, and is used as a wake-up mechanism.
$\overline{\text{CTRDY}}$	157	A16	I/O	CardBus target ready. $\overline{\text{CTRDY}}$ indicates the CardBus target's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of CCLK, when both $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are asserted; until this time, wait states are inserted.
CVS1 CVS2	179 165	F11 E13	I/O	CardBus voltage sense 1 and CardBus voltage sense 2. CVS1 and CVS2 are used in conjunction with $\overline{\text{CCD1}}$ and $\overline{\text{CCD2}}$ to identify card insertion and interrogate cards to determine the operating voltage and card type.

Table 2−16. IEEE1394 PHY/Link Interface Terminals

TERMINAL			I/O	FUNCTION
NAME	NUMBER			
	PDV	GHK		
PHY_CTL1 PHY_CTL0	198 197	F7 C7	I/O	PHY-link interface control. These bidirectional signals control passage of information between the PHY and link. The link can only drive these terminals after the PHY has granted permission following a link request (LREQ).
PHY_DATA7 PHY_DATA6 PHY_DATA5 PHY_DATA4 PHY_DATA3 PHY_DATA2 PHY_DATA1 PHY_DATA0	1 208 207 206 205 204 202 200	D1 A4 C5 E6 B5 F6 C6 B6	I/O	PHY-link interface data. These bidirectional signals pass data between the PHY and link. These terminals are driven by the link on transmissions and are driven by the PHY on receptions. Only DATA1−DATA0 are valid for 100 Mbit speed. DATA4−DATA0 are valid for 200 Mbit speed and DATA7−DATA0 are valid for 400 Mbit speed.
PHY_CLK	196	B7	I	System clock. This input provides a 49.152-MHz clock signal for data synchronization.
PHY_LREQ	194	E8	O	Link request. This signal is driven by the link to initiate a request for the PHY to perform some service.
LINKON	199	A6	I	1394 link on. This input from the PHY indicates that the link should turn on.
LPS	193	F8	O	Link power status. LPS indicates that link is powered and fully functional.

Table 2−17. Zoomed Video Interface Terminals

TERMINAL			I/O	FUNCTION
NAME	NUMBER			
	PDV	GHK		
ZV_HREF	95	W14	O	Horizontal sync to the zoomed video port
ZV_VSYNC	96	V14	O	Vertical sync to the zoomed video port
ZV_Y7 ZV_Y6 ZV_Y5 ZV_Y4 ZV_Y3 ZV_Y2 ZV_Y1 ZV_Y0	105 104 103 102 101 99 98 97	T19 W16 U15 R14 V15 W15 U14 R13	O	Video data to the zoomed video port in YUV:4:2:2 format
$\overline{\text{ZV\_UV7}}$ ZV_UV6 ZV_UV5 ZV_UV4 ZV_UV3 ZV_UV2 ZV_UV1 ZV_UV0	116 112 113 109 111 107 108 106	N17 P18 N15 R18 P17 P15 N14 R17	O	Video data to the zoomed video port in YUV:4:2:2 format
ZV_SCLK	114	P19	O	Audio SCLK PCM
ZV_MCLK	117	N18	O	Audio MCLK PCM
ZV_PCLK	120	M17	O	Pixel clock to the zoomed video port
ZV_LRCLK	118	N19	O	Audio LRCLK PCM
ZV_SDATA	119	M15	O	Audio SDATA PCM

3 Feature/Protocol Descriptions

The following sections give an overview of the PCI4410. Figure 3−1 shows connections to the PCI4410. The PCI interface includes all address/data and control signals for PCI protocol. The interrupt interface includes terminals for parallel PCI, parallel ISA, and serialized PCI and ISA signaling. Miscellaneous system interface terminals include multifunction terminals: $\overline{\text{SUSPEND}}$, $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ (power management control signal), and SPKROUT.

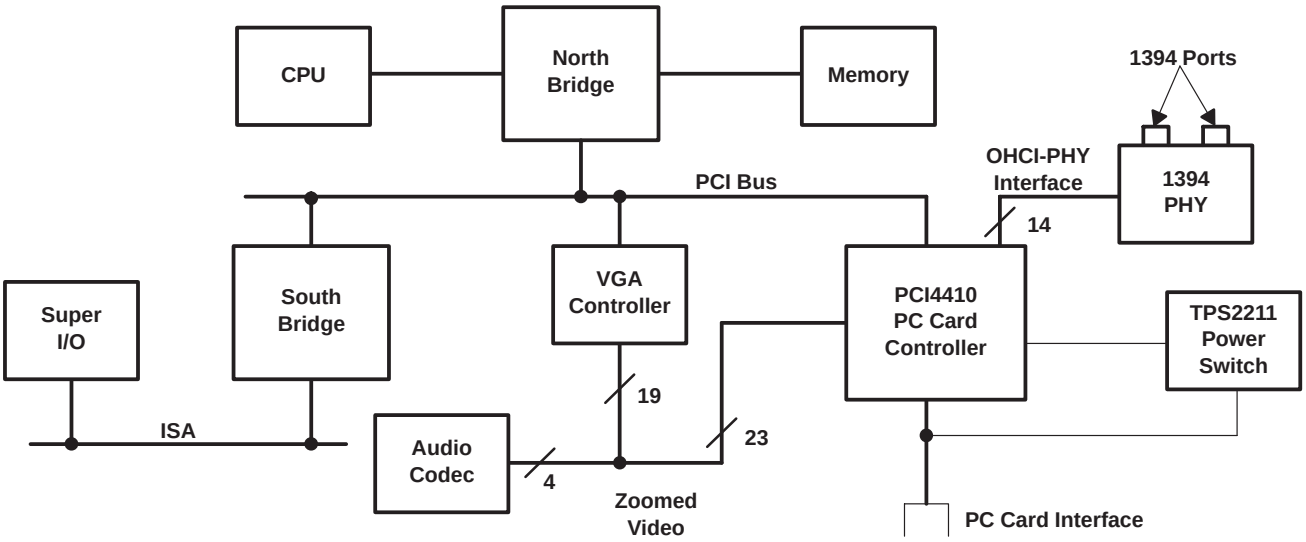


Figure 3−1. PCI4410 System Block Diagram

3.1 Power Supply Sequencing

The PCI4410 contains 3.3-V I/O buffers with 5-V tolerance requiring a core power supply and clamp voltages. The core power supply is always 3.3 V. The clamp voltages can be either 3.3 V or 5 V, depending on the interface. The following power-up and power-down sequences are recommended.

The power-up sequence is:

1. Apply 3.3-V power to the core.
2. Assert $\overline{\text{GRST}}$ to the device to disable the outputs during power-up. Output drivers must be powered up in the high-impedance state to prevent high current levels through the clamp diodes to the 5-V supply.
3. Apply the clamp voltage.

The power-down sequence is:

1. Use $\overline{\text{GRST}}$ to switch outputs to a high-impedance state.
2. Remove the clamp voltage.
3. Remove the 3.3-V power from the core.

3.2 I/O Characteristics

Figure 3−2 shows a 3-state bidirectional buffer. Section 10.2, *Recommended Operating Conditions*, provides the electrical characteristics of the inputs and outputs.

> **NOTE:** The PCI4410 meets the ac specifications of the *1997 PC Card Standard* and the *PCI Local Bus Specification.*

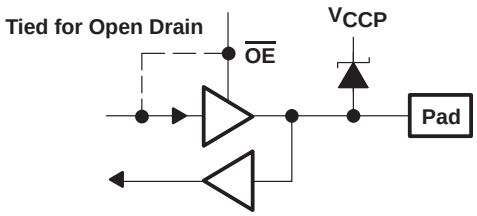


Figure 3−2. 3-State Bidirectional Buffer

NOTE: Unused pins (input or I/O) must be held high or low to prevent them from floating.

3.3 Clamping Voltages

The clamping voltages are set to match whatever external environment the PCI4410 is interfaced with: 3.3 V or 5 V. The I/O sites can be pulled through a clamping diode to a voltage rail that protects the core from external signals. The core power supply is always 3.3 V and is independent of the clamping voltages. For example, PCI signaling can be either 3.3 V or 5 V, and the PCI4410 must reliably accommodate both voltage levels. This is accomplished by using a 3.3-V I/O buffer that is 5-V tolerant, with the applicable clamping voltage applied. If a system designer desires a 5-V PCI bus, then V_{CCP} can be connected to a 5-V power supply.

The PCI4410 requires four separate clamping voltages because it supports a wide range of features. The four voltages are listed and defined in Section 10.2, *Recommended Operating Conditions*.

3.4 Peripheral Component Interconnect (PCI) Interface

The PCI4410 is fully compliant with the *PCI Local Bus Specification*. The PCI4410 provides all required signals for PCI master or slave operation, and may operate in either a 5-V or 3.3-V signaling environment by connecting the V_{CCP} terminals to the desired voltage level. In addition to the mandatory PCI signals, the PCI4410 provides the optional interrupt signal $\overline{\text{INTA}}$.

3.4.1 PCI Bus Lock ($\overline{\text{LOCK}}$)

The bus-locking protocol defined in the *PCI Local Bus Specification* is not highly recommended, but is provided on the PCI4410 as an additional compatibility feature. The PCI $\overline{\text{LOCK}}$ signal can be routed to the MFUNC4 terminal via the multifunction routing register. See Section 4.32, *Multifunction Routing Register*, for details. Note that the use of $\overline{\text{LOCK}}$ is only supported by PCI-to-CardBus bridges in the downstream direction (away from the processor).

PCI $\overline{\text{LOCK}}$ indicates an atomic operation that may require multiple transactions to complete. When $\overline{\text{LOCK}}$ is asserted, nonexclusive transactions can proceed to an address that is not currently locked. A grant to start a transaction on the PCI bus does not guarantee control of $\overline{\text{LOCK}}$; control of $\overline{\text{LOCK}}$ is obtained under its own protocol. It is possible for different initiators to use the PCI bus while a single master retains ownership of $\overline{\text{LOCK}}$. Note that the CardBus signal for this protocol is $\overline{\text{CBLOCK}}$ to avoid confusion with the bus clock.

An agent may need to do an exclusive operation because a critical access to memory might be broken into several transactions, but the master wants exclusive rights to a region of memory. The granularity of the lock is defined by PCI to be 16 bytes, aligned. The $\overline{\text{LOCK}}$ protocol defined by the *PCI Local Bus Specification* allows a resource lock without interfering with nonexclusive real-time data transfer, such as video.

The PCI bus arbiter may be designed to support only complete bus locks using the $\overline{\text{LOCK}}$ protocol. In this scenario, the arbiter will not grant the bus to any other agent (other than the $\overline{\text{LOCK}}$ master) while $\overline{\text{LOCK}}$ is asserted. A complete bus lock may have a significant impact on the performance of the video. The arbiter that supports complete bus lock must grant the bus to the cache to perform a writeback due to a snoop to a modified line when a locked operation is in progress.

The PCI4410 supports all $\overline{\text{LOCK}}$ protocol associated with PCI-to-PCI bridges, as also defined for PCI-to-CardBus bridges. This includes disabling write posting while a locked operation is in progress, which can solve a potential

deadlock when using devices such as PCI-to-PCI bridges. The potential deadlock can occur if a CardBus target supports delayed transactions and blocks access to the target until it completes a delayed read. This target characteristic is prohibited by the *PCI Local Bus Specification*, and the issue is resolved by the PCI master using $\overline{\text{LOCK}}$.

3.4.2 Loading Subsystem Identification

The subsystem vendor ID register (see Section 4.26) and subsystem ID register (see Section 4.27) make up a doubleword of PCI configuration space located at offset 40h for functions 0 and 1. This doubleword register is used for system and option card (mobile dock) identification purposes and is required by some operating systems. Implementation of this unique identifier register is a PC 99 requirement.

The PCI4410 offers two mechanisms to load a read-only value into the subsystem registers. The first mechanism relies upon the system BIOS providing the subsystem ID value. The default access mode to the subsystem registers is read-only, but can be made read/write by setting bit 5 (SUBSYSRW) in the system control register (see Section 4.29) at PCI offset 80h. When this bit is set, the BIOS can write a subsystem identification value into the registers at PCI offset 40h. The BIOS must clear the SUBSYSRW bit such that the subsystem vendor ID register and subsystem ID register are limited to read-only access. This approach saves the added cost of implementing the serial electrically erasable programmable ROM (EEPROM).

In some conditions, such as in a docking environment, the subsystem vendor ID register and subsystem ID register must be loaded with a unique identifier via a serial EEPROM. The PCI4410 loads the data from the serial EEPROM after a reset of the primary bus. Note that the $\overline{\text{SUSPEND}}$ input gates the PCI reset from the entire PCI4410 core, including the serial bus state machine (see Section 3.8.4, *Suspend Mode*, for details on using $\overline{\text{SUSPEND}}$).

The PCI4410 provides a two-line serial bus host controller that can interface to a serial EEPROM. See Section 3.6, *Serial Bus Interface*, for details on the two-wire serial bus controller and applications.

3.5 PC Card Applications

This section describes the PC Card interfaces of the PCI4410:

- Card insertion/removal and recognition
- P^2C power-switch interface
- Zoomed video support
- Speaker and audio applications
- LED socket activity indicators
- PC Card-16 DMA support
- PC Card controller programming model
- CardBus socket registers

3.5.1 PC Card Insertion/Removal and Recognition

The *1997 PC Card Standard* addresses the card-detection and recognition process through an interrogation procedure that the socket must initiate on card insertion into a cold, nonpowered socket. Through this interrogation, card voltage requirements and interface (16-bit versus CardBus) are determined.

The scheme uses the card detect and voltage sense signals. The configuration of these four terminals identifies the card type and voltage requirements of the PC Card interface. The encoding scheme is defined in the *1997 PC Card Standard* and in Table 3−1.

Table 3−1. PC Card Card-Detect and Voltage-Sense Connections

$\overline{CD2}$//$\overline{CCD2}$	$\overline{CD1}$//$\overline{CCD1}$	$\overline{VS2}$//CVS2	$\overline{VS1}$//CVS1	KEY	INTERFACE	VOLTAGE
Ground	Ground	Open	Open	5 V	16-bit PC Card	5 V
Ground	Ground	Open	Ground	5 V	16-bit PC Card	5 V and 3.3 V
Ground	Ground	Ground	Ground	5 V	16-bit PC Card	5 V, 3.3 V, and X.X V
Ground	Ground	Open	Ground	LV	16-bit PC Card	3.3 V
Ground	Connect to CVS1	Open	Connect to $\overline{CCD1}$	LV	CardBus PC Card	3.3 V
Ground	Ground	Ground	Ground	LV	16-bit PC Card	3.3 V and X.X V
Connect to CVS2	Ground	Connect to $\overline{CCD2}$	Ground	LV	CardBus PC Card	3.3 V and X.X V
Connect to CVS1	Ground	Ground	Connect to $\overline{CCD2}$	LV	CardBus PC Card	3.3 V, X.X V, and Y.Y V
Ground	Ground	Ground	Open	LV	16-bit PC Card	Y.Y V
Connect to CVS2	Ground	Connect to $\overline{CCD2}$	Open	LV	CardBus PC Card	Y.Y V
Ground	Connect to CVS2	Connect to $\overline{CCD1}$	Open	LV	CardBus PC Card	X.X V and Y.Y V
Connect to CVS1	Ground	Open	Connect to $\overline{CCD2}$	LV	CardBus PC Card	Y.Y V
Ground	Connect to CVS1	Ground	Connect to $\overline{CCD1}$	Reserved		
Ground	Connect to CVS2	Connect to $\overline{CCD1}$	Ground	Reserved		

3.5.2 P^{2}C Power-Switch Interface (TPS2211)

The PCI4410 provides a P^{2}C (PCMCIA peripheral control) interface for control of the PC Card power switch. The $\overline{VCCD}$ and VPPD terminals are used with the TI TPS2211 single-slot PC Card power interface switch to provide power switch support. Figure 3−3 shows terminal assignments for the TPS2211. Figure 3−4 illustrates a typical application, where the PCI4410 represents the PC Card controller.

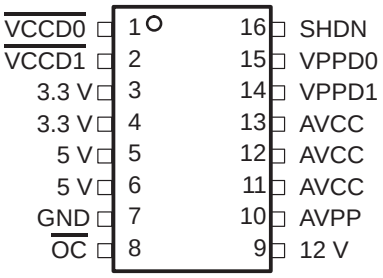


Figure 3−3. TPS2211 Terminal Assignments

The PCI4410 also includes support for the Maxim™ 1602 and Micrel MIC2562A single-channel CardBus power switches. Application of these power switches would be similar to that of the TPS2211.

Maxim is a trademark of Maxim Integrated Products, Inc.

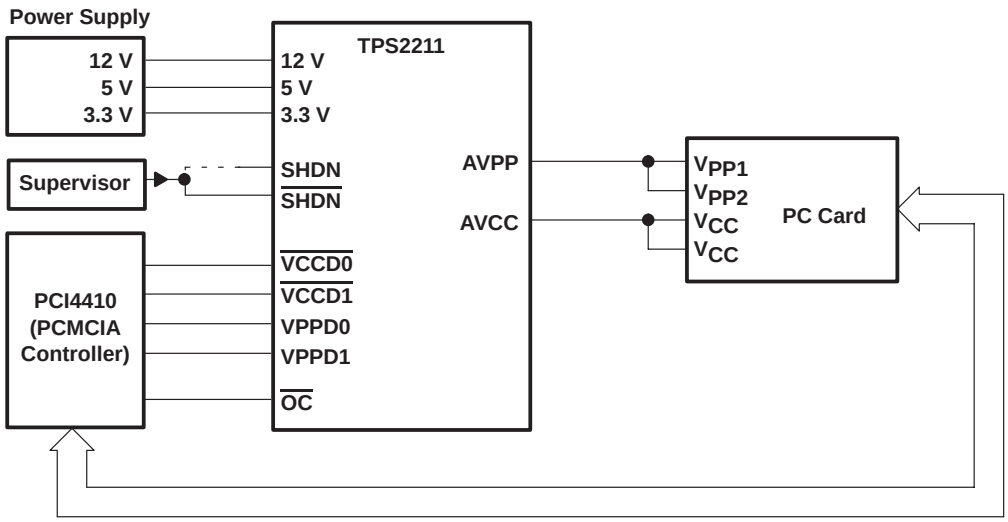


Figure 3−4. TPS2211 Typical Application

3.5.3 Zoomed Video Support

The zoomed video (ZV) port on the PCI4410 provides an internally buffered 16-bit ZV PC Card data path. This internal routing is programmed through the card control register (offset 91h, bits 5 and 6). Figure 3−5 summarizes the zoomed video subsystem implemented in the PCI4410, and details the bit functions found in the card control register.

When ZV PORT_ENABLE is enabled, the zoomed video output terminals are enabled and allow the PCI4410 to route the zoomed video data. However, no data is transmitted unless ZVENABLE (offset 91h, bit 6) is enabled. If ZVENABLE is set to low, then the ZV output port drives a logic 0 on the PCI4410's ZV bus.

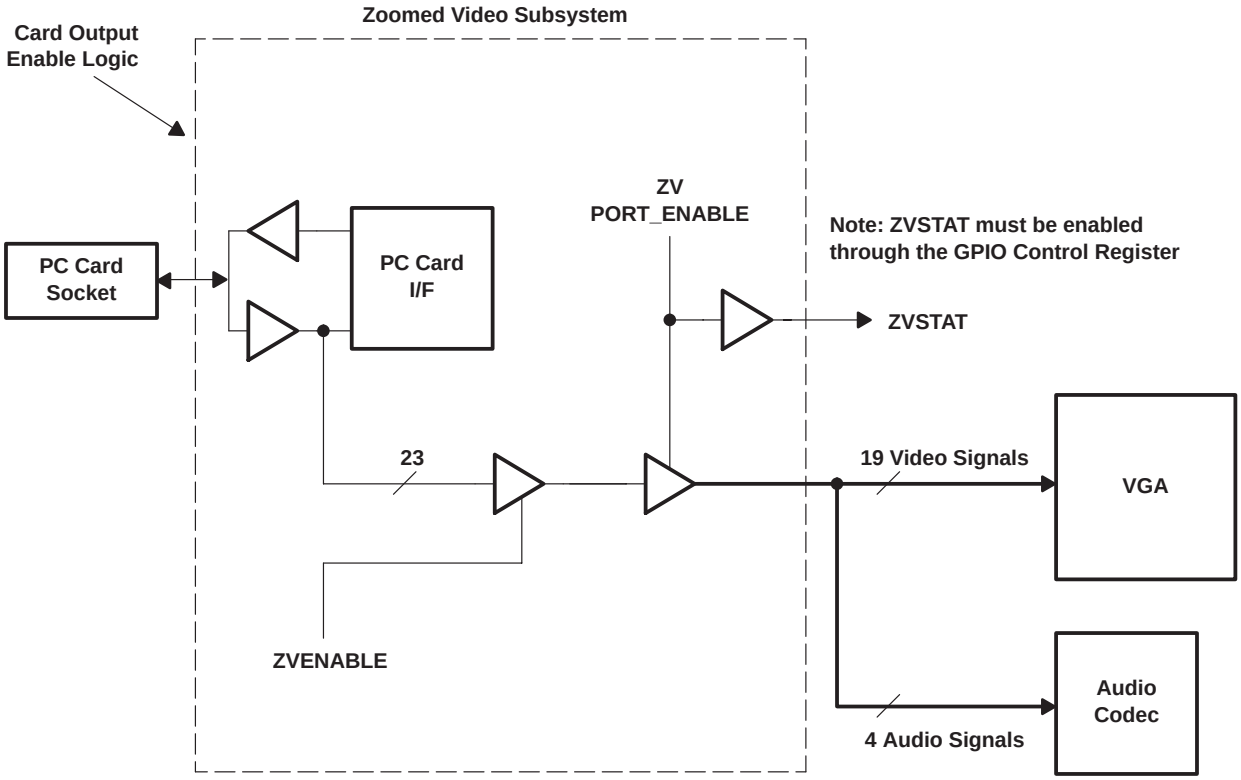


Figure 3−5. Zoomed Video Subsystem

3.5.4 Ultra Zoomed Video

Ultra zoomed video is an enhancement to the PCI4410 DMA engine and is intended to improve the 16-bit bandwidth for MPEG I and MPEG II decoder PC Cards. This enhancement allows the PCI4410 to fetch 32 bits of data from memory versus the 11XX/12XX 16-bit fetch capability. This enhancement allows a higher sustained throughput to the 16-bit PC Card because the PCI4410 prefetches an extra 16 bits (32 bits total) during each PCI read transaction. If the PCI bus becomes busy, then the PCI4410 has an extra 16 bits of data to perform back-to-back 16-bit transactions to the PC Card before having to fetch more data. This feature is built into the DMA engine and software is not required to enable this enhancement.

> **NOTE:** The 11XX and 12XX series CardBus controllers have enough 16-bit bandwidth to support MPEG II PC Card decoders. But it was decided to improve the bandwidth even more in the 14XX series CardBus controllers.

3.5.5 $\overline{\text{D3\_STAT}}$ Terminal

Additional functionality for the PCI4410 versus the 12xx series is the $\overline{\text{D3\_STAT}}$ (D3 status) pin. This pin is asserted under the following two conditions (both conditions must be true before $\overline{\text{D3\_STAT}}$ is asserted):

- Function 0 (PC Card controller) and function 1 (OHCI-Lynx) are both in D3.
- $\overline{\text{PME}}$ is enabled for either function.

3.5.6 Internal Ring Oscillator

The internal ring oscillator provides an internal clock source for the PCI4410 so that neither the PCI clock nor an external clock is required in order for the PCI4410 to power down a socket or interrogate a PC Card. This internal oscillator operates nominally at 16 kHz and can be enabled by setting bit 27 (P2CCLK) of the system control register (see Section 4.29) at PCI offset 80h to a 1. This function is disabled by default.

3.5.7 Integrated Pullup Resistors for PC Card Interface

The *1997 PC Card Standard* requires pullup resistors on various terminals to support both CardBus and 16-bit card configurations. Unlike the PCI1210/1211 which required external pullup resistors, the PCI4410 has integrated all of these pullup resistors on the terminalss below, except for the $\overline{\text{CCLKRUN}}$/WP($\overline{\text{IOIS16}}$) pullup resistor.

SIGNAL NAME	TERMINAL NUMBER	
	PDV	GHK
ADDR14/$\overline{\text{CPERR}}$	152	F14
ADDR15/$\overline{\text{CIRDY}}$	158	C15
ADDR19/$\overline{\text{CBLOCK}}$	151	E19
ADDR20/$\overline{\text{CSTOP}}$	153	E18
ADDR21/$\overline{\text{CDEVSEL}}$	155	E17
ADDR22/$\overline{\text{CTRDY}}$	157	A16
BVD1($\overline{\text{STSCHG}}$)/CSTSCHG	183	E10
BVD2($\overline{\text{SPKR}}$)/CAUDIO	182	C10
$\overline{\text{CD1}}$/$\overline{\text{CCD1}}$	123	L19
$\overline{\text{CD2}}$/$\overline{\text{CCD2}}$	185	A9
$\overline{\text{INPACK}}$/$\overline{\text{CREQ}}$	171	E12
READY/$\overline{\text{CINT}}$	180	A10
RESET/$\overline{\text{CRST}}$	167	F12
$\overline{\text{VS1}}$/CVS1	179	F11
$\overline{\text{VS2}}$/CVS2	165	E13
$\overline{\text{WAIT}}$/$\overline{\text{CSERR}}$	181	B10
WP($\overline{\text{IOIS16}}$)/$\overline{\text{CLKRUN}}$	184†	F10†

† This pin requires pullup, but the PCI1451 lacks an integrated pullup resistor.

3.5.8 SPKROUT and CAUDPWM Usage

SPKROUT carries the digital audio signal from the PC Card to the system. When a 16-bit PC Card is configured for I/O mode, the BVD2 pin becomes $\overline{\text{SPKR}}$. This terminal is also used in CardBus binary audio applications, and is referred to as CAUDIO. $\overline{\text{SPKR}}$ passes a TTL level digital audio signal to the PCI4410. The CardBus CAUDIO signal also can pass a single-amplitude binary waveform. The binary audio signals from the PC Card socket is used in the PCI4410 to produce SPKROUT. This output is enabled by bit 1 (SPKROUTEN) in the card control register (see Section 4.34).

Older controllers support CAUDIO in binary or PWM mode but use the same pin (SPKROUT). Some audio chips may not support both modes on one pin and may have a separate pin for binary and PWM. The PCI4410 implementation includes a signal for PWM, CAUDPWM, which can be routed to a MFUNC terminal. Bit 2 (AUD2MUX) located in the card control register is programmed to route a CardBus CAUDIO PWM terminal to CAUDPWM. See Section 4.32, *Multifunction Routing Register*, for details on configuring the MFUNC terminals.

Figure 3−6 illustrates a sample application using SPKROUT and CAUDPWM.

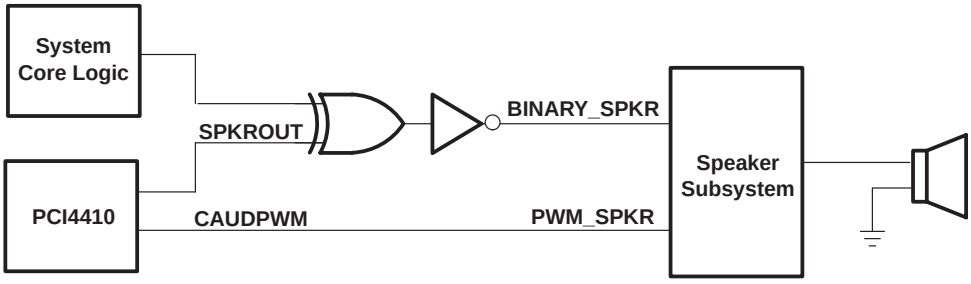


Figure 3−6. Sample Application of SPKROUT and CAUDPWM

3.5.9 LED Socket Activity Indicators

The socket activity LEDs are provided to indicate when a PC Card is being accessed. The LED_SKT signal can be routed to the multifunction terminals and is also provided on a dedicated pin (LED_SKT). When configured for LED

output, this terminal outputs an active high signal to indicate socket activity. See Section 4.32, *Multifunction Routing Register*, for details on configuring the multifunction terminals.

The LED signal is active high and is driven for 64-ms durations. When the LED is not being driven high, it is driven to a low state. Either of the two circuits shown in Figure 3−7 can be implemented to provide LED signaling. It is left for the board designer to implement the circuit that best fits the application.

The LED activity signals are valid when a card is inserted, powered, and not in reset. For PC Card-16, the LED activity signal is pulsed when READY/$\overline{\text{IREQ}}$ is low. For CardBus cards, the LED activity signal is pulsed if $\overline{\text{CFRAME}}$, $\overline{\text{CIRDY}}$, or $\overline{\text{CREQ}}$ is active.

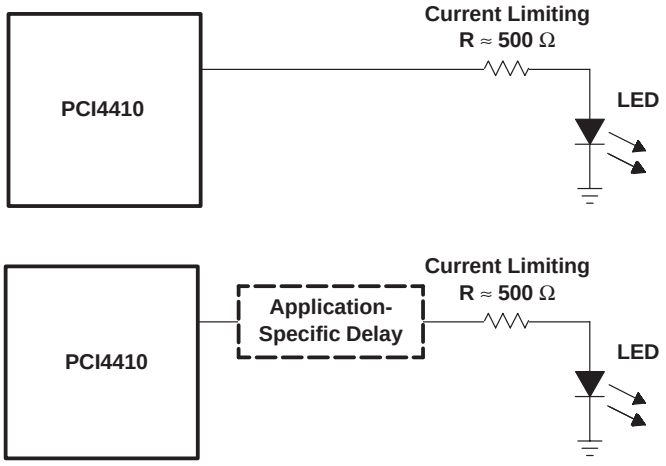


Figure 3−7. Two Sample LED Circuits

As indicated, the LED signals are driven for a period of 64 ms by a counter circuit. To avoid the possibility of the LED appearing to be stuck when the PCI clock is stopped, the LED signaling is cut off when the $\overline{\text{SUSPEND}}$ signal is asserted, when the PCI clock is to be stopped during the clock run protocol, or when in the D2 or D1 power state.

If any additional socket activity occurs during this counter cycle, then the counter is reset and the LED signal remains driven. If socket activity is frequent (at least once every 64 ms), then the LED signal remains driven.

3.5.10 PC Card-16 Distributed DMA Support

The PCI4410 supports a distributed DMA slave engine for 16-bit PC Card DMA support. The distributed DMA (DDMA) slave register set provides the programmability necessary for the slave DDMA engine. Table 3−2 provides the DDMA register configuration.

Two socket function dependent PCI configuration header registers that are critical for DDMA are the socket DMA register 0 (see Section 4.37) and the socket DMA register 1 (see Section 4.38). Distributed DMA is enabled through socket DMA register 0 and the contents of this register configure the PC Card-16 terminal ($\overline{\text{SPKR}}$, $\overline{\text{IOIS16}}$, or $\overline{\text{INPACK}}$) which is used for the DMA request signal, $\overline{\text{DREQ}}$. The base address of the DDMA slave registers and the transfer size (bytes or words) are programmed through the socket DMA register 1. See the programming model and register descriptions in Section 4 for details.

Table 3−2. Distributed DMA Registers

TYPE	REGISTER NAME				DDMA BASE ADDRESS OFFSET
R	Reserved	Page	Current address		00h
W			Base address		
R	Reserved	Reserved	Current count		04h
W			Base count		
R	N/A	Reserved	N/A	Status	08h
W	Mode		Request	Command	
R	Multichannel	Reserved	N/A	Reserved	0Ch
W	Mask		Master clear		

The DDMA registers contain control and status information consistent with the 8237 DMA controller; however, the register locations are reordered and expanded in some cases. While the DDMA register definitions are identical to those in the 8237 DMA controller of the same name, some register bits defined in the 8237 DMA controller do not apply to distributed DMA in a PCI environment. In such cases, the PCI4410 implements these obsolete register bits as read-only, nonfunctional bits. The reserved registers shown in Table 3−2 are implemented as read-only and return 0s when read. Write transactions to reserved registers have no effect.

The DDMA transfer is prefaced by several configuration steps that are specific to the PC Card and must be completed after the PC Card is inserted and interrogated. These steps include setting the proper $\overline{\text{DREQ}}$ signal assignment, setting the data transfer width, and mapping and enabling the DDMA register set. As discussed above, this is done through socket DMA register 0 and socket DMA register 1. The DMA register set is then programmed similarly to an 8237 controller, and the PCI4410 awaits a $\overline{\text{DREQ}}$ assertion from the PC Card requesting a DMA transfer.

DMA writes transfer data from the PC Card-to-PCI memory addresses. The PCI4410 accepts data 8 or 16 bits at a time, depending on the programmed data width, and then requests access to the PCI bus by asserting its $\overline{\text{REQ}}$ signal. Once the PCI bus is granted in an idle state, the PCI4410 initiates a PCI memory write command to the current memory address and transfers the data in a single data phase. After terminating the PCI cycle, the PCI4410 accepts the next byte(s) from the PC Card until the transfer count expires.

DMA reads transfer data from PCI memory addresses to the PC Card application. Upon the assertion of $\overline{\text{DREQ}}$, the PCI4410 asserts $\overline{\text{REQ}}$ to acquire the PCI bus. Once the bus is granted in an idle state, the PCI4410 initiates a PCI memory read operation to the current memory address and accepts 8 or 16 bits of data, depending on the programmed data width. After terminating the PCI cycle, the data is passed onto the PC Card. After terminating the PC Card cycle, the PCI4410 requests access to the PCI bus again until the transfer count has expired.

The PCI4410 target interface acts normally during this procedure and accepts I/O reads and writes to the DDMA registers. While a DDMA transfer is in progress and the host resets the DMA channel, the PCI4410 asserts TC and ends the PC Card cycle(s). TC is indicated in the DDMA status register (see Section 7.5). At the PC Card interface, the PCI4410 supports demand mode transfers. The PCI4410 asserts DACK during the transfer unless $\overline{\text{DREQ}}$ is deasserted before TC. TC is mapped to the $\overline{\text{OE}}$ PC Card terminal for DMA write operations and is mapped to the $\overline{\text{WE}}$ PC Card terminal for DMA read operations. The DACK signal is mapped to the PC Card $\overline{\text{REG}}$ signal in all transfers, and the $\overline{\text{DREQ}}$ terminal is routed to one of three options which is programmed through socket DMA register 0.

3.5.11 PC Card-16 PC/PCI DMA

Some chip sets provide a way for legacy I/O devices to do DMA transfers on the PCI bus. In the PC/PCI DMA protocol, the PCI4410 acts as a PCI target device to certain DMA related I/O addresses. The PCI4410 $\overline{\text{PCREQ}}$ and $\overline{\text{PCGNT}}$ signals are provided as a point-to-point connection to a chipset supporting PC/PCI DMA. The $\overline{\text{PCREQ}}$ and $\overline{\text{PCGNT}}$ signals may be routed to the MFUNC2 and MFUNC5 terminals, respectively. See Section 4.32, *Multifunction Routing Register*, for details on configuring the multifunction terminals.

Under the PC/PCI protocol, a PCI DMA slave device (such as the PCI4410) requests a DMA transfer on a particular channel using a serialized protocol on $\overline{\text{PCREQ}}$. The I/O DMA bus master arbitrates for the PCI bus and grants the channel through a serialized protocol on $\overline{\text{PCGNT}}$ when it is ready for the transfer. The I/O cycle and memory cycles are then presented on the PCI bus, which performs the DMA transfers similarly to legacy DMA master devices.

PC/PCI DMA is enabled for each PC Card-16 slot by setting bit 19 (CDREQEN) in the respective system control register (see Section 4.29). On power up this bit is reset and the card PC/PCI DMA is disabled. Bit 3 (CDMA_EN) of the system control register is a global enable for PC/PCI DMA, and is set at power up and never cleared if the PC/PCI DMA mechanism is implemented. The desired DMA channel for each PC Card-16 slot must be configured through bits 18−16 (CDMACHAN field) in the system control register. The channels are configured as indicated in Table 3−3.

Table 3−3. PC/PCI Channel Assignments

SYSTEM CONTROL REGISTER			DMA CHANNEL	CHANNEL TRANSFER DATA WIDTH
BIT 18	BIT 17	BIT16		
0	0	0	Channel 0	8-bit DMA transfers
0	0	1	Channel 1	8-bit DMA transfers
0	1	0	Channel 2	8-bit DMA transfers
0	1	1	Channel 3	8-bit DMA transfers
1	0	0	Channel 4	Not used
1	0	1	Channel 5	16-bit DMA transfers
1	1	0	Channel 6	16-bit DMA transfers
1	1	1	Channel 7	16-bit DMA transfers

As in distributed DMA, the PC Card terminal mapped to $\overline{\text{DREQ}}$ must be configured through socket DMA register 0 (see Section 4.37). The data transfer width is a function of channel number and the DDMA slave registers are not used. When a $\overline{\text{DREQ}}$ is received from a PC Card and the channel has been granted, the PCI4410 decodes the I/O addresses listed in Table 3−4 and performs actions dependent upon the address.

Table 3−4. I/O Addresses Used for PC/PCI DMA

DMA I/O ADDRESS	DMA CYCLE TYPE	TERMINAL COUNT	PCI CYCLE TYPE
00h	Normal	0	I/O read/write
04h	Normal TC	1	I/O read/write
C0h	Verify	0	I/O read
C4h	Verify TC	1	I/O read

When the PC/PCI DMA is used as a PC Card-16 DMA mechanism, it may not provide the performance levels of DDMA; however, the design of a PCI target implementing PC/PCI DMA is considerably less complex. No bus master state machine is required to support PC/PCI DMA, because the DMA control is centralized in the chipset. This DMA scheme is often referred to as centralized DMA for this reason.

3.5.12 CardBus Socket Registers

The PCI4410 contains all registers for compatibility with the *PC Card Standard*, release 7. These registers exist as the CardBus socket registers and are listed in Table 3−5.

Table 3−5. CardBus Socket Registers

REGISTER NAME	OFFSET
Socket event	00h
Socket mask	04h
Socket present state	08h
Socket force event	0Ch
Socket control	10h
Reserved	14h
Reserved	18h
Reserved	1Ch
Socket power management	20h

3.6 Serial Bus Interface

The PCI4410 provides a serial bus interface to load subsystem identification and select register defaults through a serial EEPROM and to provide a PC Card power switch interface alternative to P^{2}C. See Section 3.5.2, *P^{2}C Power-Switch Interface (TPS2211)*, for details. The PCI4410 serial bus interface is compatible with various I^{2}C and SMBus components.

3.6.1 Serial Bus Interface Implementation

The PCI4410 defaults to the serial bus interface are disabled. To enable the serial interface, a pullup resistor must be implemented on the $\overline{\text{VCCD0}}$ and $\overline{\text{VCCD1}}$ terminals and the appropriate pullup resistors must be implemented on the SDA and SCL signals, that is, the MFUNC1 and MFUNC4 terminals.

The PCI4410 implements a two-pin serial interface with one clock signal (SCL) and one data signal (SDA). When pullup resistors are provided on the VPPD0 and VPPD1 terminals, the SCL signal is mapped to the MFUNC4 terminal and the SDA signal is mapped to the MFUNC1 terminal. The PCI4410 drives SCL at nearly 100 kHz during data transfers, which is the maximum specified frequency for standard mode I^{2}C. The serial EEPROM must be located at address A0h. Figure 3−8 illustrates an example application implementing the two-wire serial bus.

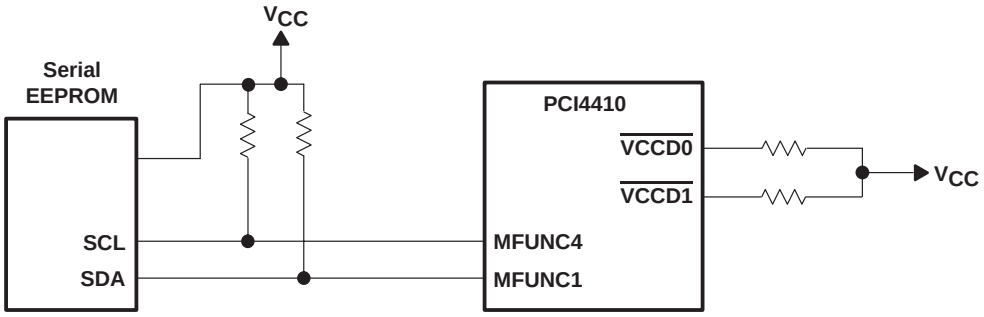


Figure 3−8. Serial EEPROM Application

Some serial device applications may include PC Card power switches, ZV source switches, card ejectors, or other devices that may enhance the user's PC Card experience. The serial EEPROM device and PC Card power switches are discussed in the sections that follow.

3.6.2 Serial Bus Interface Protocol

The SCL and SDA signals are bidirectional, open-drain signals and require pullup resistors as shown in Figure 3−8. The PCI4410 supports up to 100 Kb/s data transfer rate and is compatible with standard mode I^{2}C using 7-bit addressing.

All data transfers are initiated by the serial bus master. The beginning of a data transfer is indicated by a start condition, which is signalled when the SDA line transitions to a low state while SCL is in the high state, as illustrated

in Figure 3−9. The end of a requested data transfer is indicated by a stop condition, which is signaled by a low-to-high transition of SDA while SCL is in the high state, as shown in Figure 3−9. Data on SDA must remain stable during the high state of the SCL signal, as changes on the SDA signal during the high state of SCL are interpreted as control signals, that is, a start or a stop condition.

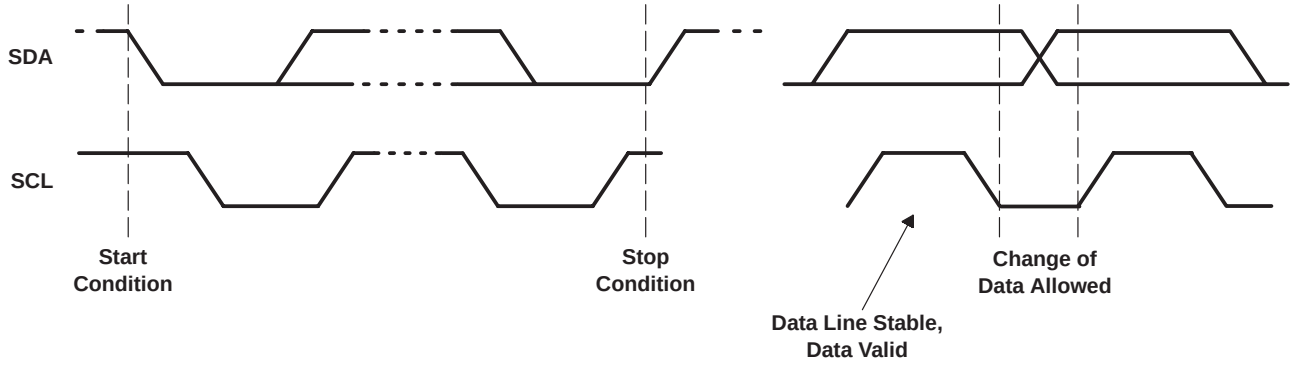


Figure 3−9. Serial Bus Start/Stop Conditions and Bit Transfers

Data is transferred serially in 8-bit bytes. The number of bytes that may be transmitted during a data transfer is unlimited; however, each byte must be completed with an acknowledge bit. An acknowledge (ACK) is indicated by the receiver pulling the SDA signal low so that it remains low during the high state of the SCL signal. Figure 3−10 illustrates the acknowledge protocol.

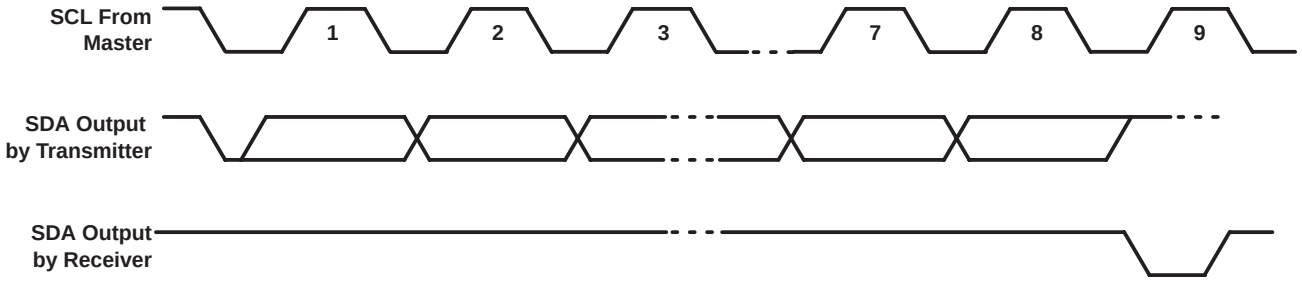


Figure 3−10. Serial Bus Protocol Acknowledge

The PCI4410 is a serial bus master; all other devices connected to the serial bus external to the PCI4410 are slave devices. As the bus master, the PCI4410 drives the SCL clock at nearly 100 kHz during bus cycles and places SCL in a high-impedance state (zero frequency) during idle states.

Typically, the PCI4410 masters byte reads and byte writes under software control. Doubleword reads are performed by the serial EEPROM initialization circuitry upon a PCI reset and may not be generated under software control. See Section 3.6.3, *Serial Bus EEPROM Application*, for details on how the PCI4410 automatically loads the subsystem identification and other register defaults through a serial bus EEPROM.

Figure 3−11 illustrates a byte write. The PCI4410 issues a start condition and sends the 7-bit slave device address and the command bit zero. A 0 in the R/$\overline{W}$ command bit indicates that the data transfer is a write. The slave device acknowledges if it recognizes the address. The word address byte is then sent by the PCI4410 and another slave acknowledgment is expected. Then the PCI4410 delivers the data byte MSB first and expects a final acknowledgment before issuing the stop condition.

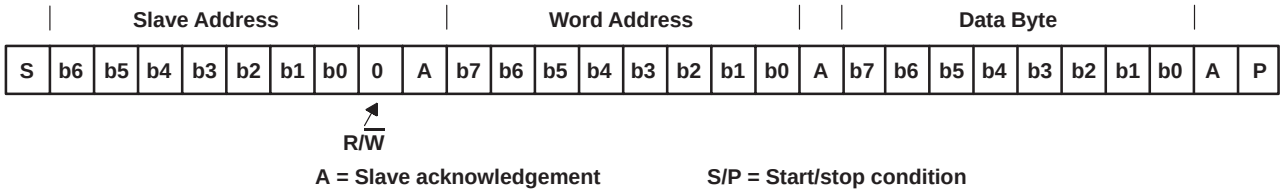


Figure 3−11. Serial Bus Protocol − Byte Write

Figure 3−12 illustrates a byte read. The read protocol is very similar to the write protocol except the R/$\overline{W}$ command bit must be set to 1 to indicate a read-data transfer. In addition, the PCI4410 master must acknowledge reception of the read bytes from the slave transmitter. The slave transmitter drives the SDA signal during read data transfers. The SCL signal remains driven by the PCI4410 master.

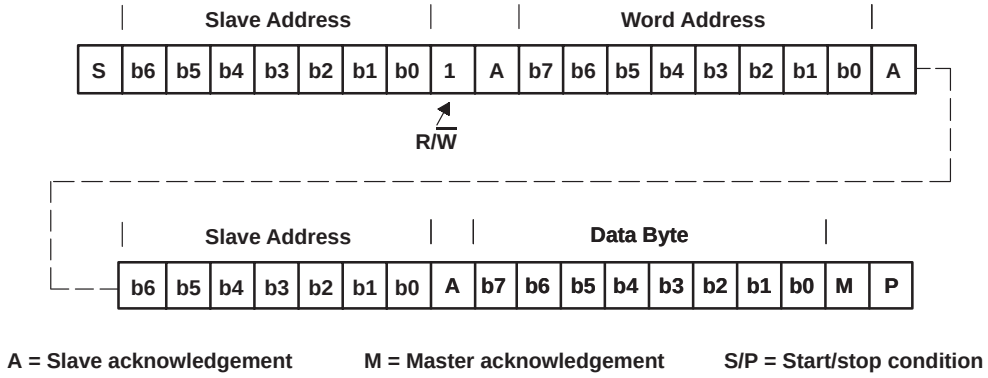


Figure 3−12. Serial Bus Protocol − Byte Read

Figure 3−13 illustrates EEPROM interface doubleword data collection protocol.

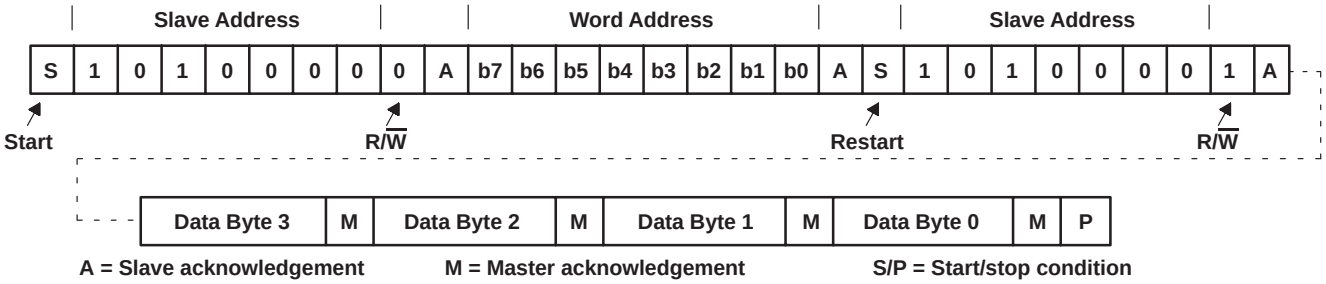


Figure 3−13. EEPROM Interface Doubleword Data Collection

3.6.3 Serial Bus EEPROM Application

When the PCI bus is reset and the serial bus interface is detected, the PCI4410 attempts to read the subsystem identification and other register defaults from a serial EEPROM. The registers and corresponding bits that may be loaded with defaults through the EEPROM are provided in Table 3−6.

Table 3−6. Registers and Bits Loadable Through Serial EEPROM

OHCI REGISTERS LOADED			
OFFSET REFERENCE	**REGISTER**	**REGISTER NAME**	**BITS LOADED FROM EEPROM**
0	3Eh	MIN_GNT and MAX_LAT (see Section 8.14)	Byte 0, bits 3−0
1	3Fh	MIN_GNT and MAX_LAT (see Section 8.14)	Byte 1, bits 3−0
2	PCI 2Ch	Subsystem identification (see Section 8.11)	Byte 0
3	PCI 2Ch	Subsystem identification (see Section 8.11)	Byte 1
4	PCI 2Ch	Subsystem identification (see Section 8.11)	Byte 2
5	PCI 2Ch	Subsystem identification (see Section 8.11)	Byte 3
6	PCI F4h	Link enhancement control (see Section 8.21)	Byte 0, bits 7, 2, 1
7		Mini-ROM address	
8	PCI 24h	GUID high (see Section 9.10)	Byte 0
9	PCI 24h	GUID high (see Section 9.10)	Byte 1
10	PCI 24h	GUID high (see Section 9.10)	Byte 2
11	PCI 24h	GUID high (see Section 9.10)	Byte 3
12	PCI 28h	GUID low (see Section 9.11	Byte 0
13	PCI 28h	GUID low (see Section 9.11)	Byte 1
14	PCI 28h	GUID low (see Section 9.11)	Byte 2
15	PCI 28h	GUID low (see Section 9.11)	Byte 3
16		Checksum	
17	PCI F4h	Link enhancement control (see Section 8.21)	Byte 1, bits 5, 4, 1, 0
18	PCI F0h	Miscellaneous configuration (see Section 8.20)	Byte 0, bits 4, 2−0
19	PCI F0h	Miscellaneous configuration (see Section 8.20)	Byte 1, bits 7, 5, 2
CARDBUS REGISTERS LOADED			
OFFSET REFERENCE	**REGISTER**	**REGISTER NAME**	**BITS LOADED FROM EEPROM**
0		Flag byte	
1	PCI 40h	Subsystem vendor ID (see Section 4.26)	Byte 0
2	PCI 40h	Subsystem vendor ID (see Section 4.26)	Byte 1
3	PCI 42h	Subsystem ID (see Section 4.27)	Byte 0
4	PCI 42h	Subsystem ID (see Section 4.27)	Byte 1
5	PCI 80h	System control (see Section 4.29)	Byte 0
6	PCI 80h	System control (see Section 4.29)	Byte 1, bits 7, 6
7	PCI 80h	System control (see Section 4.29)	Byte 3, bits 7, 5, 3, 2, 0
8	PCI 86h	General control (see Sectin 4.31)	Bits 3, 1, 0
9	PCI 8Ch	Multifunction routing (see Section 4.32)	Byte 0
10	PCI 8Ch	Multifunction routing (see Section 4.32)	Byte 1
11	PCI 8Ch	Multifunction routing (see Section 4.32)	Byte 2
12	PCI 8Ch	Multifunction routing (see Section 4.32)	Byte 3, bits 3−0
13	PCI 90h	Retry status (see Section 4.33)	Bits 7, 6
14	PCI 91h	Card control (see Section 4.34)	Bit 7
15	PCI 92h	Device control (see Section 4.35)	Bits 6−0
16	PCI 93h	Diagnostic (see Section 4.36)	Bits 7, 4−0
17	PCI A2h	Power management capabilities (see Section 4.41)	Bit 15
18	ExCA 00h	ExCA Identification and revision (see Section 5.1)	Bits 7−0

Figure 3−14 details the EEPROM data format. This format must be followed for the PCI4410 to properly load initializations from a serial EEPROM.

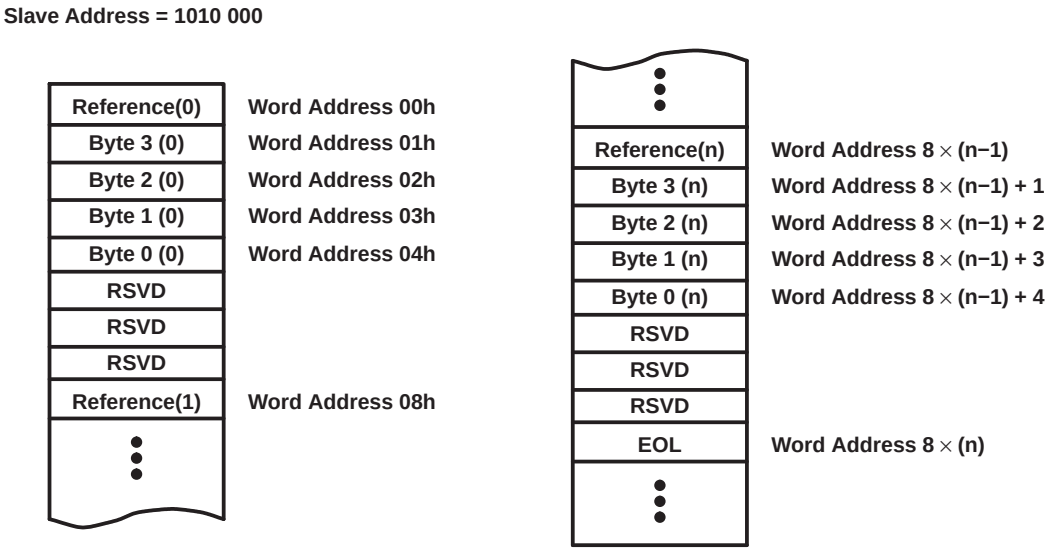


Figure 3−14. EEPROM Data Format

The byte at the EEPROM word address 00h must either contain a valid offset reference, as listed in Table 3−6, or an end-of-list (EOL) indicator. The EOL indicator is a byte value of FFh, and indicates the end of the data to load from the EEPROM. Only doubleword registers are loaded from the EEPROM, and all bit fields must be considered when the EEPROM is programmed.

The serial EEPROM is addressed at slave address 1010000b by the PCI4410. All hardware address bits for the EEPROM should be tied to the appropriate level to achieve this address. The serial EEPROM chip in the sample application circuit (see Figure 3−8) assumes the 1010b high address nibble. The lower three address bits are terminal inputs to the chip, and the sample application shows these terminal inputs tied to GND.

When a valid offset reference is read, four bytes are read from the EEPROM, MSB first, as illustrated in Figure 3−13. The address autoincrements after every byte transfer according to the doubleword read protocol. Note that the word addresses align with the data format illustrated in Figure 3−14. The PCI4410 continues to load data from the serial EEPROM until an end-of-list indicator is read. Three reserved bytes are stuffed to maintain eight-byte data structures.

Note, the eight-byte data structure is important to provide correct addressing per the doubleword read format shown in Figure 3−13. In addition, the reference offsets must be loaded in the EEPROM in sequential order, that is, 01h, 02h, 03h, 04h. If the offsets are not sequential, then the registers may be loaded incorrectly.

3.6.4 Accessing Serial Bus Devices Through Software

The PCI4410 provides a programming mechanism to control serial bus devices through software. The programming is accomplished through a doubleword of PCI configuration space at offset B0h.

3.7 Programmable Interrupt Subsystem

Interrupts provide a way for I/O devices to let the microprocessor know that they require servicing. The dynamic nature of PC Cards and the abundance of PC Card I/O applications require substantial interrupt support from the PCI4410. The PCI4410 provides several interrupt signaling schemes to accommodate the needs of a variety of platforms. The different mechanisms for dealing with interrupts in this device are based on various specifications and industry standards. The ExCA register set provides interrupt control for some 16-bit PC Card functions, and the CardBus socket register set provides interrupt control for the CardBus PC Card functions. The PCI4410 is, therefore, backward compatible with existing interrupt control register definitions, and new registers have been defined where required.

The PCI4410 detects PC Card interrupts and events at the PC Card interface and notifies the host controller using one of several interrupt signaling protocols. To simplify the discussion of interrupts in the PCI4410, PC Card interrupts are classified as either card status change (CSC) or as functional interrupts.

The method by which any type of PCI4410 interrupt is communicated to the host interrupt controller varies from system to system. The PCI4410 offers system designers the choice of using parallel PCI interrupt signaling, parallel ISA-type IRQ interrupt signaling, or the IRQSER serialized ISA and/or PCI interrupt protocol. It is possible to use the parallel PCI interrupts in combination with either parallel IRQs or serialized IRQs, as detailed in the sections that follow. All interrupt signaling is provided through the seven multifunction terminals, MFUNC0−MFUNC6. In addition, PCI interrupts ($\overline{\text{INTA}}$ and $\overline{\text{INTB}}$) are available on dedicated pins.

3.7.1 PC Card Functional and Card Status Change Interrupts

PC Card functional interrupts are defined as requests from a PC Card application for interrupt service and are indicated by asserting specially defined signals on the PC Card interface. Functional interrupts are generated by 16-bit I/O PC Cards and by CardBus PC Cards.

Card status change (CSC)-type interrupts are defined as events at the PC Card interface that are detected by the PCI4410 and may warrant notification of host card and socket services software for service. CSC events include both card insertion and removal from PC Card sockets, as well as transitions of certain PC Card signals.

Table 3−7 summarizes the sources of PC Card interrupts and the type of card associated with them. CSC and functional interrupt sources are dependent on the type of card inserted in the PC Card socket. The three types of cards that can be inserted into any PC Card socket are:

- 16-bit memory card
- 16-bit I/O card
- CardBus cards

Table 3−7. Interrupt Mask and Flag Registers

CARD TYPE	EVENT	MASK	FLAG
16-bit memory	Battery conditions (BVD1, BVD2)	ExCA offset 05h/805h bits 1 and 0	ExCA offset 04h/804h bits 1 and 0
	Wait states (READY)	ExCA offset 05h/805h bit 2	ExCA offset 04h/804h bit 2
16-bit I/O	Change in card status ($\overline{\text{STSCHG}}$)	ExCA offset 05h/805h bit 0	ExCA offset 04h/804h bit 0
	Interrupt request ($\overline{\text{IREQ}}$)	Always enabled	PCI configuration offset 91h bit 0
All 16-bit PC Cards	Power cycle complete	ExCA offset 05h/805h bit 3	ExCA offset 04h/804h bit 3
CardBus	Change in card status (CSTSCHG)	Socket mask bit 0	Socket event bit 0
	Interrupt request ($\overline{\text{CINT}}$)	Always enabled	PCI configuration offset 91h bit 0
	Power cycle complete	Socket mask bit 3	Socket event bit 3
	Card insertion or removal	Socket mask bits 2 and 1	Socket event bits 2 and 1

Functional interrupt events are valid only for 16-bit I/O and CardBus cards; that is, the functional interrupts are not valid for 16-bit memory cards. Furthermore, card insertion and removal-type CSC interrupts are independent of the card type. Table 3−8 describes the PC Card interrupt events.

Table 3−8. PC Card Interrupt Events and Description

CARD TYPE	EVENT	TYPE	SIGNAL	DESCRIPTION
16-bit memory	Battery conditions (BVD1, BVD2)	CSC	BVD1($\overline{\text{STSCHG}}$)//CSTSCHG	A transition on BVD1 indicates a change in the PC Card battery conditions.
			BVD2($\overline{\text{SPKR}}$)//CAUDIO	A transition on BVD2 indicates a change in the PC Card battery conditions.
	Wait states (READY)	CSC	READY($\overline{\text{IREQ}}$)//$\overline{\text{CINT}}$	A transition on READY indicates a change in the ability of the memory PC Card to accept or provide data.
16-bit I/O	Change in card status (STSCHG)	CSC	BVD1($\overline{\text{STSCHG}}$)//CSTSCHG	The assertion of $\overline{\text{STSCHG}}$ indicates a status change on the PC Card.
	Interrupt request (IREQ)	Functional	READY($\overline{\text{IREQ}}$)//$\overline{\text{CINT}}$	The assertion of $\overline{\text{IREQ}}$ indicates an interrupt request from the PC Card.
CardBus	Change in card status (CSTSCHG)	CSC	BVD1($\overline{\text{STSCHG}}$)//CSTSCHG	The assertion of CSTSCHG indicates a status change on the PC Card.
	Interrupt request ($\overline{\text{CINT}}$)	Functional	READY($\overline{\text{IREQ}}$)//$\overline{\text{CINT}}$	The assertion of $\overline{\text{CINT}}$ indicates an interrupt request from the PC Card.
All PC Cards	Card insertion or removal	CSC	$\overline{\text{CD1}}$//$\overline{\text{CCD1}}$, $\overline{\text{CD2}}$//$\overline{\text{CCD2}}$	A transition on either $\overline{\text{CD1}}$//$\overline{\text{CCD1}}$ or $\overline{\text{CD2}}$//$\overline{\text{CCD2}}$ indicates an insertion or removal of a 16-bit or CardBus PC Card.
	Power cycle complete	CSC	N/A	An interrupt is generated when a PC Card power-up cycle has completed.

The naming convention for PC Card signals describes the function for 16-bit memory, I/O cards, and CardBus. For example, READY($\overline{\text{IREQ}}$)//$\overline{\text{CINT}}$ includes READY for 16-bit memory cards, $\overline{\text{IREQ}}$ for 16-bit I/O cards, and $\overline{\text{CINT}}$ for CardBus cards. The 16-bit memory card signal name is first, with the I/O card signal name second, enclosed in parentheses. The CardBus signal name follows after a forward double slash (//).

The *1997 PC Card Standard* describes the power-up sequence that must be followed by the PCI4410 when an insertion event occurs and the host requests that the socket V_{CC} and V_{PP} be powered. Upon completion of this power-up sequence, the PCI4410 interrupt scheme can be used to notify the host system (see Table 3−8), denoted by the power cycle complete event. This interrupt source is considered a PCI4410 internal event because it depends on the completion of applying power to the socket rather than on a signal change at the PC Card interface.

3.7.2 Interrupt Masks and Flags

Host software may individually mask (or disable) most of the potential interrupt sources listed in Table 3−8 by setting the appropriate bits in the PCI4410. By individually masking the interrupt sources listed, software can control those events that cause a PCI4410 interrupt. Host software has some control over the system interrupt the PCI4410 asserts by programming the appropriate routing registers. The PCI4410 allows host software to route PC Card CSC and PC Card functional interrupts to separate system interrupts. Interrupt routing somewhat specific to the interrupt signaling method used is discussed in more detail in the following sections.

When an interrupt is signaled by the PCI4410, the interrupt service routine must determine which of the events listed in Table 3−7 caused the interrupt. Internal registers in the PCI4410 provide flags that report the source of an interrupt. By reading these status bits, the interrupt service routine can determine the action to be taken.

Table 3−7 details the registers and bits associated with masking and reporting potential interrupts. All interrupts can be masked except the functional PC Card interrupts, and an interrupt status flag is available for all types of interrupts.

Notice that there is not a mask bit to stop the PCI4410 from passing PC Card functional interrupts through to the appropriate interrupt scheme. These interrupts are not valid until the card is properly powered, and there should never be a card interrupt that does not require service after proper initialization.

Table 3−7 lists the various methods of clearing the interrupt flag bits. The flag bits in the ExCA registers (16-bit PC Card-related interrupt flags) can be cleared using two different methods. One method is an explicit write of 1 to the flag bit to clear and the other is by reading the flag bit register. The selection of flag bit clearing is made by bit 2 (IFCMODE) in the ExCA global control register (see Section 5.22), located at ExCA offset 1Eh/5Eh/81Eh, and defaults to the *flag cleared on read* method.

The CardBus-related interrupt flags can be cleared by an explicit write of 1 to the interrupt flag in the socket event register (see Section 6.1). Although some of the functionality is shared between the CardBus registers and the ExCA registers, software should not program the chip through both register sets when a CardBus card is functioning.

3.7.3 Using Parallel IRQ Interrupts

The seven multifunction terminals, MFUNC6−MFUNC0, implemented in the PCI4410 may be routed to obtain a subset of the ISA IRQs. The IRQ choices provide ultimate flexibility in PC Card host interruptions. To use the parallel ISA type IRQ interrupt signaling, software must program the device control register (see Section 4.35), located at PCI offset 92h, to select the parallel IRQ signaling scheme. See Section 4.32, *Multifunction Routing Register*, for details on configuring the multifunction terminals.

A system using parallel IRQs requires a minimum of one PCI terminal, $\overline{\text{INTA}}$, to signal CSC events. This requirement is dictated by certain card and socket services software. The MFUNC pins provide (at a maximum) seven different IRQs to support legacy 16-bit PC Card functions.

As an example, suppose the seven IRQs used by legacy PC Card applications are IRQ3, IRQ4, IRQ5, IRQ9, IRQ10, IRQ11, and IRQ15. The multifunction routing register must be programmed to a value of 0x0FBA5439. This routes the MFUNC terminals as illustrated in Figure 3−15. Not shown is that $\overline{\text{INTA}}$ must also be routed to the programmable interrupt controller (PIC), or to some circuitry that provides parallel PCI interrupts to the host.

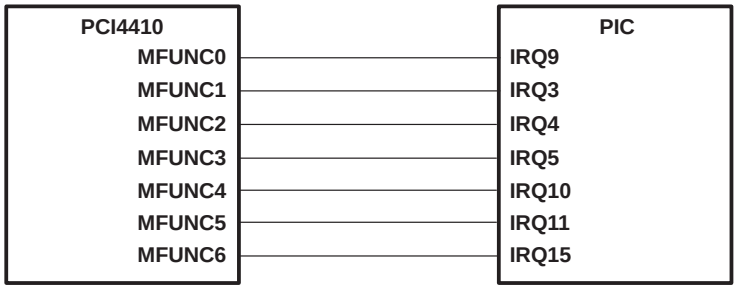


Figure 3−15. IRQ Implementation

Power-on software is responsible for programming the multifunction routing register to reflect the IRQ configuration of a system implementing the PCI4410. See Section 4.32, *Multifunction Routing Register*, for details on configuring the multifunction terminals.

The parallel ISA type IRQ signaling from the MFUNC6−MFUNC0 terminals is compatible with those input directly into the 8259 PIC. The parallel IRQ option is provided for system designs that require legacy ISA IRQs. Design constraints may demand more MFUNC6−MFUNC0 IRQ terminals than the PCI4410 makes available.

3.7.4 Using Parallel PCI Interrupts

Parallel PCI interrupts are available in parallel PCI interrupt mode, parallel IRQ and parallel PCI interrupt mode, or serialized IRQ and parallel PCI interrupt mode.

3.7.5 Using Serialized IRQSER Interrupts

The serialized interrupt protocol implemented in the PCI4410 uses a single terminal to communicate all interrupt status information to the host controller. The protocol defines a serial packet consisting of a start cycle, multiple interrupt indication cycles, and a stop cycle. All data in the packet is synchronous with the PCI clock. The packet data describes 16 parallel ISA IRQ signals and the optional 4 PCI interrupts $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, $\overline{\text{INTC}}$, and $\overline{\text{INTD}}$. For details on the IRQSER protocol refer to the document *Serialized IRQ Support for PCI Systems*.

3.7.6 SMI Support in the PCI4410

The PCI4410 provides a mechanism for interrupting the system when power changes have been made to the PC Card socket interfaces. The interrupt mechanism is designed to fit into a system maintenance interrupt (SMI) scheme. SMI interrupts are generated by the PCI4410, when enabled, after a write cycle to either the socket control register (see Section 6.5) of the CardBus register set or the ExCA power control register (see Section 5.3).

The SMI control is programmed through three bits in the system control register (see Section 4.29). These bits are SMIROUTE (bit 26), SMISTATUS (bit 25), and SMIENB (bit 24). Table 3−9 describes the SMI control bits function.

Table 3−9. SMI Control

BIT NAME	FUNCTION
SMIROUTE	This shared bit controls whether the SMI interrupts are sent as a CSC interrupt or as IRQ2.
SMISTAT	This socket-dependent bit is set when an SMI interrupt is pending. This status flag is cleared by writing back a 1.
SMIENB	When set, SMI interrupt generation is enabled.

If CSC SMI interrupts are selected, then the SMI interrupt is sent as the CSC. The CSC interrupt can be either level or edge mode, depending upon the CSCMODE bit in the ExCA global control register (see Section 5.22).

If IRQ2 is selected by SMIROUTE, then the IRQSER signaling protocol supports SMI signaling in the IRQ2 IRQ/Data slot. In a parallel ISA IRQ system, the support for an active low IRQ2 is provided only if IRQ2 is routed to MFUNC1, MFUNC3, or MFUNC6 through the multifunction routing register (see Section 4.32).

3.8 Power Management Overview

In addition to the low-power CMOS technology process used for the PCI4410, various features are designed into the device to allow implementation of popular power-saving techniques. These features and techniques are discussed in this section.

3.8.1 Clock Run Protocol

The PCI $\overline{\text{CLKRUN}}$ feature is the primary method of power management on the PCI interface of the PCI4410. $\overline{\text{CLKRUN}}$ signaling is provided through the MFUNC6 terminal. Because some chipsets do not implement $\overline{\text{CLKRUN}}$, this is not always available to the system designer, and alternative power-saving features are provided. For details on the $\overline{\text{CLKRUN}}$ protocol see the *PCI Mobile Design Guide*.

The PCI4410 does not permit the central resource to stop the PCI clock under any of the following conditions:

- Bit 1 (KEEPCLK) in the system control register (see Section 4.29) is set.
- The PC Card-16 resource manager is busy.
- The PCI4410 CardBus master state machine is busy. A cycle may be in progress on CardBus.
- The PCI4410 master is busy. There may be posted data from CardBus to PCI in the PCI4410.
- Interrupts are pending.
- The CardBus CCLK for either socket has not been stopped by the PCI4410 $\overline{\text{CCLKRUN}}$ manager.

The PCI4410 restarts the PCI clock using the CLKRUN protocol under any of the following conditions:

- A PC Card-16 IREQ or a CardBus $\overline{\text{CINT}}$ has been asserted.
- A CardBus CBWAKE (CSTSCHG) or PC Card-16 $\overline{\text{STSCHG}}$/$\overline{\text{RI}}$ event occurs.
- A CardBus attempts to start the CCLK using $\overline{\text{CCLKRUN}}$.
- A CardBus card arbitrates for the CardBus bus using $\overline{\text{CREQ}}$.
- A 16-bit DMA PC Card asserts $\overline{\text{DREQ}}$.

3.8.2 CardBus PC Card Power Management

The PCI4410 implements its own card power management engine that can turn off the CCLK to a socket when there is no activity to the CardBus PC Card. The PCI clock-run protocol is followed on the CardBus $\overline{\text{CCLKRUN}}$ interface to control this clock management.

3.8.3 16-Bit PC Card Power Management

The COE (bit 7, ExCA power control register) and PWRDWN (bit 0, ExCA global control register) bits are provided for 16-bit PC Card power management. The COE bit places the card interface in a high-impedance state to save power. The power savings when using this feature are minimal. The COE bit will reset the PC Card when used, and the PWRDWN bit will not. Furthermore, the PWRDWN bit is an automatic COE; that is, the PWRDWN performs the COE function when there is no card activity.

> **NOTE:** The 16-bit PC Card must implement the proper pullup resistors for the COE and PWRDWN modes.

3.8.4 Suspend Mode

The $\overline{\text{SUSPEND}}$ signal, provided for backward compatibility, gates the $\overline{\text{PRST}}$ (PCI reset) signal and the $\overline{\text{GRST}}$ (global reset) signal from the PCI4410. Besides gating $\overline{\text{PRST}}$ and $\overline{\text{GRST}}$, $\overline{\text{SUSPEND}}$ also gates PCLK inside the PCI4410 in order to minimize power consumption.

Gating PCLK does not create any issues with respect to the power switch interface in the PCI4410. This is because the PCI4410 does not depend on the PCI clock to clock the power switch interface. There are two methods to clock the power switch interface in the PCI4410:

- Use an external clock to the PCI4410 CLOCK terminal
- Use the internal oscillator

It should also be noted that asynchronous signals, such as card status change interrupts and $\overline{\text{RI\_OUT}}$, can be passed to the host system without a PCI clock. However, if card status change interrupts are routed over the serial interrupt stream, then the PCI clock must be restarted in order to pass the interrupt, because neither the internal oscillator nor an external clock is routed to the serial interrupt state machine. Figure 3−16 is a functional implementation diagram.

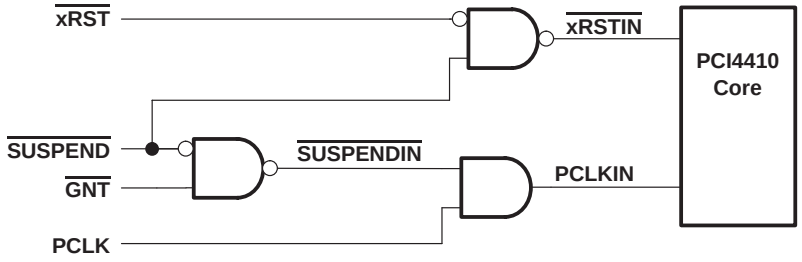


Figure 3−16. Suspend Functional Implementation

Figure 3−17 is a signal diagram of the suspend function.

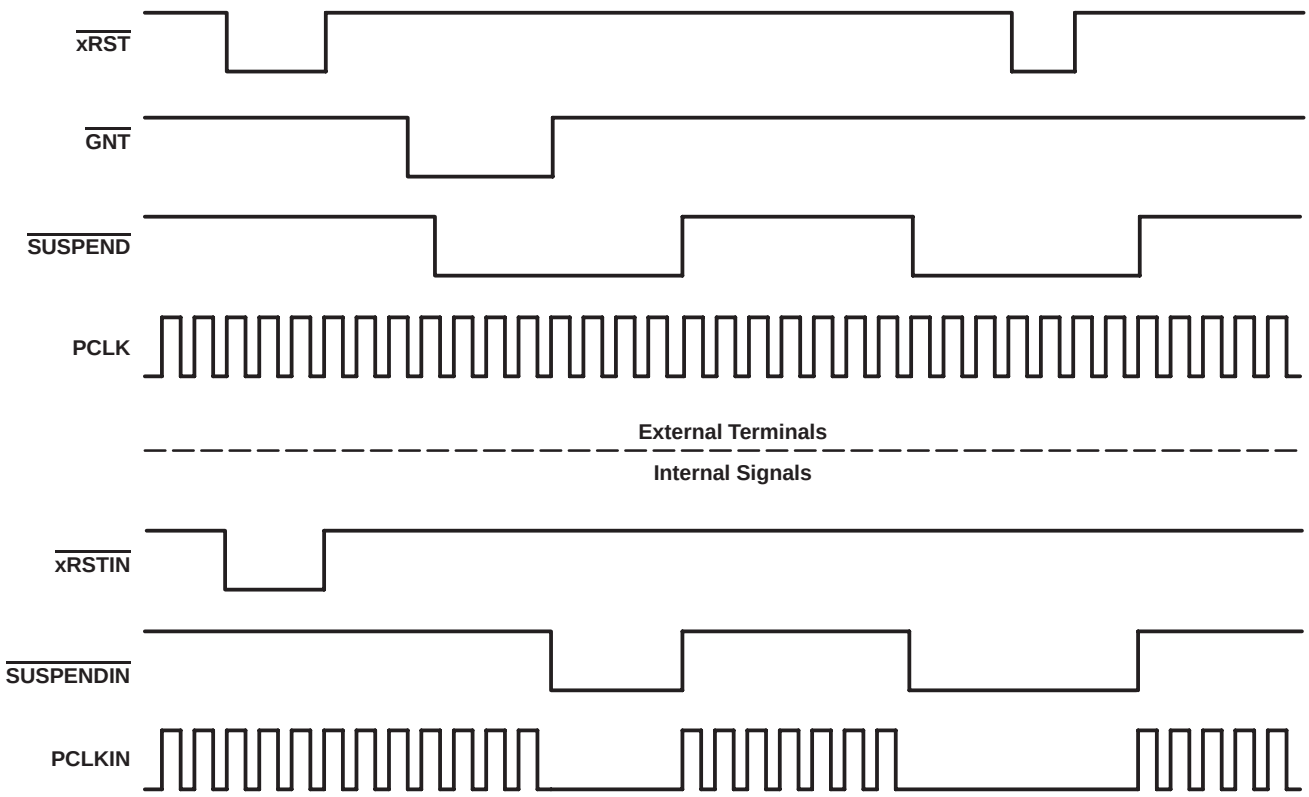


Figure 3−17. Signal Diagram of Suspend Function

3.8.5 Requirements for Suspend Mode

The suspend mode prevents the clearing of all register contents on the assertion of reset ($\overline{\text{PRST}}$ or $\overline{\text{GRST}}$) which would require the reconfiguration of the PCI4410 by software. Asserting the $\overline{\text{SUSPEND}}$ signal places the controller's PCI outputs in a high-impedance state and gates the PCLK signal internally to the controller unless a PCI transaction is currently in process ($\overline{\text{GNT}}$ is asserted). It is important that the PCI bus not be parked on the PCI4410 when $\overline{\text{SUSPEND}}$ is asserted because the outputs are in a high-impedance state.

The GPIOs, MFUNC signals, and $\overline{\text{RI\_OUT}}$ signals are all active during $\overline{\text{SUSPEND}}$, unless they are disabled in the appropriate PCI4410 registers.

3.8.6 Ring Indicate

The $\overline{\text{RI\_OUT}}$ output is an important feature in power management, allowing a system to go into a suspended mode and wake up on modem rings and other card events. TI-designed flexibility permits this signal to fit wide platform requirements. $\overline{\text{RI\_OUT}}$ on the PCI4410 can be asserted under any of the following conditions:

- A 16-bit PC Card modem in a powered socket asserts $\overline{\text{RI}}$ to indicate to the system the presence of an incoming call.
- A powered-down CardBus card asserts CSTSCHG (CBWAKE) requesting system and interface wake up.
- A powered CardBus card asserts CSTSCHG from the insertion/removal of cards or change in battery voltage levels.

Figure 3−18 shows various enable bits for the PCI4410 $\overline{\text{RI\_OUT}}$ function; however, it does not show the masking of CSC events. See Table 3−7 for a detailed description of CSC interrupt masks and flags.

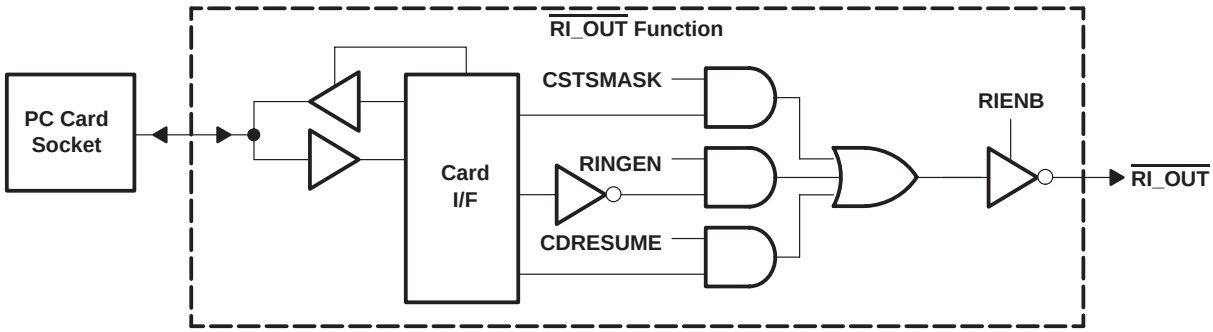


Figure 3−18. $\overline{\text{RI\_OUT}}$ Functional Diagram

$\overline{\text{RI}}$ from the 16-bit PC Card interface is masked by bit 7 (RINGEN) in the ExCA interrupt and general control register (see Section 5.4). This is programmed on a per-socket basis and is only applicable when a 16-bit card is powered in the socket.

The CBWAKE signaling to $\overline{\text{RI\_OUT}}$ is enabled through the same mask as the CSC event for CSTSCHG. The mask bit (bit 0, CSTSMASK) is programmed through the socket mask register (see Section 6.2) in the CardBus socket registers.

3.8.7 PCI Power Management

The *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges* establishes the infrastructure required to let the operating system control the power of PCI functions. This is done by defining a standard PCI interface and operations to manage the power of PCI functions on the bus. The PCI bus and the PCI functions can be assigned one of four software-visible power management states that result in varying levels of power savings.

The four power management states of PCI functions are:

- D0 − Fully-on state
- D1 and D2 − Intermediate states
- D3 − Off state

Similarly, bus power states of the PCI bus are B0−B3. The bus power states B0−B3 are derived from the device power state of the originating bridge device.

For the operating system (OS) to manage the device power states on the PCI bus, the PCI function should support four power management operations. These operations are:

- Capabilities reporting
- Power status reporting
- Setting the power state
- System wake up

The OS identifies the capabilities of the PCI function by traversing the new capabilities list. The presence of capabilities in addition to the standard PCI capabilities is indicated by a 1 in bit 4 (CAPLIST) of the status register (see Section 4.5).

The capabilities pointer provides access to the first item in the linked list of capabilities. For the PCI4410, a CardBus bridge with PCI configuration space header type 2, the capabilities pointer is mapped to an offset of 14h. The first byte of each capability register block is required to be a unique ID of that capability. PCI power management has been assigned an ID of 01h. The next byte is a pointer to the next pointer item in the list of capabilities. If there are no more items in the list, then the next item pointer should be set to 0. The registers following the next item pointer are specific to the function's capability. The PCI power management capability implements the register block outlined in Table 3−10.

Table 3−10. Power Management Registers

REGISTER NAME				OFFSET
Power management capabilities		Next item pointer	Capability ID	A0h
Data	PMCSR bridge support extensions	Power management control status (CSR)		A4h

The power management capabilities register (see Section 4.41) is a static read-only register that provides information on the capabilities of the function related to power management. The power management/control status register (offset A4h, see Section 4.42) enables control of power management states and enables/monitors power management events. The data register is an optional register that can provide dynamic data.

For more information on PCI power management, see the *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges*.

3.8.8 CardBus Bridge Power Management

The *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges* was approved by PCMCIA in December of 1997. This specification follows the device and bus state definitions provided in the *PCI Bus Power Management Interface Specification* published by the PCI Special Interest Group (SIG). The main issue addressed in the *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges* is wake up from $D3_{hot}$ or $D3_{cold}$ without losing wake-up context (also called $\overline{PME}$ context).

The specific issues addressed by the *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges* for D3 wake up are as follows:

- Preservation of device context: The specification states that a reset must occur when transitioning from D3 to D0. Some method to preserve wake-up context must be implemented so that the reset does not clear the $\overline{PME}$ context registers.
- Power source in $D3_{cold}$ if wake-up support is required from this state.

The Texas Instruments PCI4410 addresses these D3 wake-up issues in the following manner:

- Two resets are provided to handle preservation of $\overline{PME}$ context bits:
  - Global reset ($\overline{GRST}$) is used only on the initial boot up of the system after power up. It places the PCI4410 in its default state and requires BIOS to configure the device before becoming fully functional.
  - PCI reset ($\overline{PRST}$) now has dual functionality based on whether $\overline{PME}$ is enabled or not. If $\overline{PME}$ is enabled, then $\overline{PME}$ context is preserved. If $\overline{PME}$ is not enabled, then $\overline{PRST}$ acts the same as a normal PCI reset. Please see the master list of $\overline{PME}$ context bits in Section 3.8.10.
- Power source in $D3_{cold}$ if wake-up support is required from this state. Because V_{CC} is removed in $D3_{cold}$, an auxiliary power source must be supplied to the PCI4410 V_{CC} pins. Consult the *PCI14xx Implementation Guide for D3 Wake-Up* or the *PCI Power Management Interface Specification for PCI to CardBus Bridges* for further information.

3.8.9 ACPI Support

The *Advanced Configuration and Power Interface (ACPI) Specification* provides a mechanism that allows unique pieces of hardware to be described to the ACPI driver. The PCI4410 offers a generic interface that is compliant with ACPI design rules.

Two doublewords of general-purpose ACPI programming bits reside in PCI4410 PCI configuration space at offset A8h. The programming model is broken into status and control functions. In compliance with ACPI, the top level event status and enable bits reside in the general-purpose event status (see Section 4.45) and general-purpose event enable (see Section 4.46) registers.

The status and enable bits generate an event that allows the ACPI driver to call a control method associated with the pending status bit. The control method can then control the hardware by manipulating the hardware control bits or by investigating child status bits and calling their respective control methods. A hierarchical implementation would be somewhat limiting, however, as upstream devices would have to remain in some level of power state to report events.

For more information of ACPI, see the *Advanced Configuration and Power Interface (ACPI) Specification.*

3.8.10 Master List of $\overline{\text{PME}}$ Context Bits and Global Reset Only Bits

If the $\overline{\text{PME}}$ enable bit (PCI offset A4h, bit 8) is asserted, then the assertion of $\overline{\text{PRST}}$ will not clear the following $\overline{\text{PME}}$ context bits. If the $\overline{\text{PME}}$ enable bit is not asserted, then the $\overline{\text{PME}}$ context bits are cleared with $\overline{\text{PRST}}$. The $\overline{\text{PME}}$ context bits are:

- Bridge control register (PCI offset 3Eh): bit 6
- Power management control/status register (PCI offset A4h): bits 15, 8
- ExCA power control register (ExCA offset 802h): bits 4, 3, 1, 0
- ExCA interrupt and general control (ExCA offset 803h): bits 6, 5
- ExCA card status change interrupt register (ExCA offset 805h): bits 3−0
- CardBus socket event register (CardBus offset 00h): bits 3−0
- CardBus socket mask register (CardBus offset 04h): bits 3−0
- CardBus socket present state register (CardBus offset 08h): bits 13−10, 7, 5−0
- CardBus socket control register (CardBus offset 10h): bits 6−4, 2−0

Global reset places all registers in their default state regardless of the state of the $\overline{\text{PME}}$ enable bit. The $\overline{\text{GRST}}$ signal is gated only by the $\overline{\text{SUSPEND}}$ signal. This means that assertion of $\overline{\text{SUSPEND}}$ blocks the $\overline{\text{GRST}}$ signal internally, thus preserving all register contents. The registers cleared by $\overline{\text{GRST}}$ are:

- Subsystem ID/subsystem vendor ID (PCI offset 40h): bits 31–0
- PC Card 16-bit legacy mode base address register (PCI offset 44h): bits 31–1
- System control register (PCI offset 80h): bits 31−24, 22–14, 6−3, 1, 0
- General status register (PCI offset 85h): bits 2−0
- General control register (PCI offset 86h): bits 3, 1, 0
- Multifunction routing register (PCI offset 8Ch): bits 27−0
- Retry status register (PCI offset 90h): bits 7, 6, 3, 1
- Card control register (PCI offset 91h): bits 7−5, 2−0
- Device control register (PCI offset 92h): bits 7−0
- Diagnostic register (PCI offset 93h): bits 7−0
- Socket DMA register 0 (PCI offset 94h): bits 1−0
- Socket DMA register 1 (PCI offset 98h): bits 15−4, 2−0
- Power management capabilities register (PCI offset A2h): bit 15
- General-purpose event enable register (PCI offset AAh): bits 15, 11, 8, 4−0
- General-purpose output register (PCI offset AEh): bits 4−0
- PCI miscellaneous configuration register (OHCI function, PCI offset F0h): bits 15, 13, 10, 2−0
- Link enhancements register (OHCI function, PCI offset F4h): bits 13, 12, 9−7, 2, 1
- GPIO control register (OHCI function, PCI offset FCh): bits 29, 28, 24, 21, 20, 16, 15, 13, 12, 8, 7, 5, 4, 0
- Global unique ID low/high (OHCI function, PCI offset 24h−28h): bits 31−0
- ExCA identification and revision register (ExCA offset 00h): bits 7−0
- ExCA card status change register (ExCA offset 804h): bits 3−0
- ExCA global control register (ExCA offset 1Eh): bits 3−0

4 PC Card Controller Programming Model

This section describes the PCI4410 PCI configuration registers that make up the 256-byte PCI configuration header for each PCI4410 function. As noted, some bits are global in nature and are accessed only through function 0.

4.1 PCI Configuration Registers (Functions 0 and 1)

The PCI4410 is a multifunction PCI device, and the PC Card controller is integrated as PCI functions 0 and 1. The configuration header is compliant with the *PCI Local Bus Specification* as a CardBus bridge header and is PC 99 compliant as well. Table 4−1 shows the PCI configuration header, which includes both the predefined portion of the configuration space and the user-definable registers.

Table 4−1. PCI Configuration Registers (Functions 0 and 1)

REGISTER NAME				OFFSET
Device ID		Vendor ID		00h
Status		Command		04h
PCI class code			Revision ID	08h
BIST	Header type	Latency timer	Cache line size	0Ch
CardBus socket/ExCA base address				10h
Secondary status		Reserved	Capability pointer	14h
CardBus latency timer	Subordinate bus number	CardBus bus number	PCI bus number	18h
CardBus Memory base register 0				1Ch
CardBus Memory limit register 0				20h
CardBus Memory base register 1				24h
CardBus Memory limit register 1				28h
CardBus I/O base register 0				2Ch
CardBus I/O limit register 0				30h
CardBus I/O base register 1				34h
CardBus I/O limit register 1				38h
Bridge control		Interrupt pin	Interrupt line	3Ch
Subsystem ID		Subsystem vendor ID		40h
PC Card 16-bit I/F legacy-mode base address				44h
Reserved				48h−7Ch
System control				80h
Reserved	General control	General status	Reserved	84h
Reserved				88h−8Bh
Multifunction routing				8Ch
Diagnostic	Device control	Card control	Retry status	90h
Socket DMA register 0				94h
Socket DMA register 1				98h
Reserved				9Ch
Power management capabilities		Next-item pointer	Capability ID	A0h
Power management data	Power management control/status register bridge support extensions	Power management control/status		A4h
General-purpose event enable		General-purpose event status		A8h
General-purpose output		General-purpose input		ACh
Reserved				B0h−FCh

4.2 Vendor ID Register

This 16-bit register contains a value allocated by the PCI SIG (special interest group) and identifies the manufacturer of the PCI device. The vendor ID assigned to TI is 104Ch.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Vendor ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	1	0	0	0	0	0	1	0	0	1	1	0	0

Register: **Vendor ID**
Type: Read-only
Offset: 00h
Default: 104Ch

4.3 Device ID Register

This 16-bit register contains a value assigned to the PCI4410 by TI. The device identification for the PCI4410 is AC41h.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Device ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	1	0	1	0	1	1	0	0	0	1	0	0	0	0	0	1

Register: **Device ID**
Type: Read-only
Offset: 02h
Default: AC41h

4.4 Command Register

The command register provides control over the PCI4410 interface to the PCI bus. All bit functions adhere to the definitions in *PCI Local Bus Specification*. See Table 4−2 for the complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Command															
Type	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Command**
Type: Read-only, Read/Write
Offset: 04h
Default: 0000h

Table 4−2. Command Register

BIT	SIGNAL	TYPE	FUNCTION
15−10	RSVD	R	Reserved. Bits 15−10 return 0s when read.
9	FBB_EN	R	Fast back-to-back enable. The PCI4410 does not generate fast back-to-back transactions; therefore, bit 9 returns 0 when read.
8	SERR_EN	R/W	System error ($\overline{\text{SERR}}$) enable. Bit 8 controls the enable for the $\overline{\text{SERR}}$ driver on the PCI interface. $\overline{\text{SERR}}$ can be asserted after detecting an address parity error on the PCI bus. Both bits 8 and 6 must be set for the PCI4410 to report address parity errors. 0 = Disable $\overline{\text{SERR}}$ output driver (default) 1 = Enable $\overline{\text{SERR}}$ output driver
7	STEP_EN	R	Address/data stepping control. The PCI4410 does not support address/data stepping; therefore, bit 7 is hardwired to 0.
6	PERR_EN	R/W	Parity error response enable. Bit 6 controls the PCI4410's response to parity errors through $\overline{\text{PERR}}$. Data parity errors are indicated by asserting $\overline{\text{PERR}}$, whereas address parity errors are indicated by asserting $\overline{\text{SERR}}$. 0 = PCI4410 ignores detected parity error (default) 1 = PCI4410 responds to detected parity errors
5	VGA_EN	R/W	VGA palette snoop. When bit 5 is set to 1, palette snooping is enabled (that is, the PCI4410 does not respond to palette register writes and snoops the data). When bit 5 is 0, the PCI4410 treats all palette accesses like all other accesses.
4	MWI_EN	R	Memory write and invalidate enable. Bit 4 controls whether a PCI initiator device can generate memory write-and-Invalidate commands. The PCI4410 controller does not support memory write and invalidate commands. It uses memory write commands instead; therefore, this bit is hardwired to 0.
3	SPECIAL	R	Special cycles. Bit 3 controls whether or not a PCI device ignores PCI special cycles. The PCI4410 does not respond to special cycle operations; therefore, this bit is hardwired to 0.
2	MAST_EN	R/W	Bus master control. Bit 2 controls whether or not the PCI4410 can act as a PCI bus initiator (master). The PCI4410 can take control of the PCI bus only when this bit is set. 0 = Disables the PCI4410's ability to generate PCI bus accesses (default) 1 = Enables the PCI4410's ability to generate PCI bus accesses
1	MEM_EN	R/W	Memory space enable. Bit 1 controls whether or not the PCI4410 can claim cycles in PCI memory space. 0 = Disables the PCI4410's response to memory space accesses (default) 1 = Enables the PCI4410's response to memory space accesses
0	IO_EN	R/W	I/O space control. Bit 0 controls whether or not the PCI4410 can claim cycles in PCI I/O space. 0 = Disables the PCI4410 from responding to I/O space accesses (default) 1 = Enables the PCI4410 to respond to I/O space accesses

4.5 Status Register

The status register provides device information to the host system. Bits in this register may be read normally. A bit in the status register is reset when a 1 is written to that bit location; a 0 written to a bit location has no effect. All bit functions adhere to the definitions in the *PCI Local Bus Specification*. PCI bus status is shown through each function. See Table 4−3 for the complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Status															
Type	R/C	R/C	R/C	R/C	R/C	R	R	R/C	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0

Register: **Status**
Type: Read-only, Read/Write to Clear
Offset: 06h
Default: 0210h

Table 4−3. Status Register

BIT	SIGNAL	TYPE	FUNCTION
15	PAR_ERR	R/C	Detected parity error. Bit 15 is set when a parity error is detected (either address or data).
14	SYS_ERR	R/C	Signaled system error. Bit 14 is set when $\overline{\text{SERR}}$ is enabled and the PCI4410 signals a system error to the host.
13	MABORT	R/C	Received master abort. Bit 13 is set when a cycle initiated by the PCI4410 on the PCI bus has been terminated by a master abort.
12	TABT_REC	R/C	Received target abort. Bit 12 is set when a cycle initiated by the PCI4410 on the PCI bus was terminated by a target abort.
11	TABT_SIG	R/C	Signaled target abort. Bit 11 is set by the PCI4410 when it terminates a transaction on the PCI bus with a target abort.
10−9	PCI_SPEED	R	$\overline{\text{DEVSEL}}$ timing. These bits encode the timing of $\overline{\text{DEVSEL}}$ and are hardwired 01b, indicating that the PCI4410 asserts PCI_SPEED at a medium speed on nonconfiguration cycle accesses.
8	DATAPAR	R/C	Data parity error detected. 0 = The conditions for setting bit 8 have not been met. 1 = A data parity error occurred, and the following conditions were met: a. $\overline{\text{PERR}}$ was asserted by any PCI device including the PCI4410. b. The PCI4410 was the bus master during the data parity error. c. The parity error response bit is set in the command.
7	FBB_CAP	R	Fast back-to-back capable. The PCI4410 cannot accept fast back-to-back transactions; therefore, bit 7 is hardwired to 0.
6	UDF	R	User-definable feature support. The PCI4410 does not support the user-definable features; therefore, bit 6 is hardwired to 0.
5	66MHZ	R	66-MHz capable. The PCI4410 operates at a maximum PCLK frequency of 33 MHz; therefore, bit 5 is hardwired to 0.
4	CAPLIST	R	Capabilities list. Bit 4 returns 1 when read. This bit indicates that capabilities in addition to standard PCI capabilities are implemented. The linked list of PCI power management capabilities is implemented in this function.
3−0	RSVD	R	Reserved. Bits 3−0 return 0s when read.

4.6 Revision ID Register

The revision ID register indicates the silicon revision of the PCI4410.

Bit	7	6	5	4	3	2	1	0
Name	Revision ID							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	1

Register: **Revision ID**
Type: Read-only
Offset: 08h
Default: 01h

4.7 PCI Class Code Register

The class code register recognizes the PCI4410 as a bridge device (06h) and CardBus bridge device (07h) with a 00h programming interface.

Bit	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCI class code																							
	Base class								Subclass								Programming interface							
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

Register: **PCI class code**
Type: Read-only
Offset: 09h
Default: 060700h

4.8 Cache Line Size Register

The cache line size register is programmed by host software to indicate the system cache line size.

Bit	7	6	5	4	3	2	1	0
Name	Cache line size							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Cache line size**
Type: Read/Write
Offset: 0Ch
Default: 00h

4.9 Latency Timer Register

The latency timer register specifies the latency timer for the PCI4410 in units of PCI clock cycles. When the PCI4410 is a PCI bus initiator and asserts $\overline{\text{FRAME}}$, the latency timer begins counting from zero. If the latency timer expires before the PCI4410 transaction has terminated, then the PCI4410 terminates the transaction when its $\overline{\text{GNT}}$ is deasserted.

Bit	7	6	5	4	3	2	1	0
Name	Latency timer							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Latency timer**
Type: Read/Write
Offset: 0Dh
Default: 00h

4.10 Header Type Register

This register returns 82h when read, indicating that the PCI4410 configuration spaces adhere to the CardBus bridge PCI header. The CardBus bridge PCI header ranges from PCI register 0 to 7Fh, and 80h−FFh are user-definable extension registers.

Bit	7	6	5	4	3	2	1	0
Name	Header type							
Type	R	R	R	R	R	R	R	R
Default	1	0	0	0	0	0	1	0

Register: **Header type**
Type: Read-only
Offset: 0Eh
Default: 82h

4.11 BIST Register

Because the PCI4410 does not support a built-in self-test (BIST), this register returns the value of 00h when read.

Bit	7	6	5	4	3	2	1	0
Name	BIST							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **BIST**
Type: Read-only
Offset: 0Fh
Default: 00h

4.12 CardBus Socket/ExCA Base-Address Register

The CardBus socket/ExCA base-address register is programmed with a base address referencing the CardBus socket registers and the memory-mapped ExCA register set. Bits 31−12 are read/write and allow the base address to be located anywhere in the 32-bit PCI memory address space on a 4-Kbyte boundary. Bits 11−0 are read-only, returning 0s when read. When software writes all 1s to this register, the value read back is FFFF F000h, indicating that at least 4K bytes of memory address space are required. The CardBus registers start at offset 000h, and the memory-mapped ExCA registers begin at offset 800h.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CardBus socket/ExCA base-address															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CardBus socket/ExCA base-address															
Type	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **CardBus socket/ExCA base-address**
Type: Read-only, Read/Write
Offset: 10h
Default: 0000 0000h

4.13 Capability Pointer Register

The capability pointer register provides a pointer into the PCI configuration header where the PCI power management register block resides. PCI header doublewords at A0h and A4h provide the power management (PM) registers. The socket has its own capability pointer register. This register returns A0h when read.

Bit	7	6	5	4	3	2	1	0
Name	Capability pointer							
Type	R	R	R	R	R	R	R	R
Default	1	0	1	0	0	0	0	0

Register: **Capability pointer**
Type: Read-only
Offset: 14h
Default: A0h

4.14 Secondary Status Register

The secondary status register is compatible with the PCI-to-PCI bridge secondary status register and indicates CardBus-related device information to the host system. This register is very similar to the PCI status register (offset 06h); status bits are cleared by writing a 1. See Table 4−4 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Secondary status															
Type	R/C	R/C	R/C	R/C	R/C	R	R	R/C	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Register: **Secondary status**
Type: Read-only, Read/Write to Clear
Offset: 16h
Default: 0200h

Table 4−4. Secondary Status Register

BIT	SIGNAL	TYPE	FUNCTION
15	CBPARITY	R/C	Detected parity error. Bit 15 is set when a CardBus parity error is detected (either address or data).
14	CBSERR	R/C	Signaled system error. Bit 14 is set when $\overline{\text{CSERR}}$ is signaled by a CardBus card. The PCI4410 does not assert $\overline{\text{CSERR}}$.
13	CBMABORT	R/C	Received master abort. Bit 13 is set when a cycle initiated by the PCI4410 on the CardBus bus has been terminated by a master abort.
12	REC_CBTA	R/C	Received target abort. Bit 12 is set when a cycle initiated by the PCI4410 on the CardBus bus is terminated by a target abort.
11	SIG_CBTA	R/C	Signaled target abort. Bit 11 is set by the PCI4410 when it terminates a transaction on the CardBus bus with a target abort.
10−9	CB_SPEED	R	$\overline{\text{CDEVSEL}}$ timing. These bits encode the timing of $\overline{\text{CDEVSEL}}$ and are hardwired 01b, indicating that the PCI4410 asserts CB_SPEED at a medium speed.
8	CB_DPAR	R/C	CardBus data parity error detected. 0 = The conditions for setting bit 8 have not been met. 1 = A data parity error occurred and the following conditions were met: a. $\overline{\text{CPERR}}$ was asserted on the CardBus interface. b. The PCI4410 was the bus master during the data parity error. c. The parity error response bit is set in the bridge control.
7	CBFBB_CAP	R	Fast back-to-back capable. The PCI4410 cannot accept fast back-to-back transactions; therefore, bit 7 is hardwired to 0.
6	CB_UDF	R	User-definable feature support. The PCI4410 does not support the user-definable features; therefore, bit 6 is hardwired to 0.
5	CB66MHZ	R	66-MHz capable. The PCI4410 CardBus interface operates at a maximum CCLK frequency of 33 MHz; therefore, bit 5 is hardwired to 0.
4−0	RSVD	R	Reserved. Bits 4−0 return 0s when read.

4.15 PCI Bus Number Register

This register is programmed by the host system to indicate the bus number of the PCI bus to which the PCI4410 is connected. The PCI4410 uses this register in conjunction with the CardBus bus number and subordinate bus number registers to determine when to forward PCI configuration cycles to its secondary buses.

Bit	7	6	5	4	3	2	1	0
Name	PCI bus number							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **PCI bus number**
Type: Read/Write
Offset: 18h
Default: 00h

4.16 CardBus Bus Number Register

This register is programmed by the host system to indicate the bus number of the CardBus bus to which the PCI4410 is connected. The PCI4410 uses this register in conjunction with the PCI bus number and subordinate bus number registers to determine when to forward PCI configuration cycles to its secondary buses.

Bit	7	6	5	4	3	2	1	0
Name	CardBus bus number							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **CardBus bus number**
Type: Read/Write
Offset: 19h
Default: 00h

4.17 Subordinate Bus Number Register

This register is programmed by the host system to indicate the highest-numbered bus below the CardBus bus. The PCI4410 uses this register in conjunction with the PCI bus number and CardBus bus number registers to determine when to forward PCI configuration cycles to its secondary buses.

Bit	7	6	5	4	3	2	1	0
Name	Subordinate bus number							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **Subordinate bus number**
Type: Read/Write
Offset: 1Ah
Default: 00h

4.18 CardBus Latency Timer Register

This register is programmed by the host system to specify the latency timer for the PCI4410 CardBus interface in units of CCLK cycles. When the PCI4410 is a CardBus initiator and asserts $\overline{\text{CFRAME}}$, the CardBus latency timer begins counting. If the latency timer expires before the PCI4410 transaction has terminated, then the PCI4410 terminates the transaction at the end of the next data phase. A recommended minimum value for this register is 20h, which allows most transactions to be completed.

Bit	7	6	5	4	3	2	1	0
Name	CardBus latency timer							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **CardBus latency timer**
Type: Read/Write
Offset: 1Bh
Default: 00h

4.19 Memory Base Registers 0, 1

The memory base registers indicate the lower address of a PCI memory address range. These registers are used by the PCI4410 to determine when to forward a memory transaction to the CardBus bus and when to forward a CardBus cycle to PCI. Bits 31−12 of these registers are read/write and allow the memory base to be located anywhere in the 32-bit PCI memory space on 4-Kbyte boundaries. Bits 11−0 are read-only and always return 0s. Write transactions to these bits have no effect. Bits 8 and 9 of the bridge control register (see Section 4.25) specify whether memory windows 0 and 1 are prefetchable or nonprefetchable. The memory base register or the memory limit register must be nonzero for the PCI4410 to claim any memory transactions through CardBus memory windows (that is, these windows are not enabled by default to pass the first 4K bytes of memory to CardBus).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Memory base registers 0, 1															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Memory base registers 0, 1															
Type	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Memory base registers 0, 1**
Type: Read-only, Read/Write
Offset: 1Ch, 24h
Default: 0000 0000h

4.20 Memory Limit Registers 0, 1

The memory limit registers indicate the upper address of a PCI memory address range. These registers are used by the PCI4410 to determine when to forward a memory transaction to the CardBus bus and when to forward a CardBus cycle to PCI. Bits 31−12 of these registers are read/write and allow the memory base to be located anywhere in the 32-bit PCI memory space on 4-Kbyte boundaries. Bits 11−0 are read-only and always return 0s. Write transactions to these bits have no effect. Bits 8 and 9 of the bridge control register specify whether memory windows 0 and 1 are prefetchable or nonprefetchable. The memory base register or the memory limit register must be nonzero for the PCI4410 to claim any memory transactions through CardBus memory windows (that is, these windows are not enabled by default to pass the first 4K bytes of memory to CardBus).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Memory limit registers 0, 1															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Memory limit registers 0, 1															
Type	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Memory limit registers 0, 1**
Type: Read-only, Read/Write
Offset: 20h, 28h
Default: 0000 0000h

4.21 I/O Base Registers 0, 1

The I/O base registers indicate the lower address of a PCI I/O address range. These registers are used by the PCI4410 to determine when to forward an I/O transaction to the CardBus bus and when to forward a CardBus cycle to the PCI bus. The lower 16 bits of this register locate the bottom of the I/O window within a 64-Kbyte page, and the upper 16 bits (31−16) are a page register which locates this 64-Kbyte page in 32-bit PCI I/O address space. Bits 31−2 are read/write. Bits 1 and 0 are read-only and always return 0s, forcing I/O windows to be aligned on a natural doubleword boundary.

NOTE: Either the I/O base or the I/O limit register must be nonzero to enable any I/O transactions.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	I/O base registers 0, 1															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	I/O base registers 0, 1															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **I/O base registers 0, 1**
Type: Read-only, Read/Write
Offset: 2Ch, 34h
Default: 0000 0000h

4.22 I/O Limit Registers 0, 1

The I/O limit registers indicate the upper address of a PCI I/O address range. These registers are used by the PCI4410 to determine when to forward an I/O transaction to the CardBus bus and when to forward a CardBus cycle to PCI. The lower 16 bits of this register locate the top of the I/O window within a 64-Kbyte page, and the upper 16 bits are a page register that locates this 64-Kbyte page in 32-bit PCI I/O address space. Bits 15−2 are read/write and allow the I/O limit address to be located anywhere in the 64-Kbyte page (indicated by bits 31−16 of the appropriate I/O base) on doubleword boundaries.

Bits 31−16 are read-only and always return 0s when read. The page is set in the I/O base register. Bits 1 and 0 are read-only and always return 0s, forcing I/O windows to be aligned on a natural doubleword boundary. Write transactions to read-only bits have no effect. The PCI4410 assumes that the lower 2 bits of the limit address are 1s.

NOTE: The I/O base or the I/O limit register must be nonzero to enable an I/O transaction.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	I/O limit registers 0, 1															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	I/O limit registers 0, 1															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **I/O limit registers 0, 1**
Type: Read-only, Read/Write
Offset: 30h, 38h
Default: 0000 0000h

4.23 Interrupt Line Register

The interrupt line register communicates interrupt line routing information.

Bit	7	6	5	4	3	2	1	0
Name	Interrupt line							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	1	1	1	1	1	1	1	1

Register: **Interrupt line**
Type: Read/Write
Offset: 3Ch
Default: FFh

4.24 Interrupt Pin Register

The value read from the interrupt pin register is function dependent and depends on the interrupt signaling mode, selected through bits 2−1 (INTMODE field) of the device control register (see Section 4.35). The PCI4410 defaults to serialized PCI and ISA interrupt mode.

Bit	7	6	5	4	3	2	1	0
Name	Interrupt pin							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	1

Register: **Interrupt pin**
Type: Read-only
Offset: 3Dh
Default: 01h

4.25 Bridge Control Register

The bridge control register provides control over various PCI4410 bridging functions. See Table 4−5 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Bridge control															
Type	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0

Register: **Bridge control**
Type: Read-only, Read/Write
Offset: 3Eh
Default: 0340h

Table 4−5. Bridge Control Register

BIT	SIGNAL	TYPE	FUNCTION
15−11	RSVD	R	Reserved. Bits 15−11 return 0s when read.
10	POSTEN	R/W	Write posting enable. Enables write posting to and from the CardBus sockets. Write posting enables posting of write data on burst cycles. Operating with write posting disabled inhibits performance on burst cycles. Note that bursted write data can be posted, but various write transactions may not.
9	PREFETCH1	R/W	Memory window 1 type. Bit 9 specifies whether or not memory window 1 is prefetchable. This bit is socket dependent. Bit 9 is encoded as: 0 = Memory window 1 is nonprefetchable. 1 = Memory window 1 is prefetchable (default).
8	PREFETCH0	R/W	Memory window 0 type. Bit 8 specifies whether or not memory window 0 is prefetchable. This bit is encoded as: 0 = Memory window 0 is nonprefetchable. 1 = Memory window 0 is prefetchable (default).
7	INTR	R/W	PCI interrupt − IREQ routing enable. Bit 7 selects whether PC Card functional interrupts are routed to PCI interrupts or to the IRQ specified in the ExCA registers. 0 = Functional interrupts routed to PCI interrupts (default) 1 = Functional interrupts routed to IRQ interrupts
6	CRST	R/W	CardBus reset. When bit 6 is set, $\overline{\text{CRST}}$ is asserted on the CardBus interface. $\overline{\text{CRST}}$ can also be asserted by passing a $\overline{\text{PRST}}$ assertion to CardBus. 0 = $\overline{\text{CRST}}$ deasserted 1 = $\overline{\text{CRST}}$ asserted (default)
5	MABTMODE	R/W	Master abort mode. Bit 5 controls how the PCI4410 responds to a master abort when the PCI4410 is an initiator on the CardBus interface. 0 = Master aborts not signaled (default) 1 = Signal target abort on PCI. Signal $\overline{\text{SERR}}$ (if enabled)
4	RSVD	R	Reserved. Bit 4 returns 0 when read.
3	VGAEN	R/W	VGA enable. Bit 3 affects how the PCI4410 responds to VGA addresses. When this bit is set, accesses to VGA addresses are forwarded.
2	ISAEN	R/W	ISA mode enable. Bit 2 affects how the PCI4410 passes I/O cycles within the 64-Kbyte ISA range. This bit is not common between sockets. When this bit is set, the PCI4410 does not forward the last 768 bytes of each 1K I/O range to CardBus.
1	CSERREN	R/W	$\overline{\text{CSERR}}$ enable. Bit 1 controls the response of the PCI4410 to $\overline{\text{CSERR}}$ signals on the CardBus bus. 0 = $\overline{\text{CSERR}}$ is not forwarded to PCI $\overline{\text{SERR}}$. 1 = $\overline{\text{CSERR}}$ is forwarded to PCI $\overline{\text{SERR}}$.
0	CPERREN	R/W	CardBus parity error response enable. Bit 0 controls the response of the PCI4410 to CardBus parity errors. 0 = CardBus parity errors are ignored. 1 = CardBus parity errors are reported using $\overline{\text{CPERR}}$.

4.26 Subsystem Vendor ID Register

The subsystem vendor ID register is used for system and option-card identification purposes and may be required for certain operating systems. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register (see Section 4.29).

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Subsystem vendor ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Subsystem vendor ID**
Type: Read-only (Read/Write if enabled by SUBSYSRW)
Offset: 40h
Default: 0000h

4.27 Subsystem ID Register

The subsystem ID register is used for system and option-card identification purposes and may be required for certain operating systems. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register (see Section 4.29).

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Subsystem ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Subsystem ID**
Type: Read-only (Read/Write if enabled by SUBSYSRW)
Offset: 42h
Default: 0000h

4.28 PC Card 16-Bit I/F Legacy-Mode Base-Address Register

The PCI4410 supports the index/data scheme of accessing the ExCA registers, which is mapped by this register. An address written to this register is the address for the index register and the address + 1 is the data address. Using this access method, applications requiring index/data ExCA access can be supported. The base address can be mapped anywhere in 32-bit I/O space on a word boundary; hence, bit 0 is read-only, returning 1 when read. See Section 5, *ExCA Compatibility Registers*, for register offsets.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PC Card 16-bit I/F legacy-mode base-address															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PC Card 16-bit I/F legacy-mode base-address															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Register: **PC Card 16-bit I/F legacy-mode base-address**
Type: Read-only, Read/Write
Offset: 44h
Default: 0000 0001h

4.29 System Control Register

System-level initializations are performed through programming this doubleword register. See Table 4−6 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	System control															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/C	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	System control															
Type	R/W	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0

Register: **System control**
Type: Read-only, Read/Write, Read/Write to Clear
Offset: 80h
Default: 0044 9060h

Table 4−6. System Control Register

BIT	SIGNAL	TYPE	FUNCTION
31−30	SER_STEP	R/W	Serialized PCI interrupt routing step. Bits 31 and 30 configure the serialized PCI interrupt stream signaling and accomplish an even distribution of interrupts signaled on the four PCI interrupt slots. Bits 31 and 30 are global to all PCI4410 functions. 00 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ signal in $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ slots (default) 01 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ signal in $\overline{\text{INTB}}$/$\overline{\text{INTC}}$ slots 10 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ signal in $\overline{\text{INTC}}$/$\overline{\text{INTD}}$ slots 11 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ signal in $\overline{\text{INTD}}$/$\overline{\text{INTA}}$ slots
29	TIE_INTB_INTA	R/W	Tie $\overline{\text{INTB}}$ to $\overline{\text{INTA}}$. When bit 29 is set to 1, $\overline{\text{INTB}}$ is tied to $\overline{\text{INTA}}$ (default is 0).
28	DIAGNOSTIC	R/W	TI diagnostic (IIC_Test) bit (default is 0).
27	OSEN	R/W	Internal oscillator enable. 0 = Internal oscillator disabled (default) 1 = Internal oscillator enabled.
26	SMIROUTE	R/W	SMI interrupt routing. Bit 26 selects whether IRQ2 or CSC is signaled when a write occurs to power a PC Card socket. 0 = PC Card power change interrupts routed to IRQ2 (default) 1 = A CSC interrupt is generated on PC Card power changes.
25	SMISTATUS	R/C	SMI interrupt status. This bit is set when bit 24 (SMIENB) is set and a write occurs to set the socket power. Writing a 1 to bit 25 clears the status. 0 = SMI interrupt signaled (default) 1 = SMI interrupt not signaled
24	SMIENB	R/W	SMI interrupt mode enable. When bit 24 is set and a write to the socket power control occurs, the SMI interrupt signaling is enabled and generates an interrupt.
23	PCIPMEN	R/W	*PCI Bus Power Management Interface Specification* revision 1.1 enable. 0 = Use *PCI Bus Power Management Interface Specification* revision 1.0 implementation (default). 1 = Use *PCI Bus Power Management Interface Specification* revision 1.1 implementation. Note: See power management capability register (PCI offset A2h) (Section 4.41), VERSION bits 2−0 for additional information.
22	CBRSVD	R/W	CardBus reserved terminals signaling. When a CardBus card is inserted and bit 22 is set, the RSVD CardBus terminals are driven low. When this bit is 0, these signals are placed in a high-impedance state. 0 = 3-state CardBus RSVD 1 = Drive Cardbus RSVD low (default)
21	VCCPROT	R/W	V_{CC} protection enable. 0 = V_{CC} protection enabled for 16-bit cards (default) 1 = V_{CC} protection disabled for 16-bit cards
20	REDUCEZV	R/W	Reduced zoomed video enable. When this bit is enabled, pins A25−A22 of the card interface for PC Card-16 cards are placed in the high-impedance state. This bit should not be set for normal ZV operation. This bit is encoded as: 0 = Reduced zoomed video disabled (default) 1 = Reduced zoomed video enabled
19	CDREQEN	R/W	PC/PCI DMA card enable. When bit 19 is set, the PCI4410 allows 16-bit PC Cards to request PC/PCI DMA using the $\overline{\text{DREQ}}$ signaling. $\overline{\text{DREQ}}$ is selected through the socket DMA register 0 (see Section 4.37). 0 = Ignore $\overline{\text{DREQ}}$ signaling from PC Cards (default) 1 = Signal DMA request on $\overline{\text{DREQ}}$
18−16	CDMACHAN	R/W	PC/PCI DMA channel assignment. Bits 18−16 are encoded as: 0−3 = 8-bit DMA channels 4 = PCI master; not used (default) 5−7 = 16-bit DMA channels
15	MRBURSTDN	R/W	Memory read burst enable downstream. When bit 15 is set, memory read transactions are allowed to burst downstream. 0 = Downstream memory read burst is disabled. 1 = Downstream memory read burst is enabled (default).

Table 4−6. System Control Register (Continued)

BIT	SIGNAL	TYPE	FUNCTION
14	MRBURSTUP	R/W	Memory read burst enable upstream. When bit 14 is set, the PCI4410 allows memory read transactions to burst upstream. 0 = Upstream memory read burst is disabled (default). 1 = Upstream memory read burst is enabled.
13	SOCACTIVE	R	Socket activity status. When set, bit 13 indicates access has been performed to or from a PC card and is cleared upon read of this status bit. 0 = No socket activity (default) 1 = Socket activity
12	RSVD	R	Reserved. Bit 12 returns 1 when read.
11	PWRSTREAM	R	Power stream in progress status bit. When set, bit 11 indicates that a power stream to the power switch is in progress and a powering change has been requested. This bit is cleared when the power stream is complete. 0 = Power stream is complete and delay has expired. 1 = Power stream is in progress.
10	DELAYUP	R	Power-up delay in progress status. When set, bit 9 indicates that a power-up stream has been sent to the power switch and proper power may not yet be stable. This bit is cleared when the power-up delay has expired.
9	DELAYDOWN	R	Power-down delay in progress status. When set, bit 10 indicates that a power-down stream has been sent to the power switch and proper power may not yet be stable. This bit is cleared when the power-down delay has expired.
8	INTERROGATE	R	Interrogation in progress. When set, bit 8 indicates an interrogation is in progress and clears when interrogation completes. This bit is socket dependent. 0 = Interrogation not in progress (default) 1 = Interrogation in progress
7	AUTOPWRSWEN	R/W	Auto power switch enable. 0 = Bit 5 (AUTOPWRSWEN) in ExCA power control register (see Section 5.3) is disabled. (default). 1 = Bit 5 (AUTOPWRSWEN) in ExCA power control register (see Section 5.3) is enabled.
6	PWRSAVINGS	R/W	Power savings mode enable. When this bit is set, if a CB card is inserted, idle, and without a CB clock, then the applicable CB state machine will not be clocked.
5	SUBSYSRW	R/W	Subsystem ID (see Section 4.27), subsystem vendor ID (see Section 4.26), ExCA identification and revision (see Section 5.1) registers read/write enable. 0 = Subsystem ID, subsystem vendor ID, ExCA identification and revision registers are read/write. 1 = Subsystem ID, subsystem vendor ID, ExCA identification and revision registers are read-only (default).
4	CB_DPAR	R/W	CardBus data parity error $\overline{\text{SERR}}$ signaling enable 0 = CardBus data parity error not signaled on PCI $\overline{\text{SERR}}$ 1 = CardBus data parity eror signaled on PCI $\overline{\text{SERR}}$
3	CDMA_EN	R/W	PC/PCI DMA enable. Bit 3 enables PC/PCI DMA when set if MFUNC0−MFUNC6 are configured for centralized DMA. 0 = Centralized DMA disabled (default) 1 = Centralized DMA enabled
2	ExCAPower	R/W	ExCA power control bit. Enabled by selecting the 82365SL mode. 0 = Enables 3.3 V 1 = Enables 5 V
1	KEEPCLK	R/W	Keep clock. This bit works with PCI and CB $\overline{\text{CLKRUN}}$ protocols. 0 = Allows normal functioning of both $\overline{\text{CLKRUN}}$ protocols (default) 1 = Does not allow CB clock or PCI clock to be stopped using the $\overline{\text{CLKRUN}}$ protocols
0	RIMUX	R/W	$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ multiplex enable. 0 = $\overline{\text{RI\_OUT}}$ and $\overline{\text{PME}}$ are both routed to the $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ terminal. If both are enabled at the same time, then $\overline{\text{RI\_OUT}}$ has precedence over $\overline{\text{PME}}$. 1 = Only $\overline{\text{PME}}$ is routed to the $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ terminal.

4.30 General Status Register

The general status register provides the general device status information. The status of the serial EEPROM interface is provided through this register. See Table 4−7 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	General status							
Type	R	R	R	R	R	R/U	R	R
Default	0	0	0	0	0	X	0	0

Register: **General status**
Type: Read/UpdateRead-only, Read/Clear
Offset: 85h (Function 0)
Default: 00h

Table 4−7. General Status Register

BIT	SIGNAL	TYPE	FUNCTION
7−3	RSVD	R	Reserved. Bits 7−3 return 0s when read.
2	EEDETECT	R	Serial EEPROM detect. Serial EEPROM is detected by sampling a logic high on SCL while $\overline{\text{PRST}}$ is low. When this bit is set, the serial ROM is detected. This status bit is encoded as: 0 = EEPROM not detected (default) 1 = EEPROM detected
1	DATAERR	R/C	Serial EEPROM data error status. This bit indicates when a data error occurs on the serial EEPROM interface. This bit may be set due to a missing acknowledge. This bit is cleared by a writeback of 1. 0 = No error detected (default) 1 = Data error detected
0	EEBUSY	R	Serial EEPROM busy status. This bit indicates the status of the PCI4410 serial EEPROM circuitry. This bit is set during the loading of the subsystem ID value. 0 = Serial EEPROM circuitry is not busy (default). 1 = Serial EEPROM circuitry is busy.

4.31 General Control Register

The general control register provides top level PCI arbitration control. See Table 4−8 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	General control							
Type	R	R	R	R	R/W	R	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **General control**
Type: Read-Only, Read/Write
Offset: 86h
Default: 00h

Table 4−8. General Control Register

BIT	SIGNAL	TYPE	FUNCTION
7−4	RSVD	R	Reserved. Bits 7−4 return 0s when read.
3	DISABLE_OHCI	R/W	When bit 3 is set, the open HCI 1394 controller function is completely nonaccessible and nonfunctional.
2	RSVD	R	Reserved. Bit 2 returns 0 when read.
1−0	ARB_CTRL	RW	Controls top level PCI arbitration. 00 = 1394 open HCI priority 01 = CardBus priority 10 = Fair round robin 11 = Reserved (fair round robin)

4.32 Multifunction Routing Register

The multifunction routing register is used to configure the MFUNC0−MFUNC6 terminals. These terminals may be configured for various functions. All multifunction terminals default to the general-purpose input configuration. This register is intended to be programmed once at power-on initialization. The default value for this register may also be loaded through a serial bus EEPROM. See Table 4−9 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Multifunction routing															
Type	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Multifunction routing															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Multifunction routing**
Type: Read-only, Read/Write
Offset: 8Ch
Default: 0000 0000h

Table 4−9. Multifunction Routing Register

BIT	SIGNAL	TYPE	FUNCTION
31−28	RSVD	R	Bits 31−28 return 0s when read.
27−24	MFUNC6	R/W	Multifunction terminal 6 configuration. These bits control the internal signal mapped to the MFUNC6 terminal as follows: 0000 = RSVD 0001 = $\overline{\text{CLKRUN}}$ 0010 = IRQ2 0011 = IRQ3 0100 = IRQ4 0101 = IRQ5 0110 = IRQ6 0111 = IRQ7 1000 = IRQ8 1001 = IRQ9 1010 = IRQ10 1011 = IRQ11 1100 = IRQ12 1101 = IRQ13 1110 = IRQ14 1111 = IRQ15
23−20	MFUNC5	R/W	Multifunction terminal 5 configuration. These bits control the internal signal mapped to the MFUNC5 terminal as follows: 0000 = GPI4 0001 = GPO4 0010 = $\overline{\text{PCGNT}}$ 0011 = IRQ3 0100 = IRQ4 0101 = $\overline{\text{D3\_STAT}}$ 0110 = $\overline{\text{ZVSTAT}}$ 0111 = $\overline{\text{ZVSEL0}}$ 1000 = CAUDPWM 1001 = IRQ9 1010 = IRQ10 1011 = IRQ11 1100 = LED_SKT 1101 = Diagnostic setup: OHCI test 1110 = $\overline{\text{GPE}}$ 1111 = IRQ15
19−16	MFUNC4	R/W	Multifunction terminal 4 configuration. These bits control the internal signal mapped to the MFUNC4 terminal as follows: NOTE: When the serial bus mode is implemented by pulling up the $\overline{\text{VCCD0}}$ and $\overline{\text{VCCD1}}$ terminals, the MFUNC4 terminal provides the SCL signaling. 0000 = GPI3 0001 = GPO3 0010 = PCI $\overline{\text{LOCK}}$ 0011 = IRQ3 0100 = IRQ4 0101 = IRQ5 0110 = $\overline{\text{ZVSTAT}}$ 0111 = $\overline{\text{ZVSEL0}}$ 1000 = CAUDPWM 1001 = IRQ9 1010 = IRQ10 1011 = IRQ11 1100 = $\overline{\text{RI\_OUT}}$ 1101 = LED_SKT 1110 = $\overline{\text{GPE}}$ 1111 = IRQ15
15−12	MFUNC3	R/W	Multifunction terminal 3 configuration. These bits control the internal signal mapped to the MFUNC3 terminal as follows: 0000 = RSVD 0001 = IRQSER 0010 = IRQ2 0011 = IRQ3 0100 = IRQ4 0101 = IRQ5 0110 = IRQ6 0111 = IRQ7 1000 = IRQ8 1001 = IRQ9 1010 = IRQ10 1011 = IRQ11 1100 = IRQ12 1101 = IRQ13 1110 = IRQ14 1111 = IRQ15
11−8	MFUNC2	R/W	Multifunction terminal 2 configuration. These bits control the internal signal mapped to the MFUNC2 terminal as follows: 0000 = GPI2 0001 = GPO2 0010 = $\overline{\text{PCREQ}}$ 0011 = IRQ3 0100 = IRQ4 0101 = IRQ5 0110 = $\overline{\text{ZVSTAT}}$ 0111 = $\overline{\text{ZVSEL0}}$ 1000 = CAUDPWM 1001 = IRQ9 1010 = IRQ10 1011 = IRQ11 1100 = $\overline{\text{RI\_OUT}}$ 1101 = $\overline{\text{D3\_STAT}}$ 1110 = $\overline{\text{GPE}}$ 1111 = IRQ7

Table 4−9. Multifunction Routing Register (Continued)

BIT	SIGNAL	TYPE	FUNCTION
7−4	MFUNC1	R/W	Multifunction terminal 1 configuration. These bits control the internal signal mapped to the MFUNC1 terminal as follows: NOTE: When the serial bus mode is implemented by pulling up the $\overline{\text{VCCD0}}$ and $\overline{\text{VCCD1}}$ terminals, the MFUNC1 terminal provides the SDA signaling. 0000 = GPI1 0100 = IRQ4 1000 = CAUDPWM 1100 = LED_SKT 0001 = $\overline{\text{GPO1}}$ 0101 = IRQ5 1001 = IRQ9 1101 = $\overline{\text{IRQ13}}$ 0010 = D3_STAT 0110 = $\overline{\text{ZVSTAT}}$ 1010 = IRQ10 1110 = $\overline{\text{GPE}}$ 0011 = IRQ3 0111 = $\overline{\text{ZVSEL0}}$ 1011 = IRQ11 1111 = IRQ15
3−0	MFUNC0	R/W	Multifunction terminal 0 configuration. These bits control the internal signal mapped to the MFUNC0 terminal as follows: 0000 = GPI0 0100 = IRQ4 1000 = CAUDPWM 1100 = LED_SKT 0001 = $\overline{\text{GPO0}}$ 0101 = IRQ5 1001 = IRQ9 1101 = $\overline{\text{IRQ13}}$ 0010 = $\overline{\text{INTA}}$ 0110 = $\overline{\text{ZVSTAT}}$ 1010 = IRQ10 1110 = $\overline{\text{GPE}}$ 0011 = IRQ3 0111 = $\overline{\text{ZVSEL0}}$ 1011 = IRQ11 1111 = IRQ15

4.33 Retry Status Register

The retry status register enables the retry timeout counters and displays the retry expiration status. The flags are set when the PCI4410 retries a PCI or CardBus master request and the master does not return within 2^{15} PCI clock cycles. The flags are cleared by writing a 1 to the bit. These bits are expected to be incorporated into the PCI command, PCI status, and bridge control registers by the PCI SIG. See Table 4−10 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Retry status							
Type	R/W	R/W	R	R	R/C	R	R/C	R
Default	1	1	0	0	0	0	0	0

Register: **Retry status**
Type: Read-only, Read/Write, Read/Write to Clear
Offset: 90h
Default: C0h

Table 4−10. Retry Status Register

BIT	SIGNAL	TYPE	FUNCTION
7	PCIRETRY	R/W	PCI retry timeout counter enable. Bit 7 is encoded: 0 = PCI retry counter disabled 1 = PCI retry counter enabled (default)
6	CBRETRY	R/W	CardBus retry timeout counter enable. Bit 6 is encoded: 0 = CardBus retry counter disabled 1 = CardBus retry counter enabled (default)
5−4	RSVD	R	Reserved. Bits 5 and 4 return 0s when read.
3	TEXP_CB	R/C	CardBus target retry expired. Write a 1 to clear bit 3. 0 = Inactive (default) 1 = Retry has expired.
2	RSVD	R	Reserved. Bit 2 returns 0 when read.
1	TEXP_PCI	R/C	PCI target retry expired. Write a 1 to clear bit 1. 0 = Inactive (default) 1 = Retry has expired.
0	RSVD	R	Reserved. Bit 0 returns 0 when read.

4.34 Card Control Register

The card control register is provided for PCI1130 compatibility. $\overline{\text{RI\_OUT}}$ is enabled through this register. See Table 4−11 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Card control							
Type	R/W	R/W	R/W	R	R	R/W	R/W	R/C
Default	0	0	0	0	0	0	0	0

Register: **Card control**
Type: Read-only, Read/Write, Read/Write to Clear
Offset: 91h
Default: 00h

Table 4−11. Card Control Register

BIT	SIGNAL	TYPE	FUNCTION
7	RIENB	R/W	Ring indicate output enable. 0 = Disables any routing of $\overline{\text{RI\_OUT}}$ signal (default) 1 = Enables $\overline{\text{RI\_OUT}}$ signal for routing to the $\overline{\text{RI\_OUT}}/\overline{\text{PME}}$ terminal, when bit 0 (RIMUX) in the system control register (see Section 4.29) is set to 0, and for routing to MFUNC2 or MFUNC4.
6	ZVENABLE	R/W	Compatibility ZV mode enable. When set, the PC Card socket interface ZV terminals enter a high-impedance state. This bit defaults to 0.
5	ZV PORT_ENABLE	R/W	ZV output port enable. When bit 5 is set, the ZV output port is enabled. If bit 6 (ZVENABLE) is set, then ZV data from the PC Card interface is routed to the ZV output port. Otherwise, the ZV output port drives a stable 0 pattern on all pins. When bit 5 is not set, the ZV output port pins are placed in a high-impedance state. Default is 0.
4−3	RSVD	R	Reserved. Bits 4 and 3 return 0 when read.
2	AUD2MUX	R/W	CardBus audio-to-IRQMUX. When set, the CAUDIO CardBus signal is routed to the corresponding multifunction terminal which may be configured for CAUDPWM.
1	SPKROUTEN	R/W	Speaker out enable. When bit 1 is set, $\overline{\text{SPKR}}$ on the PC Card is enabled and is routed to SPKROUT. The SPKROUT terminal drives data only when the socket's SPKROUTEN bit is set. This bit is encoded as: 0 = $\overline{\text{SPKR}}$ to SPKROUT not enabled (default) 1 = $\overline{\text{SPKR}}$ to SPKROUT enabled
0	IFG	R/C	Interrupt flag. Bit 0 is the interrupt flag for 16-bit I/O PC Cards and for CardBus cards. Bit 0 is set when a functional interrupt is signaled from a PC Card interface. Write back a 1 to clear this bit. 0 = No PC Card functional interrupt detected (default) 1 = PC Card functional interrupt detected

4.35 Device Control Register

The device control register is provided for PCI1130 compatibility. The interrupt mode select and the socket-capable force bits are programmed through this register. See Table 4−12 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Device control							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	1	1	0	0	1	1	0

Register: **Device control**
Type: Read-only, Read/Write
Offset: 92h
Default: 66h

Table 4−12. Device Control Register

BIT	SIGNAL	TYPE	FUNCTION
7	SKTPWR_LOCK	R/W	Socket power lock bit. When this bit is set to 1, software will not be able to power down the PC Card socket while in D3. This may be necessary to support wake on LAN or RING if the operating system is programmed to power down a socket when the CardBus controller is placed in the D3 state.
6	3VCAPABLE	R/W	3-V socket capable force 0 = Not 3-V capable 1 = 3-V capable (default)
5	IO16V2	R/W	Diagnostic bit. This bit defaults to 1.
4	BUS_HOLDER_EN	R/W	Bus holder cell enable/disable. Setting bit 4 to 1 enables the bus holder cells on the 1394 link interface. Default state is 0, bus holder cells disabled.
3	TEST	R/W	TI test. Only a 0 should be written to bit 3.
2−1	INTMODE	R/W	Interrupt signaling mode. Bits 2 and 1 select the interrupt signaling mode. The interrupt signaling mode bits are encoded: 00 = Parallel PCI interrupts only 01 = Parallel IRQ and parallel PCI interrupts 10 = IRQ serialized interrupts and parallel PCI interrupt 11 = IRQ and PCI serialized interrupts (default)
0	RSVD	R/W	Reserved. Bit 0 is reserved for test purposes. Only 0 should be written to this bit.

4.36 Diagnostic Register

The diagnostic register is provided for internal TI test purposes. In additon, the diagnostic register can be used to control CSC interrupt routing, enable asynchronous interrupts, and alter the PCI vendor ID and device ID register fields. See Table 4−13 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Diagnostic							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	1	0	0	0	0	1

Register: **Diagnostic**
Type: Read/Write
Offset: 93h
Default: 61h

Table 4−13. Diagnostic Register

BIT	SIGNAL	TYPE	FUNCTION
7	TRUE_VAL	R/W	This bit defaults to 0. Ths bit will cause software to fail to recognize the PCI4410 when set to 1. This bit is encoded as: 0 = Reads true values from the PCI vendor ID and PCI device ID registers (default) 1 = Reads all 1s from the PCI vendor ID and PCI device ID registers
6	RSVD	R/W	Reserved. Bit 6 returns 0 when read.
5	CSC	R/W	CSC interrupt routing control 0 = CSC interrupts routed to PCI if ExCA 803 (see Section 5.4) bit 4 = 1. 1 = CSC interrupts routed to PCI if ExCA 805 (see Section 5.6) bits 7−4 = 0000b (default). In this case, the setting of ExCA 803 bit 4 is a don't care.
4	DIAG4	R/W	Diagnostic RETRY_DIS. Delayed transaction disabled.
3	DIAG3	R/W	Diagnostic RETRY_EXT. Extends the latency from 16 to 64.
2	DIAG2	R/W	Diagnostic DISCARD_TIM_SEL_CB. Set = 2^{10}, reset = 2^{15}.
1	DIAG1	R/W	Diagnostic DISCARD_TIM_SEL_PCI. Set = 2^{10}, reset = 2^{15}.
0	ASYNCINT	R/W	Asynchronous interrupt enable. 0 = CSC interrupt is not generated asynchronously. 1 = CSC interrupt is generated asynchronously (default).

4.37 Socket DMA Register 0

The socket DMA register 0 provides control over the PC Card DMA request ($\overline{\text{DREQ}}$) signaling. See Table 4−14 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Socket DMA register 0															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Socket DMA register 0															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Socket DMA register 0**
Type: Read-only, Read/Write
Offset: 94h
Default: 0000 0000h

Table 4−14. Socket DMA Register 0

BIT	SIGNAL	TYPE	FUNCTION
31−2	RSVD	R	Reserved. Bits 31−2 return 0s when read.
1−0	DREQPIN	R/W	DMA request ($\overline{\text{DREQ}}$). Bits 1 and 0 indicate which pin on the 16-bit PC Card interface acts as $\overline{\text{DREQ}}$ during DMA transfers. This field is encoded as: 00 = $\overline{\text{Socket}}$ not configured for DMA (default). 01 = $\overline{\text{DREQ}}$ uses $\overline{\text{SPKR}}$. 10 = $\overline{\text{DREQ}}$ uses $\overline{\text{IOIS16}}$. 11 = DREQ uses INPACK.

4.38 Socket DMA Register 1

The socket DMA register 1 provides control over the distributed DMA (DDMA) registers and the PCI portion of DMA transfers. The DMA base address locates the DDMA registers in a 16-byte region within the first 64K bytes of PCI I/O address space. See Table 4−15 for a complete description of the register contents.

NOTE: 32-bit transfers are not supported; the maximum transfer possible for 16-bit PC Cards is 16 bits.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Socket DMA register 1															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Socket DMA register 1															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Socket DMA register 1**
Type: Read-only, Read/Write
Offset: 98h
Default: 0000 0000h

Table 4−15. Socket DMA Register 1

BIT	SIGNAL	TYPE	FUNCTION
31−16	RSVD	R	Reserved. Bits 31−16 return 0s when read.
15−4	DMABASE	R/W	DMA base address. Locates the socket's DMA registers in PCI I/O space. This field represents a 16-bit PCI I/O address. The upper 16 bits of the address are hardwired to 0, forcing this window to within the lower 64K bytes of I/O address space. The lower 4 bits are hardwired to 0 and are included in the address decode. Thus, the window is aligned to a natural 16-byte boundary.
3	EXTMODE	R	Extended addressing. This feature is not supported by the PCI4410 and always returns a 0.
2−1	XFERSIZE	R/W	Transfer size. Bits 2 and 1 specify the width of the DMA transfer on the PC Card interface and are encoded as: 00 = Transfers are 8 bits (default). 01 = Transfers are 16 bits. 10 = Reserved 11 = Reserved
0	DDMAEN	R/W	DDMA registers decode enable. Enables the decoding of the distributed DMA registers based on the value of bits 15−4 (DMABASE field). 0 = Disabled (default) 1 = Enabled

4.39 Capability ID Register

The capability ID register identifies the linked list item as the register for PCI power management. The register returns 01h when read, which is the unique ID assigned by the PCI SIG for the PCI location of the capabilities pointer and the value.

Bit	7	6	5	4	3	2	1	0
Name	Capability ID							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	1

Register: **Capability ID**
Type: Read-only
Offset: A0h
Default: 01h

4.40 Next-Item Pointer Register

The next-item pointer register indicates the next item in the linked list of the PCI power management capabilities. Because the PCI4410 functions include only one capabilities item, this register returns 0s when read.

Bit	7	6	5	4	3	2	1	0
Name	Next-item pointer							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **Next-item pointer**
Type: Read-only
Offset: A1h
Default: 00h

4.41 Power Management Capabilities Register

This register contains information on the capabilities of the PC Card function related to power management. Both PCI4410 CardBus bridge functions support D0, D1, D2, and D3 power states. See Table 4−16 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Power management capabilities															
Type	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	1	1	1	1	1	1	1	0	0	0	1	1	0	0	0	1

Register: **Power management capabilities**
Type: Read/Write, Read-only
Offset: A2h
Default: FE31h

Table 4−16. Power Management Capabilities Register

BIT	SIGNAL	TYPE	FUNCTION
15	PME_SUPPORT	R/W	$\overline{\text{PME}}$ support. This 5-bit field indicates the power states from which the PCI4410 device functions may assert $\overline{\text{PME}}$. A 0 (zero) for any bit indicates that the function cannot assert the $\overline{\text{PME}}$ signal while in that power state. These five bits return 11111b when read. Each of these bits is described below: Bit 15 defaults to the value 1 indicating the $\overline{\text{PME}}$ signal can be asserted from the $D3_{cold}$ state. This bit is R/W because wake-up support from $D3_{cold}$ is contingent on the system providing an auxiliary power source to the V_{CC} terminals. If the system designer chooses not to provide an auxiliary power source to the V_{CC} terminals for $D3_{cold}$ wake-up support, then BIOS should write a 0 to this bit.
14−11	PME_SUPPORT	R	Bit 14 contains the value 1, indicating that the $\overline{\text{PME}}$ signal can be asserted from $D3_{hot}$ state. Bit 13 contains the value 1, indicating that the $\overline{\text{PME}}$ signal can be asserted from D2 state. Bit 12 contains the value 1, indicating that the $\overline{\text{PME}}$ signal can be asserted from D1 state. Bit 11 contains the value 1, indicating that the $\overline{\text{PME}}$ signal can be asserted from the D0 state.
10	D2_SUPPORT	R	D2 support. Bit 10 returns a 1 when read, indicating that the CardBus function supports the D2 device power state.
9	D1_SUPPORT	R	D1 support. Bit 9 returns a 1 when read, indicating that the CardBus function supports the D1 device power state.
8−6	RSVD	R	Reserved. Bits 8−6 return 0s when read.
5	DSI	R	Device-specific initialization. Bit 5 returns 1 when read, indicating that the CardBus controller function requires special initialization (beyond the standard PCI configuration header) before the generic class device driver is able to use it.
4	AUX_PWR	R	Auxiliary power source. Bit 4 is meaningful only if bit 15 (PME_Support, $D3_{cold}$) is set. When bit 4 is set, it indicates that support for $\overline{\text{PME}}$ in $D3_{cold}$ requires auxiliary power supplied by the system by way of a proprietary delivery vehicle. When bit 4 is 0, it indicates that the function supplies its own auxiliary power source. Because the PCI4410 requires an auxiliary power supply, this bit returns 1.
3	PMECLK	R	$\overline{\text{PME}}$ clock. Bit 3 returns 0 when read, indicating that no host bus clock is required for the PCI4410 to generate $\overline{\text{PME}}$.
2−0	VERSION	R	Version. Bits 2−0 return 001b when read, indicating that there are four bytes of general-purpose power management (PM) registers as described in the *PCI Bus Power Management Interface Specification*. See system control register (PCI offset 80h) (Section 4.29), PCIPMEN bit 23, for additional information. It is recommended that the PCIPMEN bit be set by BIOS. If PCIPMEN is set, then VERSION bits 2−0 will return 010b, indicating support for version 1.1 of the *PCI Bus Power Management Interface Specification.*

4.42 Power Management Control/Status Register

The power management control/status register determines and changes the current power state of the PCI4410 CardBus function. The contents of this register are not affected by the internally generated reset caused by the transition from $D3_{hot}$ to D0 state. All PCI, ExCA, and CardBus registers are reset as a result of a $D3_{hot}$ to D0 state transition. TI-specific registers, PCI power management registers, and the legacy base address register are not reset. See Table 4−17 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Power management control/status															
Type	R/C	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Power management control/status**
Type: Read-only, Read/Write, Read/Write to Clear
Offset: A4h
Default: 0000h

Table 4−17. Power Management Control/Status Register

BIT	SIGNAL	TYPE	FUNCTION
15	PMESTAT	R/C	$\overline{\text{PME}}$ status. Bit 15 is set when the CardBus function would normally assert $\overline{\text{PME}}$, independent of the state of bit 8 (PME_EN). Bit 15 is cleared by a writeback of 1, and this also clears the $\overline{\text{PME}}$ signal if $\overline{\text{PME}}$ was asserted by this function. Writing a 0 to this bit has no effect.
14−13	DATASCALE	R	Data scale. This 2-bit field returns 0s when read. The CardBus function does not return any dynamic data as indicated by bit 4 (DYN_DATA_PME_EN).
12−9	DATASEL	R	Data select. This 4-bit field returns 0s when read. The CardBus function does not return any dynamic data as indicated by bit 4 (DYN_DATA_PME_EN).
8	PME_EN	R/W	$\overline{\text{PME}}$ enable. When set to 1, bit 8 enables the function to assert $\overline{\text{PME}}$. When reset to 0, the assertion of $\overline{\text{PME}}$ is disabled.
7−5	RSVD	R	Reserved. Bits 7−5 return 0s when read.
4	DYN_DATA_PME_EN	R	Dynamic data $\overline{\text{PME}}$ enable. Bit 4 returns 0 when read because the CardBus function does not report dymanic data.
3−2	RSVD	R	Reserved. Bits 3−2 return 0s when read.
1−0	PWR_STATE	R/W	Power state. This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state. This field is encoded as: 00 = D0 01 = D1 10 = D2 11 = $D3_{hot}$

4.43 Power Management Control/Status Register Bridge Support Extensions

The power management control/status register bridge support extensions support PCI-bridge-specific functionality. See Table 4−18 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Power management control/status register bridge support extensions							
Type	R	R/W	R	R	R	R	R	R
Default	1	1	0	0	0	0	0	0

Register: **Power management control/status register bridge support extensions**
Type: Read-only
Offset: A6h
Default: C0h

Table 4−18. Power Management Control/Status Register Bridge Support Extensions

BIT	SIGNAL	TYPE	FUNCTION
7	BPCC_EN	R	BPCC_Enable. Bus power/clock control enable. This bit returns 1 when read. This bit is encoded as: 0 = Bus power/clock control is disabled. 1 = Bus power/clock control is enabled (default). A 0 indicates that the bus power/clock control policies defined in the *PCI Bus Power Management Interface Specification* are disabled. When the bus power/clock control enable mechanism is disabled, the bridge's power management control/status register power state field (see Section 4.42, bits 1−0) cannot be used by the system software to control the power or the clock of the bridge's secondary bus. A 1 indicates that the bus power/clock control mechanism is enabled.
6	B2_$\overline{B3}$	R/W	B2/B3 support for $D3_{hot}$. The state of this bit determines the action that is to occur as a direct result of programming the function to $D3_{hot}$. This bit is only meaningful if bit 7 (BPCC_EN) is a 1. This bit is encoded as: 0 = When the bridge is programmed to $D3_{hot}$, its secondary bus will have its power removed (B3). 1 = When the bridge function is programmed to $D3_{hot}$, its secondary bus's PCI clock will be stopped (B2). (Default)
5−0	RSVD	R	Reserved. Bits 5−0 return 0s when read.

4.44 Power Management Data Register

The power management data register returns 0s when read, because the CardBus functions do not report dynamic data.

Bit	7	6	5	4	3	2	1	0
Name	Power management data							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **Power management data**
Type: Read-only
Offset: A7h
Default: 00h

4.45 General-Purpose Event Status Register

The general-purpose event status register contains status bits that are set by different events. The bits in this register and the corresponding $\overline{\text{GPE}}$ are cleared by writing a 1 to the corresponding bit location. See Table 4−19 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	General-purpose event status															
Type	R/C	R	R	R	R/C	R	R	R/C	R	R	R	R/C	R/C	R/C	R/C	R/C
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **General-purpose event status**
Type: Read-only, Read/Write to Clear
Offset: A8h
Default: 0000h

Table 4−19. General-Purpose Event Status Register

BIT	SIGNAL	TYPE	FUNCTION
15	ZV_STS	R/C	PC card ZV status. Bit 15 is set on a change in status of bit 6 (ZVENABLE) in the card control register (see Section 4.34).
14−12	RSVD	R	Reserved. Bits 14−12 return 0s when read.
11	PWR_STS	R/C	Power change status. Bit 11 is set when software has changed the power state of the socket. A change in either V_{CC} or V_{PP} for the socket causes this bit to be set.
10−9	RSVD	R	Reserved. Bits 10 and 9 return 0s when read.
8	VPP12_STS	R/C	12-V V_{PP} request status. Bit 8 is set when software has changed the requested Vpp level to or from 12 V for the PC Card socket.
7−5	RSVD	R	Reserved. Bits 7−5 return 0s when read.
4	GP4_STS	R/C	GPI4 Status. Bit 4 is set on a change in status of the MFUNC5 terminal input level. This bit does not depend upon the state of a corresponding bit in the general-purpose event enable register.
3	GP3_STS	R/C	GPI3 Status. Bit 3 is set on a change in status of the MFUNC4 terminal input level. This bit does not depend upon the state of a corresponding bit in the general-purpose event enable register.
2	GP2_STS	R/C	GPI2 Status. Bit 2 is set on a change in status of the MFUNC2 terminal input level. This bit does not depend upon the state of a corresponding bit in the general-purpose event enable register.
1	GP1_STS	R/C	GPI1 Status. Bit 1 is set on a change in status of the MFUNC1 terminal input level. This bit does not depend upon the state of a corresponding bit in the general-purpose event enable register.
0	GP0_STS	R/C	GPI0 Status. Bit 0 is set on a change in status of the MFUNC0 terminal input level. This bit does not depend upon the state of a corresponding bit in the general-purpose event enable register.

4.46 General-Purpose Event Enable Register

The general-purpose event enable register contains bits that are set to enable a $\overline{\text{GPE}}$ signal. The $\overline{\text{GPE}}$ signal is driven until the corresponding status bit is cleared and the event is serviced. The $\overline{\text{GPE}}$ can only be signaled if one of the multifunction terminals, MFUNC6−MFUNC0, is configured for $\overline{\text{GPE}}$ signaling. See Table 4−20 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	General-purpose event enable															
Type	R/W	R	R	R	R/W	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **General-purpose event enable**
Type: Read-only, Read/Write
Offset: AAh
Default: 0000h

Table 4−20. General-Purpose Event Enable Register

BIT	SIGNAL	TYPE	FUNCTION
15	ZV_EN	R/W	PC card socket ZV enable. When bit 15 is set, a $\overline{\text{GPE}}$ is signaled on a change in status of bit 6 (ZVENABLE) in the card control register (see Section 4.34).
14−12	RSVD	R	Reserved. Bits 14−12 return 0s when read.
11	PWR_EN	R/W	Power change enable. When bit 11 is set, a $\overline{\text{GPE}}$ is signaled when software has changed the power state of the socket.
10−9	RSVD	R	Reserved. Bits 10 and 9 return 0s when read.
8	VPP12_EN	R/W	12 V V_{PP} request enable. When bit 8 is set, a $\overline{\text{GPE}}$ is signaled when software has changed the requested V_{PP} level to or from 12 V for the card socket.
7−5	RSVD	R	Reserved. Bits 7−5 return 0s when read.
4	GP4_EN	R/W	GPI4 enable. When bit 4 is set, a $\overline{\text{GPE}}$ is signaled when there has been a change in status of the MFUNC5 terminal input level if configured as GPI4.
3	GP3_EN	R/W	GPI3 enable. When bit 3 is set, a $\overline{\text{GPE}}$ is signaled when there has been a change in status of the MFUNC4 terminal input level if configured as GPI3.
2	GP2_EN	R/W	GPI2 enable. When bit 2 is set, a $\overline{\text{GPE}}$ is signaled when there has been a change in status of the MFUNC2 terminal input if configured as GPI2.
1	GP1_EN	R/W	GPI1 enable. When bit 1 is set, a $\overline{\text{GPE}}$ is signaled when there has been a change in status of the MFUNC1 terminal input if configured as GPI1.
0	GP0_EN	R/W	GPI0 enable. When bit 0 is set, a $\overline{\text{GPE}}$ is signaled when there has been a change in status of the MFUNC0 terminal input if configured as GPI0.

4.47 General-Purpose Input Register

The general-purpose input register provides the logical value of the data input from the GPI terminals, MFUNC5, MFUNC4, and MFUNC2−MFUNC0. See Table 4−21 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	General-purpose input															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X

Register: **General-purpose input**
Type: Read-only
Offset: ACh
Default: 00XXh

Table 4−21. General-Purpose Input Register

BIT	SIGNAL	TYPE	FUNCTION
15−5	RSVD	R	Reserved. Bits 15−5 return 0s when read.
4	GPI4_DATA	R	GPI4 data bit. The value read from bit 4 represents the logical value of the data input from the MFUNC5 terminal.
3	GPI3_DATA	R	GPI3 data bit. The value read from bit 3 represents the logical value of the data input from the MFUNC4 terminal.
2	GPI2_DATA	R	GPI2 data bit. The value read from bit 2 represents the logical value of the data input from the MFUNC2 terminal.
1	GPI1_DATA	R	GPI1 data bit. The value read from bit 1 represents the logical value of the data input from the MFUNC1 terminal.
0	GPI0_DATA	R	GPI0 data bit. The value read from bit 0 represents the logical value of the data input from the MFUNC0 terminal.

4.48 General-Purpose Output Register

The general-purpose output register is used for control of the general-purpose outputs. See Table 4−22 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	General-purpose output															
Type	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **General-purpose output**
Type: Read-only, Read/Write
Offset: AEh
Default: 0000h

Table 4−22. General-Purpose Output Register

BIT	SIGNAL	TYPE	FUNCTION
15−5	RSVD	R	Reserved. Bits 15−5 return 0s when read.
4	GPO4_DATA	R/W	GPO4 data bit. The value written to bit 4 represents the logical value of the data driven to the MFUNC5 terminal if configured as GPO4. Read transactions return the last data value written.
3	GPO3_DATA	R/W	GPIO3 data bit. The value written to bit 3 represents the logical value of the data driven to the MFUNC4 terminal if configured as GPO3. Read transactions return the last data value written.
2	GPO2_DATA	R/W	GPO2 data bit. The value written to bit 2 represents the logical value of the data driven to the MFUNC2 terminal if configured as GPO2. Read transactions return the last data value written.
1	GPO1_DATA	R/W	GPO1 data bit. The value written to bit 1 represents the logical value of the data driven to the MFUNC1 terminal if configured as GPO1. Read transactions return the last data value written.
0	GPO0_DATA	R/W	GPO0 data bit. The value written to bit 0 represents the logical value of the data driven to the MFUNC0 terminal if configured as GPO0. Read transactions return the last data value written.

5 ExCA Compatibility Registers

The ExCA registers implemented in the PCI4410 are register-compatible with the Intel 82365SL−DF PCMCIA controller. ExCA registers are identified by an offset value that is compatible with the legacy I/O index/data scheme used on the Intel 82365 ISA controller. The ExCA registers are accessed through this scheme by writing the register offset value into the index register (I/O base) and reading or writing the data register (I/O base + 1). The I/O base address used in the index/data scheme is programmed in the PC Card 16-bit I/F legacy-mode base address register (see Section 4.28). The offsets from this base address run contiguously from 00h to 3Fh for the socket. See Figure 5−1 for an ExCA I/O mapping illustration.

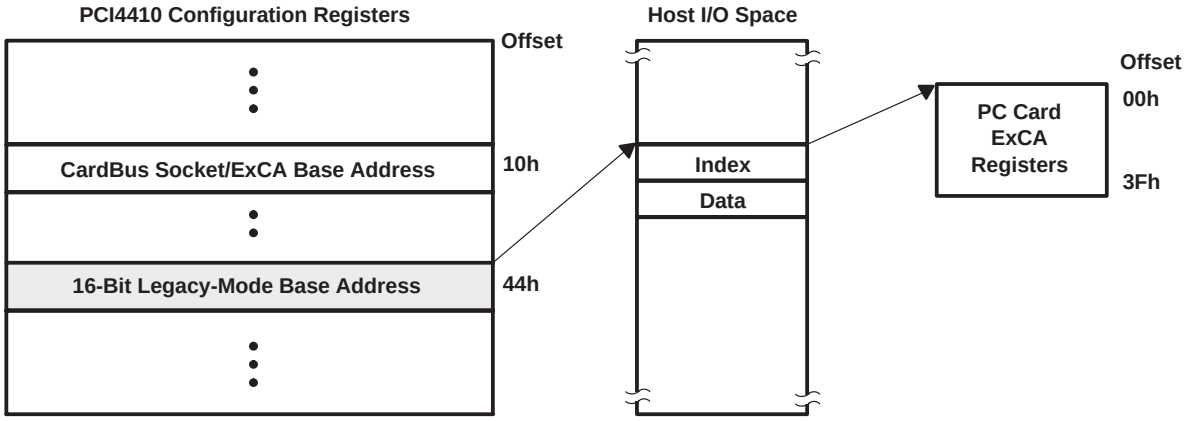


Figure 5−1. ExCA Register Access Through I/O

The TI PCI4410 also provides a memory-mapped alias of the ExCA registers by directly mapping them into PCI memory space. They are located through the CardBus socket/ExCA base address register (see Section 4.12) at memory offset 800h. See Figure 5−2 for an ExCA memory mapping illustration. This illustration also identifies the CardBus socket register mapping, which is mapped into the same 4K window at memory offset 0h.

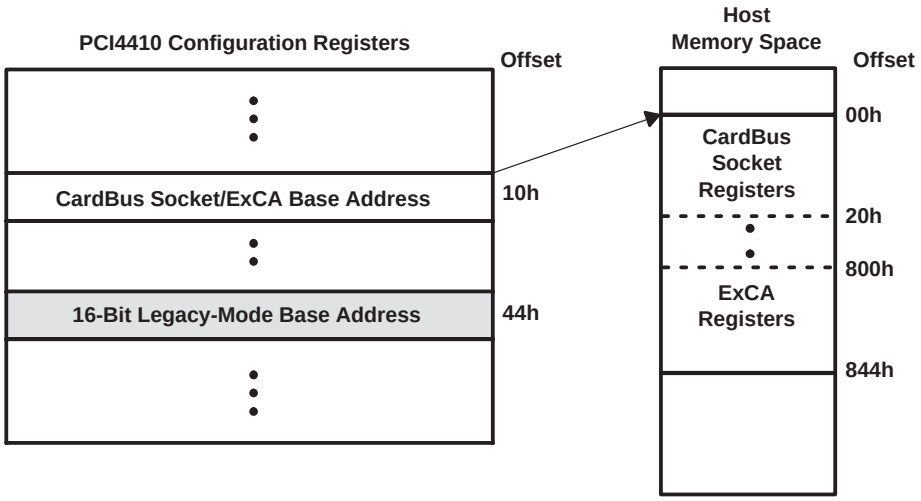


Figure 5−2. ExCA Register Access Through Memory

As defined by the 82365SL−DL Specification, the interrupt registers in the ExCA register set control such card functions as reset, type, interrupt routing, and interrupt enables. Special attention must be paid to the interrupt routing registers and the host interrupt signaling method selected for the PCI4410 to ensure that all possible PCI4410 interrupts can potentially be routed to the programmable interrupt controller. The ExCA registers that are critical to the interrupt signaling are the ExCA interrupt and general control register (see Section 5.4) and the ExCA card status-change-interrupt configuration register (see Section 5.6).

Access to I/O mapped 16-bit PC Cards is available to the host system via two ExCA I/O windows. These are regions of host I/O address space into which the card I/O space is mapped. These windows are defined by start, end, and offset addresses programmed in the ExCA registers described in this section. I/O windows have byte granularity.

Access to memory mapped 16-bit PC Cards is available to the host system via five ExCA memory windows. These are regions of host memory space into which the card memory space is mapped. These windows are defined by start, end, and offset addresses programmed in the ExCA registers described in this section. Table 5−1 identifies each ExCA register and its respective ExCA offset. Memory windows have 4-Kbyte granularity.

Table 5−1. ExCA Registers and Offsets

ExCA REGISTER NAME	CARDBUS SOCKET ADDRESS OFFSET (HEX)	ExCA OFFSET (HEX)
Identification and revision	800	00
Interface status	801	01
Power control	802	02
Interrupt and general control	803	03
Card status change	804	04
Card status-change-interrupt configuration	805	05
Address window enable	806	06
I / O window control	807	07
I / O window 0 start-address low byte	808	08
I / O window 0 start-address high byte	809	09
I / O window 0 end-address low byte	80A	0A
I / O window 0 end-address high byte	80B	0B
I / O window 1 start-address low byte	80C	0C
I / O window 1 start-address high byte	80D	0D
I / O window 1 end-address low byte	80E	0E
I / O window 1 end-address high byte	80F	0F
Memory window 0 start-address low byte	810	10
Memory window 0 start-address high byte	811	11
Memory window 0 end-address low byte	812	12
Memory window 0 end-address high byte	813	13
Memory window 0 offset-address low byte	814	14
Memory window 0 offset-address high byte	815	15
Card detect and general control	816	16
Reserved	817	17
Memory window 1 start-address low byte	818	18
Memory window 1 start-address high byte	819	19
Memory window 1 end-address low byte	81A	1A
Memory window 1 end-address high byte	81B	1B
Memory window 1 offset-address low byte	81C	1C
Memory window 1 offset-address high byte	81D	1D

Table 5−1. ExCA Registers and Offsets (Continued)

ExCA REGISTER NAME	CARDBUS SOCKET ADDRESS OFFSET (HEX)	ExCA OFFSET (HEX)
Global control	81E	1E
Reserved	81F	1F
Memory window 2 start-address low byte	820	20
Memory window 2 start-address high byte	821	21
Memory window 2 end-address low byte	822	22
Memory window 2 end-address high byte	823	23
Memory window 2 offset-address low byte	824	24
Memory window 2 offset-address high byte	825	25
Reserved	826	26
Reserved	827	27
Memory window 3 start-address low byte	828	28
Memory window 3 start-address high byte	829	29
Memory window 3 end-address low byte	82A	2A
Memory window 3 end-address high byte	82B	2B
Memory window 3 offset-address low byte	82C	2C
Memory window 3 offset-address high byte	82D	2D
Reserved	82E	2E
Reserved	82F	2F
Memory window 4 start-address low byte	830	30
Memory window 4 start-address high byte	831	31
Memory window 4 end-address low byte	832	32
Memory window 4 end-address high byte	833	33
Memory window 4 offset-address low byte	834	34
Memory window 4 offset-address high byte	835	35
I/O window 0 offset-address low byte	836	36
I/O window 0 offset-address high byte	837	37
I/O window 1 offset-address low byte	838	38
I/O window 1 offset-address high byte	839	39
Reserved	83A	3A
Reserved	83B	3B
Reserved	83C	3C
Reserved	83D	3D
Reserved	83E	3E
Reserved	83F	3F
Memory window page 0	840	−
Memory window page 1	841	−
Memory window page 2	842	−
Memory window page 3	843	−
Memory window page 4	844	−

5.1 ExCA Identification and Revision Register

The ExCA identification and revision register provides host software with information on 16-bit PC Card support and Intel 82365SL-DF compatibility. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register (see Section 4.29). See Table 5−2 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA identification and revision							
Type	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Default	1	0	0	0	0	1	0	0

Register: **ExCA identification and revision**
Type: Read-only, Read/Write
Offset: CardBus socket address + 800h; ExCA offset 00h
Default: 84h

Table 5−2. ExCA Identification and Revision Register

BIT	SIGNAL	TYPE	FUNCTION
7−6	IFTYPE	R	Interface type. These bits, which are hardwired as 10b, identify the 16-bit PC Card support provided by the PCI4410. The PCI4410 supports both I/O and memory 16-bit PC cards.
5−4	RSVD	R/W	Reserved.
3−0	365REV	R/W	Intel 82365SL-DF revision. This field stores the Intel 82365SL-DF revision supported by the PCI4410. Host software can read this field to determine compatibility to the Intel 82365SL-DF register set. Writing 0010b to this field puts the controller in 82365SL mode. This field defaults to 0100b upon PCI4410 reset.

5.2 ExCA Interface Status Register

The ExCA interface status register provides information on the current status of the PC Card interface. An X in the default bit value indicates that the value of the bit after reset depends on the state of the PC Card interface. See Table 5−3 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA interface status							
Type	R	R	R	R	R	R	R	R
Default	0	0	X	X	X	X	X	X

Register: **ExCA interface status**
Type: Read-only
Offset: CardBus socket address + 801h; ExCA offset 01h
Default: 00XX XXXXb

Table 5−3. ExCA Interface Status Register

BIT	SIGNAL	TYPE	FUNCTION
7	RSVD	R	Reserved. Bit 7 returns 0 when read.
6	CARDPWR	R	Card Power. Bit 6 indicates the current power status of the PC Card socket. This bit reflects how the ExCA power control register (see Section 5.3) is programmed. Bit 6 is encoded as: 0 = V_{CC} and V_{PP} to the socket turned off (default) 1 = V_{CC} and V_{PP} to the socket turned on
5	READY	R	Ready. Bit 5 indicates the current status of the READY signal at the PC Card interface. 0 = PC Card not ready for data transfer 1 = PC Card ready for data transfer
4	CARDWP	R	Card write protect. Bit 4 indicates the current status of WP at the PC Card interface. This signal reports to the PCI4410 whether or not the memory card is write protected. Furthermore, write protection for an entire PCI4410 16-bit memory window is available by setting the appropriate bit in the ExCA memory window offset-address high-byte register. 0 = WP is 0. PC Card is read/write. 1 = WP is 1. PC Card is read-only.
3	CDETECT2	R	Card detect 2. Bit 3 indicates the status of $\overline{CD2}$ at the PC Card interface. Software may use this and bit 2 (CDETECT1) to determine if a PC Card is fully seated in the socket. 0 = $\overline{CD2}$ is 1. No PC Card is inserted. 1 = $\overline{CD2}$ is 0. PC Card is at least partially inserted.
2	CDETECT1	R	Card detect 1. Bit 2 indicates the status of $\overline{CD1}$ at the PC Card interface. Software may use this and bit 3 (CDETECT2) to determine if a PC Card is fully seated in the socket. 0 = $\overline{CD1}$ is 1. No PC Card is inserted. 1 = $\overline{CD1}$ is 0. PC Card is at least partially inserted.
1−0	BVDSTAT	R	Battery voltage detect. When a 16-bit memory card is inserted, the field indicates the status of the battery voltage detect signals (BVD1, BVD2) at the PC Card interface, where bit 1 reflects the BVD2 status and bit 0 reflects BVD1. 00 = Battery dead 01 = Battery dead 10 = Battery low; warning 11 = Battery good When a 16-bit I/O card is inserted, this field indicates the status of $\overline{SPKR}$ (bit 1) and $\overline{STSCHG}$ (bit 0) at the PC Card interface. In this case, the two bits in this field directly reflect the current state of these card outputs.

5.3 ExCA Power Control Register

The ExCA power control register provides PC Card power control. Bit 7 (COE) of this register controls the 16-bit output enables on the socket interface, and can be used for power management in 16-bit PC Card applications. See Table 5−4 and Table 5−5 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA power control							
Type	R/W	R	R/W	R/W	R/W	R	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA power control**
Type: Read-only, Read/Write
Offset: CardBus socket address + 802h; ExCA offset 02h
Default: 00h

Table 5−4. ExCA Power Control Register 82365SL Support

BIT	SIGNAL	TYPE	FUNCTION
7	COE	R/W	Card output enable. Bit 7 controls the state of all of the 16-bit outputs on the PCI4410. This bit is encoded as: 0 = 16-bit PC Card outputs disabled (default) 1 = 16-bit PC Card outputs enabled
6	RSVD	R	Reserved. Bit 6 returns 0 when read.
5	AUTOPWRSWEN	R/W	Auto power switch enable. This bit is enabled by bit 7 of the system control register (see Section 4.29). 0 = Automatic socket power switching based on card detects is disabled. 1 = Automatic socket power switching based on card detects is enabled.
4	CAPWREN	R/W	PC Card power enable. 0 = V_{CC} = V_{PP1} = V_{PP2} = No connection 1 = V_{CC} is enabled and controlled by bit 2 (ExCAPower) of the system control register (see Section 4.29), V_{PP1} and V_{PP2} are controlled according to bits 1−0 (EXCAVPP field).
3−2	RSVD	R	Reserved. Bits 3 and 2 return 0s when read.
1−0	EXCAVPP	R/W	PC Card V_{PP} power control. Bits 1 and 0 are used to request changes to card V_{PP}. The PCI4410 ignores this field unless V_{CC} to the socket is enabled (that is, 5 V or 3.3 V). This field is encoded as: 00 = No connection (default) 01 = V_{CC} 10 = 12 V 11 = Reserved

Table 5−5. ExCA Power Control Register 82365SL-DF Support

BIT	SIGNAL	TYPE	FUNCTION
7	COE	R/W	Card output enable. Bit 7 controls the state of all of the 16-bit outputs on the PCI4410. This bit is encoded as: 0 = 16-bit PC Card outputs disabled (default) 1 = 16-bit PC Card outputs enabled
6−5	RSVD	R	Reserved. Bits 6 and 5 return 0s when read.
4−3	EXCAVCC	R/W	V_{CC}. Bits 4 and 3 are used to request changes to card V_{CC}. This field is encoded as: 00 = 0 V (default) 01 = 0 V reserved 10 = 5 V 11 = 3 V
2	RSVD	R	Reserved. Bit 2 returns 0 when read.
1−0	EXCAVPP	R/W	V_{PP}. Bits 1 and 0 are used to request changes to card V_{PP}. The PCI4410 ignores this field unless V_{CC} to the socket is enabled. This field is encoded as: 00 = No connection (default) 01 = V_{CC} 10 = 12 V 11 = Reserved

5.4 ExCA Interrupt and General Control Register

The ExCA interrupt and general control register controls interrupt routing for I/O interrupts, as well as other critical 16-bit PC Card functions. See Table 5−6 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA interrupt and general control							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA interrupt and general control**
Type: Read/Write
Offset: CardBus socket address + 803h; ExCA offset 03h
Default: 00h

Table 5−6. ExCA Interrupt and General Control Register

BIT	SIGNAL	TYPE	FUNCTION
7	RINGEN	R/W	Card ring indicate enable. Bit 7 enables the ring indicate function of BVD1/$\overline{\text{RI}}$. This bit is encoded as: 0 = Ring indicate disabled (default) 1 = Ring indicate enabled
6	RESET	R/W	Card reset. Bit 6 controls the 16-bit PC Card RESET, and allows host software to force a card reset. Bit 6 affects 16-bit cards only. This bit is encoded as: 0 = RESET signal asserted (default) 1 = RESET signal deasserted
5	CARDTYPE	R/W	Card type. Bit 5 indicates the PC card type. This bit is encoded as: 0 = Memory PC Card installed (default) 1 = I/O PC Card installed
4	CSCROUTE	R/W	PCI Interrupt CSC routing enable bit. When bit 4 is set (high), the card status change interrupts are routed to PCI interrupts. When low, the card status-change interrupts are routed using bits 7−4 (CSCSELECT field) in the ExCA card status-change-interrupt configuration register (see Section 5.6). This bit is encoded as: 0 = CSC interrupts are routed by ExCA registers (default). 1 = CSC interrupts are routed to PCI interrupts.
3−0	INTSELECT	R/W	Card interrupt select for I/O PC Card functional interrupts. Bits 3−0 select the interrupt routing for I/O PC Card functional interrupts. This field is encoded as: 0000 = No interrupt routing (default) . CSC interrupts routed to PCI interrupts. These bit settings, along with bit 4 (CSCROUTE) are combined through an OR function for backwards compatibility. 0001 = IRQ1 enabled 0010 = SMI enabled 0011 = IRQ3 enabled 0100 = IRQ4 enabled 0101 = IRQ5 enabled 0100 = IRQ6 enabled 0111 = IRQ7 enabled 1000 = IRQ8 enabled 1001 = IRQ9 enabled 1010 = IRQ10 enabled 1011 = IRQ11 enabled 1100 = IRQ12 enabled 1101 = IRQ13 enabled 1110 = IRQ14 enabled 1111 = IRQ15 enabled

5.5 ExCA Card Status-Change Register

The ExCA card status-change register controls interrupt routing for I/O interrupts as well as other critical 16-bit PC Card functions. The register enables these interrupt sources to generate an interrupt to the host. When the interrupt source is disabled, the corresponding bit in this register always reads 0. When an interrupt source is enabled, the corresponding bit in this register is set to indicate that the interrupt source is active. After generating the interrupt to the host, the interrupt service routine must read this register to determine the source of the interrupt. The interrupt service routine is responsible for resetting the bits in this register as well. Resetting a bit is accomplished by one of two methods: a read of this register or an explicit writeback of 1 to the status bit. The choice of these two methods is based on bit 2 (interrupt flag clear mode select) in the ExCA global control register (see Section 5.22). See Table 5−7 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA card status-change							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **ExCA card status-change**
Type: Read-only
Offset: CardBus socket address + 804h; ExCA offset 04h
Default: 00h

Table 5−7. ExCA Card Status-Change Register

BIT	SIGNAL	TYPE	FUNCTION
7−4	RSVD	R	Reserved. Bits 7−4 return 0s when read.
3	CDCHANGE	R	Card detect change. Bit 3 indicates whether a change on $\overline{CD1}$ or $\overline{CD2}$ occurred at the PC Card interface. This bit is encoded as: 0 = No change detected on either $\overline{CD1}$ or $\overline{CD2}$ 1 = Change detected on either $\overline{CD1}$ or $\overline{CD2}$
2	READYCHANGE	R	Ready change. When a 16-bit memory is installed in the socket, bit 2 includes whether the source of a PCI4410 interrupt was due to a change on READY at the PC Card interface, indicating that the PC Card is now ready to accept new data. This bit is encoded as: 0 = No low-to-high transition detected on READY (default) 1 = Detected low-to-high transition on READY When a 16-bit I/O card is installed, bit 2 is always 0.
1	BATWARN	R	Battery warning change. When a 16-bit memory card is installed in the socket, bit 1 indicates whether the source of a PCI4410 interrupt was due to a battery-low warning condition. This bit is encoded as: 0 = No battery warning condition (default) 1 = Detected battery warning condition When a 16-bit I/O card is installed, bit 1 is always 0.
0	BATDEAD//$\overline{RI}$	R	Battery dead or status change. When a 16-bit memory card is installed in the socket, bit 0 indicates whether the source of a PCI4410 interrupt was due to a battery dead condition. This bit is encoded as: 0 = $\overline{STSCHG}$ deasserted (default) 1 = $\overline{STSCHG}$ asserted Ring indicate. When an I/O card is installed in the socket and the PCI4410 is configured for ring-indicate operation, bit 0 indicates the status of $\overline{RI}$.

5.6 ExCA Card Status-Change-Interrupt Configuration Register

The ExCA card status-change-interrupt configuration register controls interrupt routing for card status-change interrupts, as well as masking CSC interrupt sources. See Table 5−8 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA status-change-interrupt configuration							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA card status-change-interrupt configuration**
Type: Read/Write
Offset: CardBus socket address + 805h; ExCA offset 05h
Default: 00h

Table 5−8. ExCA Card Status-Change-Interrupt Configuration Register

BIT	SIGNAL	TYPE	FUNCTION
7−4	CSCSELECT	R/W	Interrupt select for card status change. Bits 7−4 select the interrupt routing for card status change interrupts. 0000 = CSC interrupts routed to PCI interrupts if bit 5 (CSC) of the diagnostic register is set to 1 (see Section 4.36). In this case bit 4 (CSCROUTE) of the ExCA interrupt and general control register is a “don’t care” (see Section 5.4). This is the default setting. 0000 = No ISA interrupt routing if bit 5 (CSC) of the diagnostic register is set to 0 (see Section 4.36). In this case, CSC interrupts are routed to PCI interrupts by setting bit 4 (CSCROUTE) of the ExCA interrupt and general control register to 1 (see Section 5.4). This field is encoded as: 0000 = No interrupt routing (default) 1000 = IRQ8 enabled 0001 = IRQ1 enabled 1001 = IRQ9 enabled 0010 = SMI enabled 1010 = IRQ10 enabled 0011 = IRQ3 enabled 1011 = IRQ11 enabled 0100 = IRQ4 enabled 1100 = IRQ12 enabled 0101 = IRQ5 enabled 1101 = IRQ13 enabled 0110 = IRQ6 enabled 1110 = IRQ14 enabled 0111 = IRQ7 enabled 1111 = IRQ15 enabled
3	CDEN	R/W	Card detect enable. Bit 3 enables interrupts on $\overline{CD1}$ or $\overline{CD2}$ changes. This bit is encoded as: 0 = Disables interrupts on $\overline{CD1}$ or $\overline{CD2}$ line changes (default) 1 = Enables interrupts on $\overline{CD1}$ or $\overline{CD2}$ line changes
2	READYEN	R/W	Ready enable. Bit 2 enables/disables a low-to-high transition on PC Card READY to generate a host interrupt. This interrupt source is considered a card status change. This bit is encoded as: 0 = Disables host interrupt generation (default) 1 = Enables host interrupt generation
1	BATWARNEN	R/W	Battery warning enable. Bit 1 enables/disables a battery warning condition to generate a CSC interrupt. This bit is encoded as: 0 = Disables host interrupt generation (default) 1 = Enables host interrupt generation
0	BATDEADEN	R/W	Battery dead enable. Bit 0 enables/disables the generation of a CSC interrupt for a battery dead contition (16-bit memory PC card) or assertion of the $\overline{STSCHG}$ signal (16-bit I/O PC card). 0 = Disables host interrupt generation (default) 1 = Enables host interrupt generation

5.7 ExCA Address Window Enable Register

The ExCA address window enable register enables/disables the memory and I/O windows to the 16-bit PC Card. By default, all windows to the card are disabled. The PCI4410 does not acknowledge PCI memory or I/O cycles to the card if the corresponding enable bit in this register is 0, regardless of the programming of the memory or I/O window start/end/offset address registers. See Table 5−9 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA address window enable							
Type	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA address window enable**
Type: Read-only, Read/Write
Offset: CardBus socket address + 806h; ExCA offset 06h
Default: 00h

Table 5−9. ExCA Address Window Enable Register

BIT	SIGNAL	TYPE	FUNCTION
7	IOWIN1EN	R/W	I/O window 1 enable. Bit 7 enables/disables I/O window 1 for the PC Card. This bit is encoded as: 0 = I/O window 1 disabled (default) 1 = I/O window 1 enabled
6	IOWIN0EN	R/W	I/O window 0 enable. Bit 6 enables/disables I/O window 0 for the PC Card. This bit is encoded as: 0 = I/O window 0 disabled (default) 1 = I/O window 0 enabled
5	RSVD	R	Reserved. Bit 5 returns 0 when read.
4	MEMWIN4EN	R/W	Memory window 4 enable. Bit 4 enables/disables memory window 4 for the PC Card. This bit is encoded as: 0 = Memory window 4 disabled (default) 1 = Memory window 4 enabled
3	MEMWIN3EN	R/W	Memory window 3 enable. Bit 3 enables/disables memory window 3 for the PC Card. This bit is encoded as: 0 = Memory window 3 disabled (default) 1 = Memory window 3 enabled
2	MEMWIN2EN	R/W	Memory window 2 enable. Bit 2 enables/disables memory window 2 for the PC Card. This bit is encoded as: 0 = Memory window 2 disabled (default) 1 = Memory window 2 enabled
1	MEMWIN1EN	R/W	Memory window 1 enable. Bit 1 enables/disables memory window 1 for the PC Card. This bit is encoded as: 0 = Memory window 1 disabled (default) 1 = Memory window 1 enabled
0	MEMWIN0EN	R/W	Memory window 0 enable. Bit 0 enables/disables memory window 0 for the PC Card. This bit is encoded as: 0 = Memory window 0 disabled (default) 1 = Memory window 0 enabled

5.8 ExCA I/O Window Control Register

The ExCA I/O window control register contains parameters related to I/O window sizing and cycle timing. See Table 5−10 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O window control							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA I/O window control**
Type: Read/Write
Offset: CardBus socket address + 807h; ExCA offset 07h
Default: 00h

Table 5−10. ExCA I/O Window Control Register

BIT	SIGNAL	TYPE	FUNCTION
7	WAITSTATE1	R/W	I/O window 1 wait state. Bit 7 controls the I/O window 1 wait state for 16-bit I/O accesses. Bit 7 has no effect on 8-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 16-bit cycles have standard length (default). 1 = 16-bit cycles are extended by one equivalent ISA wait state.
6	ZEROWS1	R/W	I/O window 1 zero wait state. Bit 6 controls the I/O window 1 wait state for 8-bit I/O accesses. Bit 6 has no effect on 16-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 8-bit cycles have standard length (default). 1 = 8-bit cycles are reduced to equivalent of three ISA cycles.
5	IOSIS16W1	R/W	I/O window 1 $\overline{\text{IOIS16}}$ source. Bit 5 controls the I/O window 1 automatic data sizing feature that uses $\overline{\text{IOIS16}}$ from the PC Card to determine the data width of the I/O data transfer. This bit is encoded as: 0 = Window data width determined by $\overline{\text{DATASIZE1}}$, bit 4 (default). 1 = Window data width determined by $\overline{\text{IOIS16}}$.
4	DATASIZE1	R/W	I/O window 1 data size. Bit 4 controls the I/O window 1 data size. Bit 4 is ignored if bit 5 (IOSIS16W1) is set. This bit is encoded as: 0 = Window data width is 8 bits (default). 1 = Window data width is 16 bits.
3	WAITSTATE0	R/W	I/O window 0 wait state. Bit 3 controls the I/O window 0 wait state for 16-bit I/O accesses. Bit 3 has no effect on 8-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 16-bit cycles have standard length (default). 1 = 16-bit cycles are extended by one equivalent ISA wait state.
2	ZEROWS0	R/W	I/O window 0 zero wait state. Bit 2 controls the I/O window 0 wait state for 8-bit I/O accesses. Bit 2 has no effect on 16-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 8-bit cycles have standard length (default). 1 = 8-bit cycles are reduced to equivalent of three ISA cycles.
1	IOSIS16W0	R/W	I/O window 0 $\overline{\text{IOIS16}}$ source. Bit 1 controls the I/O window 0 automatic data sizing feature that uses $\overline{\text{IOIS16}}$ from the PC Card to determine the data width of the I/O data transfer. This bit is encoded as: 0 = Window data width is determined by $\overline{\text{DATASIZE0}}$, bit 0 (default). 1 = Window data width is determined by $\overline{\text{IOIS16}}$.
0	DATASIZE0	R/W	I/O window 0 data size. Bit 0 controls the I/O window 0 data size. Bit 0 is ignored if bit 1 (IOSIS16W0) is set. This bit is encoded as: 0 = Window data width is 8 bits (default). 1 = Window data width is 16 bits.

5.9 ExCA I/O Windows 0 and 1 Start-Address Low-Byte Registers

These registers contain the low byte of the 16-bit I/O window start address for I/O windows 0 and 1. The 8 bits of these registers correspond to the lower 8 bits of the start address.

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O windows 0 and 1 start-address low byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA I/O window 0 start-address low byte**
Offset: CardBus socket address + 808h; ExCA offset 08h
Register: **ExCA I/O window 1 start-address low byte**
Offset: CardBus socket address + 80Ch; ExCA offset 0Ch
Type: Read/Write
Default: 00h
Size: One byte

5.10 ExCA I/O Windows 0 and 1 Start-Address High-Byte Registers

These registers contain the high byte of the 16-bit I/O window start address for I/O windows 0 and 1. The 8 bits of these registers correspond to the upper 8 bits of the start address.

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O windows 0 and 1 start-address high byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA I/O window 0 start-address high byte**
Offset: CardBus socket address + 809h; ExCA offset 09h
Register: **ExCA I/O window 1 start-address high byte**
Offset: CardBus socket address + 80Dh; ExCA offset 0Dh
Type: Read/write
Default: 00h
Size: One byte

5.11 ExCA I/O Windows 0 and 1 End-Address Low-Byte Registers

These registers contain the low byte of the 16-bit I/O window end address for I/O windows 0 and 1. The 8 bits of these registers correspond to the lower 8 bits of the end address.

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O windows 0 and 1 end-address low byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA I/O window 0 end-address low byte**
Offset: CardBus socket address + 80Ah; ExCA offset 0Ah
Register: **ExCA I/O window 1 end-address low byte**
Offset: CardBus socket address + 80Eh; ExCA offset 0Eh
Type: Read/Write
Default: 00h
Size: One byte

5.12 ExCA I/O Windows 0 and 1 End-Address High-Byte Registers

These registers contain the high byte of the 16-bit I/O window end address for I/O windows 0 and 1. The 8 bits of these registers correspond to the upper 8 bits of the end address.

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O windows 0 and 1 end-address high byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA I/O window 0 end-address high byte**
Offset: CardBus socket address + 80Bh; ExCA offset 0Bh
Register: **ExCA I/O window 1 end-address high byte**
Offset: CardBus socket address + 80Fh; ExCA offset 0Fh
Type: Read/write
Default: 00h
Size: One byte

5.13 ExCA Memory Windows 0−4 Start-Address Low-Byte Registers

These registers contain the low byte of the 16-bit memory window start address for memory windows 0, 1, 2, 3, and 4. The 8 bits of these registers correspond to bits A19−A12 of the start address.

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory windows 0−4 start-address low byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA memory window 0 start-address low byte**
Offset: CardBus socket address + 810h; ExCA offset 10h
Register: **ExCA memory window 1 start-address low byte**
Offset: CardBus socket address + 818h; ExCA offset 18h
Register: **ExCA memory window 2 start-address low byte**
Offset: CardBus socket address + 820h; ExCA offset 20h
Register: **ExCA memory window 3 start-address low byte**
Offset: CardBus socket address + 828h; ExCA offset 28h
Register: **ExCA memory window 4 start-address low byte**
Offset: CardBus socket address + 830h; ExCA offset 30h
Type: Read/Write
Default: 00h
Size: One byte

5.14 ExCA Memory Windows 0−4 Start-Address High-Byte Registers

These registers contain the high nibble of the 16-bit memory window start address for memory windows 0, 1, 2, 3, and 4. The lower 4 bits of these registers correspond to bits A23−A20 of the start address. In addition, the memory window data width and wait states are set in this register. See Table 5−11 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory windows 0−4 start-address high byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA memory window 0 start-address high byte**
Offset: CardBus socket address + 811h; ExCA offset 11h
Register: **ExCA memory window 1 start-address high byte**
Offset: CardBus socket address + 819h; ExCA offset 19h
Register: **ExCA memory window 2 start-address high byte**
Offset: CardBus socket address + 821h; ExCA offset 21h
Register: **ExCA memory window 3 start-address high byte**
Offset: CardBus socket address + 829h; ExCA offset 29h
Register: **ExCA memory window 4 start-address high byte**
Offset: CardBus socket address + 831h; ExCA offset 31h
Type: Read/Write
Default: 00h
Size: One byte

Table 5−11. ExCA Memory Windows 0−4 Start-Address High-Byte Registers

BIT	SIGNAL	TYPE	FUNCTION
7	DATASIZE	R/W	Data size. Bit 7 controls the memory window data width. This bit is encoded as: 0 = Window data width is 8 bits (default). 1 = Window data width is 16 bits.
6	ZEROWAIT	R/W	Zero wait state. Bit 6 controls the memory window wait state for 8- and 16-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 8- and 16-bit cycles have standard length (default). 1 = 8-bit cycles are reduced to equivalent of three ISA cycles. 16-bit cycles are reduced to equivalent of two ISA cycles.
5−4	SCRATCH	R/W	Scratch pad bits. Bits 5 and 4 have no effect on memory window operation.
3−0	STAHN	R/W	Start-address high nibble. Bits 3−0 represent the upper address bits A23−A20 of the memory window start address.

5.15 ExCA Memory Windows 0−4 End-Address Low-Byte Registers

These registers contain the low byte of the 16-bit memory window end address for memory windows 0, 1, 2, 3, and 4. The 8 bits of these registers correspond to bits A19−A12 of the end address.

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory windows 0−4 end-address low byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA memory window 0 end-address low byte**
Offset: CardBus socket address + 812h; ExCA offset 12h
Register: **ExCA memory window 1 end-address low byte**
Offset: CardBus socket address + 81Ah; ExCA offset 1Ah
Register: **ExCA memory window 2 end-address low byte**
Offset: CardBus socket address + 822h; ExCA offset 22h
Register: **ExCA memory window 3 end-address low byte**
Offset: CardBus socket address + 82Ah; ExCA offset 2Ah
Register: **ExCA memory window 4 end-address low byte**
Offset: CardBus socket address + 832h; ExCA offset 32h
Type: Read/Write
Default: 00h
Size: One byte

5.16 ExCA Memory Windows 0−4 End-Address High-Byte Registers

These registers contain the high nibble of the 16-bit memory window end address for memory windows 0, 1, 2, 3, and 4. The lower 4 bits of these registers correspond to bits A23−A20 of the end address. In addition, the memory window wait states are set in this register. See Table 5−12 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory windows 0−4 end-address high byte							
Type	R/W	R/W	R	R	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA memory window 0 end-address high byte**
Offset: CardBus socket address + 813h; ExCA offset 13h
Register: **ExCA memory window 1 end-address high byte**
Offset: CardBus socket address + 81Bh; ExCA offset 1Bh
Register: **ExCA memory window 2 end-address high byte**
Offset: CardBus socket address + 823h; ExCA offset 23h
Register: **ExCA memory window 3 end-address high byte**
Offset: CardBus socket address + 82Bh; ExCA offset 2Bh
Register: **ExCA memory window 4 end-address high byte**
Offset: CardBus socket address + 833h; ExCA offset 33h
Type: Read-only, Read/Write
Default: 00h
Size: One byte

Table 5−12. ExCA Memory Windows 0−4 End-Address High-Byte Registers

BIT	SIGNAL	TYPE	FUNCTION
7−6	MEMWS	R/W	Wait state. Bits 7 and 6 specify the number of equivalent ISA wait states to be added to 16-bit memory accesses. The number of wait states added is equal to the binary value of these two bits.
5−4	RSVD	R	Reserved. Bits 5 and 4 return 0s when read.
3−0	ENDHN	R/W	End-address high nibble. Bits 3−0 represent the upper address bits A23−A20 of the memory window end address.

5.17 ExCA Memory Windows 0−4 Offset-Address Low-Byte Registers

These registers contain the low byte of the 16-bit memory window offset address for memory windows 0, 1, 2, 3, and 4. The 8 bits of these registers correspond to bits A19−A12 of the offset address.

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory windows 0−4 offset-address low byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA memory window 0 offset-address low byte**
Offset: CardBus socket address + 814h; ExCA offset 14h
Register: **ExCA memory window 1 offset-address low byte**
Offset: CardBus socket address + 81Ch; ExCA offset 1Ch
Register: **ExCA memory window 2 offset-address low byte**
Offset: CardBus socket address + 824h; ExCA offset 24h
Register: **ExCA memory window 3 offset-address low byte**
Offset: CardBus socket address + 82Ch; ExCA offset 2Ch
Register: **ExCA memory window 4 offset-address low byte**
Offset: CardBus socket address + 834h; ExCA offset 34h
Type: Read/Write
Default: 00h
Size: One byte

5.18 ExCA Memory Windows 0−4 Offset-Address High-Byte Registers

These registers contain the high 6 bits of the 16-bit memory window offset address for memory windows 0, 1, 2, 3, and 4. The lower 6 bits of these registers correspond to bits A25−A20 of the offset address. In addition, the write protection and common/attribute memory configurations are set in this register. See Table 5−13 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory windows 0−4 offset-address high byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA memory window 0 offset-address high byte**
Offset: CardBus socket address + 815h; ExCA offset 15h
Register: **ExCA memory window 1 offset-address high byte**
Offset: CardBus socket address + 81Dh; ExCA offset 1Dh
Register: **ExCA memory window 2 offset-address high byte**
Offset: CardBus socket address + 825h; ExCA offset 25h
Register: **ExCA memory window 3 offset-address high byte**
Offset: CardBus socket address + 82Dh; ExCA offset 2Dh
Register: **ExCA memory window 4 offset-address high byte**
Offset: CardBus socket address + 835h; ExCA offset 35h
Type: Read/Write
Default: 00h
Size: One byte

Table 5−13. ExCA Memory Windows 0−4 Offset-Address High-Byte Registers

BIT	SIGNAL	TYPE	FUNCTION
7	WINWP	R/W	Write protect. Bit 7 specifies whether write operations to this memory window are enabled. This bit is encoded as: 0 = Write operations are allowed (default). 1 = Write operations are not allowed.
6	REG	R/W	Bit 6 specifies whether this memory window is mapped to card attribute or common memory. This bit is encoded as: 0 = Memory window is mapped to common memory (default). 1 = Memory window is mapped to card attribute memory.
5−0	OFFHB	R/W	Offset-address high byte. Bits 5−0 represent the upper address bits A25−A20 of the memory window offset address.

5.19 ExCA I/O Windows 0 and 1 Offset-Address Low-Byte Registers

These registers contain the low byte of the 16-bit I/O window offset address for I/O windows 0 and 1. The 8 bits of these registers correspond to the lower 8 bits of the offset address, and bit 0 is always 0.

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O windows 0 and 1 offset-address low byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Default	0	0	0	0	0	0	0	0

Register: **ExCA I/O window 0 offset-address low byte**
Offset: CardBus socket address + 836h; ExCA offset 36h
Register: **ExCA I/O window 1 offset-address low byte**
Offset: CardBus socket address + 838h; ExCA offset 38h
Type: Read-only, Read/Write
Default: 00h
Size: One byte

5.20 ExCA I/O Windows 0 and 1 Offset-Address High-Byte Registers

These registers contain the high byte of the 16-bit I/O window offset address for I/O windows 0 and 1. The 8 bits of these registers correspond to the upper 8 bits of the offset address.

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O windows 0 and 1 offset-address high byte							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA I/O window 0 offset-address high byte**
Offset: CardBus socket address + 837h; ExCA offset 37h
Register: **ExCA I/O window 1 offset-address high byte**
Offset: CardBus socket address + 839h; ExCA offset 39h
Type: Read/Write
Default: 00h
Size: One byte

5.21 ExCA I/O Card Detect and General Control Register

The ExCA card detect and general control register controls how the ExCA registers for the socket respond to card removal, as well as reports the status of $\overline{VS1}$ and $\overline{VS2}$ at the PC Card interface. See Table 5−14 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O card detect and general control							
Type	R	R	R/W	R/W	R	R	R/W	R
Default	X	X	0	0	0	0	0	0

Register: **ExCA card detect and general control**
Type: Read-only, Read/Write
Offset: CardBus socket address + 816h; ExCA offset 16h
Default: XX00 0000b

Table 5−14. ExCA I/O Card Detect and General Control Register

BIT	SIGNAL	TYPE	FUNCTION
7	VS2STAT	R	$\overline{VS2}$ state. Bit 7 reports the current state of $\overline{VS2}$ at the PC Card interface and, therefore, does not have a default value. 0 = $\overline{VS2}$ low 1 = $\overline{VS2}$ high
6	VS1STAT	R	$\overline{VS1}$ state. Bit 6 reports the current state of $\overline{VS1}$ at the PC Card interface and, therefore, does not have a default value. 0 = $\overline{VS1}$ low 1 = $\overline{VS1}$ high
5	SWCSC	R/W	Software card detect interrupt. If bit 3 (CDEN) in the ExCA card status-change-interrupt configuration register is set (see Section 5.6), then writing a 1 to bit 5 causes a card-detect card-status change interrupt for the associated card socket. If bit 3 (CDEN) in the ExCA card status-change-interrupt configuration register is cleared to 0 (see Section 5.6), then writing a 1 to bit 5 has no effect. A read operation of this bit always returns 0.
4	CDRESUME	R/W	Card detect resume enable. If bit 4 is set to 1, then once a card detect change has been detected on $\overline{CD1}$ and $\overline{CD2}$ inputs, $\overline{RI\_OUT}$ goes from high to low. $\overline{RI\_OUT}$ remains low until bit 0 (card status change) in the ExCA card status-change register is cleared (see Section 5.5). If this bit is a 0, then the card detect resume functionality is disabled. 0 = Card detect resume disabled (default) 1 = Card detect resume enabled
3−2	RSVD	R	Reserved. Bits 3 and 2 return 0s when read.
1	REGCONFIG	R/W	Register configuration on card removal. Bit 1 controls how the ExCA registers for the socket react to a card removal event. This bit is encoded as: 0 = No change to ExCA registers on card removal (default) 1 = Reset ExCA registers on card removal
0	RSVD	R	Reserved. Bit 0 returns 0 when read.

5.22 ExCA Global Control Register

The ExCA global control register controls the PC Card socket. The host interrupt mode bits in this register are retained for Intel 82365SL-DF compatibility. See Table 5−15 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA global control							
Type	R	R	R	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA global control**
Type: Read-only, Read/Write
Offset: CardBus socket address + 81Eh; ExCA offset 1Eh
Default: 00h

Table 5−15. ExCA Global Control Register

BIT	SIGNAL	TYPE	FUNCTION
7−5	RSVD	R	Reserved. Bits 7−5 return 0s when read.
4	No function	R/W	This bit has no assigned function.
3	INTMODE	R/W	Level/edge interrupt mode select. Bit 3 selects the signaling mode for the PCI4410 host interrupt. This bit is encoded as: 0 = Host interrupt is edge mode (default). 1 = Host interrupt is level mode.
2	IFCMODE	R/W	Interrupt flag clear mode select. Bit 2 selects the interrupt flag clear mechanism for the flags in the ExCA card status-change register (see Section 5.5). This bit is encoded as: 0 = Interrupt flags are cleared by read of CSC register (default). 1 = Interrupt flags are cleared by explicit writeback of 1.
1	CSCMODE	R/W	Card status change level/edge mode select. Bit 1 selects the signaling mode for the PCI4410 host interrupt for card status changes. This bit is encoded as: 0 = Host interrupt is edge mode (default). 1 = Host interrupt is level mode.
0	PWRDWN	R/W	Power-down mode select. When bit 0 is set to 1, the PCI4410 is in power-down mode. In power-down mode, the PCI4410 card outputs are high impedance until an active cycle is executed on the card interface. Following an active cycle, the outputs are again high impedance. The PCI4410 still receives DMA requests, functional interrupts, and/or card status change interrupts; however, an actual card access is required to wake up the interface. This bit is encoded as: 0 = Power-down mode is disabled (default). 1 = Power-down mode is enabled.

5.23 ExCA Memory Windows 0−4 Page Register

The upper 8 bits of a 4-byte PCI memory address are compared to the contents of this register when addresses for 16-bit memory windows are decoded. Each window has its own page register, all of which default to 00h. By programming this register to a nonzero value, host software can locate 16-bit memory windows in any 1 of 256 16-Mbyte regions in the 4-Gbyte PCI address space. These registers are only accessible when the ExCA registers are memory mapped; that is, these registers cannot be accessed using the index/data I/O scheme.

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory windows 0−4 page							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA memory windows 0−4 page**
Type: Read/Write
Offset: CardBus socket address + 840h, 841h, 842h, 843h, 844h
Default: 00h

6 CardBus Socket Registers

The *1997 PC Card Standard* requires a CardBus socket controller to provide five 32-bit registers that report and control socket-specific functions. The PCI4410 provides the CardBus socket/ExCA base-address register (see Section 4.12) to locate these CardBus socket registers in PCI memory address space. Each socket has a separate base address register for accessing the CardBus socket registers (see Figure 6−1). Table 6−1 gives the location of the socket registers in relation to the CardBus socket/ExCA base address.

The PCI4410 implements an additional register at offset 20h that provides power management control for the socket.

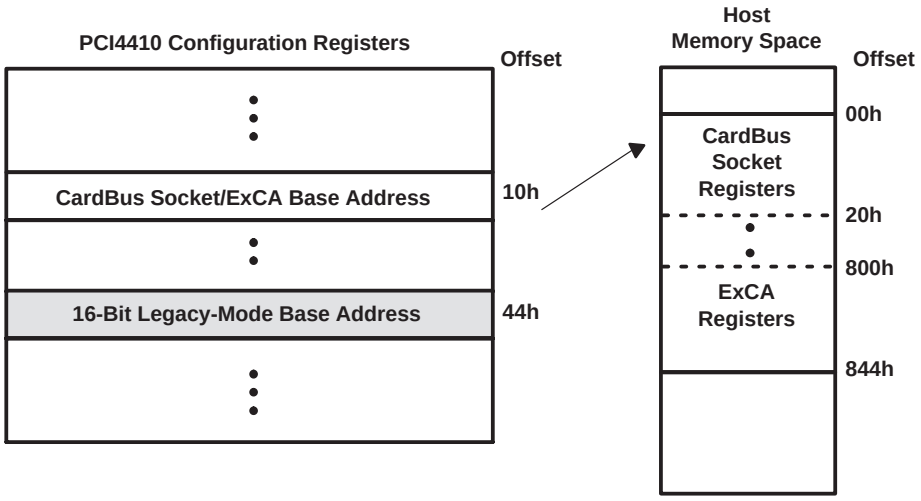


Figure 6−1. Accessing CardBus Socket Registers Through PCI Memory

Table 6−1. CardBus Socket Registers

REGISTER NAME	OFFSET
Socket event	00h
Socket mask	04h
Socket present state	08h
Socket force event	0Ch
Socket control	10h
Reserved	14h
Reserved	18h
Reserved	1Ch
Socket power management	20h

6.1 Socket Event Register

The socket event register indicates a change in socket status has occurred. These bits do not indicate what the change is, only that one has occurred. Software must read the socket present state register (see Section 6.3) for current status. Each bit in this register can be cleared by writing a 1 to that bit. The bits in this register can be set to a 1 by software by writing a 1 to the corresponding bit in the socket force event register (see Section 6.4). All bits in this register are cleared by PCI reset. They can be immediately set again, if, when coming out of PC Card reset, the bridge finds the status unchanged (that is, CSTSCHG reasserted or card detect is still true). Software must clear this register before enabling interrupts. If it is not cleared when interrupts are enabled, then an interrupt is generated (but not masked) based on any bit set. See Table 6−2 for a complete description of the register contents.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	Socket event															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Socket event															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R/C	R/C	R/C	R/C
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Socket event**
Type: Read-only, Read/Write to Clear
Offset: CardBus socket address + 00h
Default: 0000 0000h

Table 6−2. Socket Event Register

BIT	SIGNAL	TYPE	FUNCTION
31−4	RSVD	R	Reserved. Bits 31−4 return 0s when read.
3	PWREVENT	R/C	Power cycle. Bit 3 is set when the PCI4410 detects that bit 3 (PWRCYCLE) in the socket present state register (see Section 6.3) has changed state. This bit is cleared by writing a 1.
2	CD2EVENT	R/C	$\overline{\text{CCD2}}$. Bit 2 is set when the PCI4410 detects that bit 2 (CDETECT2) in the socket present state register (see Section 6.3) has changed state. This bit is cleared by writing a 1.
1	CD1EVENT	R/C	$\overline{\text{CCD1}}$. Bit 1 is set when the PCI4410 detects that bit 1 (CDETECT1) in the socket present state register (see Section 6.3) has changed state. This bit is cleared by writing a 1.
0	CSTSEVENT	R/C	CSTSCHG. Bit 0 is set when bit 0 (CARDSTS) in the socket present state register (see Section 6.3) has changed state. For CardBus cards, bit 0 is set on the rising edge of CSTSCHG. For 16-bit PC Cards, bit 0 is set on both transitions of CSTSCHG. This bit is reset by writing a 1.

6.2 Socket Mask Register

The socket mask register allows software to control the CardBus card events that generate a status change interrupt. The state of these mask bits does not prevent the corresponding bits from reacting in the socket event register (see Section 6.1). See Table 6−3 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Socket mask															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Socket mask															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Socket mask**
Type: Read-only, Read/Write
Offset: CardBus socket address + 04h
Default: 0000 0000h

Table 6−3. Socket Mask Register

BIT	SIGNAL	TYPE	FUNCTION
31−4	RSVD	R	Reserved. Bits 31−4 return 0s when read.
3	PWRMASK	R/W	Power cycle. Bit 3 masks bit 3 (PWRCYCLE) in the socket present state register (see Section 6.3) from causing a status change interrupt. 0 = PWRCYCLE event does not cause CSC interrupt (default). 1 = PWRCYCLE event causes CSC interrupt.
2−1	CDMASK	R/W	Card detect mask. Bits 2 and 1 mask bits 1 and 2 (CDETECT1 and CDETECT2) in the socket present state register (see Section 6.3) from causing a CSC interrupt. 00 = Insertion/removal does not cause CSC interrupt (default). 01 = Reserved (undefined) 10 = Reserved (undefined) 11 = Insertion/removal causes CSC interrupt.
0	CSTSMASK	R/W	CSTSCHG mask. Bit 0 masks bit 0 (CARDSTS) in the socket present state register (see Section 6.3) from causing a CSC interrupt. 0 = CARDSTS event does not cause CSC interrupt (default). 1 = CARDSTS event causes CSC interrupt.

6.3 Socket Present State Register

The socket present state register reports information about the socket interface. Write transactions to the socket force event register (see Section 6.4) are reflected here, as well as general socket interface status. Information about PC Card V_{CC} support and card type is only updated at each insertion. Also note that the PCI4410 uses $\overline{CCD1}$ and $\overline{CCD2}$ during card identification, and changes on these signals during this operation are not reflected in this register. See Table 6−4 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Socket present state															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Socket present state															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	X	0	0	0	X	X	X

Register: **Socket present state**
Type: Read-only
Offset: CardBus socket address + 08h
Default: 3000 00XXh

Table 6−4. Socket Present State Register

BIT	SIGNAL	TYPE	FUNCTION
31	YVSOCKET	R	YV socket. Bit 31 indicates whether or not the socket can supply V_{CC} = Y.Y V to PC Cards. The PCI4410 does not support Y.Y-V V_{CC}; therefore, this bit is hardwired to 0.
30	XVSOCKET	R	XV socket. Bit 30 indicates whether or not the socket can supply V_{CC} = X.X V to PC Cards. The PCI4410 does not support X.X-V V_{CC}; therefore, this bit is hardwired to 0.
29	3VSOCKET	R	3-V socket. Bit 29 indicates whether or not the socket can supply V_{CC} = 3.3 V to PC Cards. The PCI4410 does support 3.3-V V_{CC}; therefore, this bit is always set unless overridden by the socket force event register (see Section 6.4).
28	5VSOCKET	R	5-V socket. Bit 28 indicates whether or not the socket can supply V_{CC} = 5 V to PC Cards. The PCI4410 does support 5-V V_{CC}; therefore, this bit is always set unless overridden by the socket force event register (see Section 6.4).
27−14	RSVD	R	Reserved. Bits 27−14 return 0s when read.
13	YVCARD	R	YV card. Bit 13 indicates whether or not the PC Card inserted in the socket supports V_{CC} = Y.Y V.
12	XVCARD	R	XV card. Bit 12 indicates whether or not the PC Card inserted in the socket supports V_{CC} = X.X V.
11	3VCARD	R	3-V card. Bit 11 indicates whether or not the PC Card inserted in the socket supports V_{CC} = 3.3 V.
10	5VCARD	R	5-V card. Bit 10 indicates whether or not the PC Card inserted in the socket supports V_{CC} = 5 V.
9	BADVCCREQ	R	Bad V_{CC} request. Bit 9 indicates that the host software has requested that the socket be powered at an invalid voltage. 0 = Normal operation (default) 1 = Invalid V_{CC} request by host software
8	DATALOST	R	Data lost. Bit 8 indicates that a PC Card removal event may have caused lost data because the cycle did not terminate properly or because write data still resides in the PCI4410. 0 = Normal operation (default) 1 = Potential data loss due to card removal
7	NOTACARD	R	Not a card. Bit 7 indicates that an unrecognizable PC Card has been inserted in the socket. This bit is not updated until a valid PC Card is inserted into the socket. 0 = Normal operation (default) 1 = Unrecognizable PC Card detected

Table 6−4. Socket Present State Register (Continued)

BIT	SIGNAL	TYPE	FUNCTION

6	IREQCINT	R	READY($\overline{\text{IREQ}}$)//$\overline{\text{CINT}}$. Bit 6 indicates the current status of READY($\overline{\text{IREQ}}$)//$\overline{\text{CINT}}$ at the PC Card interface.
			0 = READY($\overline{\text{IREQ}}$)//$\overline{\text{CINT}}$ low
			1 = READY($\overline{\text{IREQ}}$)//$\overline{\text{CINT}}$ high

5	CBCARD	R	CardBus card detected. Bit 5 indicates that a CardBus PC Card is inserted in the socket. This bit is not updated until another card interrogation sequence occurs (card insertion).
4	16BITCARD	R	16-bit card detected. Bit 4 indicates that a 16-bit PC Card is inserted in the socket. This bit is not updated until another card interrogation sequence occurs (card insertion).
3	PWRCYCLE	R	Power cycle. Bit 3 indicates that the status of each card powering request. This bit is encoded as: 0 = Socket powered down (default) 1 = Socket powered up
2	CDETECT2	R	$\overline{\text{CCD2}}$. Bit 2 reflects the current status of $\overline{\text{CCD2}}$ at the PC Card interface. Changes to this signal during card interrogation are not reflected here. 0 = $\overline{\text{CCD2}}$ low (PC Card may be present) 1 = $\overline{\text{CCD2}}$ high (PC Card not present)
1	CDETECT1	R	$\overline{\text{CCD1}}$. Bit 1 reflects the current status of $\overline{\text{CCD1}}$ at the PC Card interface. Changes to this signal during card interrogation are not reflected here. 0 = $\overline{\text{CCD1}}$ low (PC Card may be present) 1 = $\overline{\text{CCD1}}$ high (PC Card not present)
0	CARDSTS	R	CSTSCHG. Bit 0 reflects the current status of CSTSCHG at the PC Card interface. 0 = CSTSCHG low 1 = CSTSCHG high

6.4 Socket Force Event Register

The socket force event register is used to force changes to the socket event register (see Section 6.1) and the socket present state register (see Section 6.3). Bit 14 (CVSTEST) in this register must be written when forcing changes that require card interrogation. See Table 6−5 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Socket force event															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Socket force event															
Type	R	W	W	W	W	W	W	W	W	R	W	W	W	W	W	W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Socket force event**
Type: Read-only, Write-only
Offset: CardBus socket address + 0Ch
Default: 0000 0000h

Table 6−5. Socket Force Event Register

BIT	SIGNAL	TYPE	FUNCTION
31−15	RSVD	R	Reserved. Bits 31−15 return 0s when read.
14	CVSTEST	W	Card VS test. When bit 14 is set, the PCI4410 re-interrogates the PC Card, updates the socket present state register (see Section 6.3), and enables the socket control register (see Section 6.5).
13	FYVCARD	W	Force YV card. Write transactions to bit 13 cause bit 13 (YVCARD) in the socket present state register to be written (see Section 6.3). When set, this bit disables the socket control register (see Section 6.5).
12	FXVCARD	W	Force XV card. Write transactions to bit 12 cause bit 12 (XVCARD) in the socket present state register to be written (see Section 6.3). When set, this bit disables the socket control register (see Section 6.5).
11	F3VCARD	W	Force 3-V card. Write transactions to bit 11 cause bit 11 (3VCARD) in the socket present state register to be written (see Section 6.3). When set, this bit disables the socket control register (see Section 6.5).
10	F5VCARD	W	Force 5-V card. Write transactions to bit 10 cause bit 10 (5VCARD) in the socket present state register to be written (see Section 6.3). When set, this bit disables the socket control register (see Section 6.5).
9	FBADVCCREQ	W	Force bad V_{CC} request. Changes to bit 9 (BADVCCREQ) in the socket present state register (see Section 6.3) can be made by writing to bit 9.
8	FDATALOST	W	Force data lost. Write transactions to bit 8 cause bit 8 (DATALOST) in the socket present state register to be written (see Section 6.3).
7	FNOTACARD	W	Force not a card. Write transactions to bit 7 cause bit 7 (NOTACARD) in the socket present state register to be written (see Section 6.3).
6	RSVD	R	Reserved. Bit 6 returns 0 when read.
5	FCBCARD	W	Force CardBus card. Write transactions to bit 5 cause bit 5 (CBCARD) in the socket present state register to be written (see Section 6.3).
4	F16BITCARD	W	Force 16-bit card. Write transactions to bit 4 cause bit 4 (16BITCARD) in the socket present state register to be written (see Section 6.3).
3	FPWRCYCLE	W	Force power cycle. Write transactions to bit 3 cause bit 3 (PWREVENT) in the socket event register to be written (see Section 6.1), and bit 3 (PWRCYCLE) in the socket present state register is unaffected (see Section 6.3).
2	FCDETECT2	W	Force $\overline{\text{CCD2}}$. Write transactions to bit 2 cause bit 2 (CD2EVENT) in the socket event register to be written (see Section 6.1), and bit 2 (CDETECT2) in the socket present state register is unaffected (see Section 6.3).
1	FCDETECT1	W	Force $\overline{\text{CCD1}}$. Write transactions to bit 1 cause bit 1 (CD1EVENT) in the socket event register to be written (see Section 6.1), and bit 1 (CDETECT1) in the socket present state register is unaffected (see Section 6.3).
0	FCARDSTS	W	Force CSTSCHG. Write transactions to bit 0 cause bit 0 (CSTSEVENT) in the socket event register to be written (see Section 6.1), and bit 0 (CARDSTS) in the socket present state register is unaffected (see Section 6.3).

6.5 Socket Control Register

The socket control register provides control of the voltages applied to the socket and instructions for CB $\overline{\text{CLKRUN}}$ protocol. The PCI4410 ensures that the socket is powered up only at acceptable voltages when a CardBus card is inserted. See Table 6−6 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Socket control															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Socket control															
Type	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Socket control**
Type: Read-only, Read/Write
Offset: CardBus socket address + 10h
Default: 0000 0000h

Table 6−6. Socket Control Register

BIT	SIGNAL	TYPE	FUNCTION
31−8	RSVD	R	Reserved. Bits 31−8 return 0s when read.
7	STOPCLK	R/W	CB $\overline{\text{CLKRUN}}$ protocol instructions. 0 = CB $\overline{\text{CLKRUN}}$ protocol can only attempt to stop/slow the CB clock if the socket is idle and the PCI $\overline{\text{CLKRUN}}$ protocol is preparing to stop/slow the PCI bus clock. 1 = CB $\overline{\text{CLKRUN}}$ protocol can attempt to stop/slow the CB clock if the socket is idle.
6−4	VCCCTRL	R/W	V_{CC} control. Bits 6−4 request card V_{CC} changes. 000 = Request power off (default) 001 = Reserved 010 = Request V_{CC} = 5 V 011 = Request V_{CC} = 3.3 V 100 = Request V_{CC} = X.X V 101 = Request V_{CC} = Y.Y V 110 = Reserved 111 = Reserved
3	RSVD	R	Reserved. Bit 3 returns 0 when read.
2−0	VPPCTRL	R/W	V_{PP} control. Bits 2−0 request card V_{PP} changes. 000 = Request power off (default) 001 = Request V_{PP} = 12 V 010 = Request V_{PP} = 5 V 011 = Request V_{PP} = 3.3 V 100 = Request V_{PP} = X.X V 101 = Request V_{PP} = Y.Y V 110 = Reserved 111 = Reserved

6.6 Socket Power Management Register

This register provides power management control over the socket through a mechanism for slowing or stopping the clock on the card interface when the card is idle. See Table 6−7 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Socket power management															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Socket power management															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Socket power management**
Type: Read-only, Read/Write
Offset: CardBus socket address + 20h
Default: 0000 0000h

Table 6−7. Socket Power Management Register

BIT	SIGNAL	TYPE	FUNCTION
31−26	RSVD	R	Reserved. Bits 31−26 return 0s when read.
25	SKTACCES	R	Socket access status. This bit provides information on when a socket access has occurred. This bit is cleared by a read access. 0 = A PC Card access has not occurred (default). 1 = A PC Card access has occurred.
24	SKTMODE	R	Socket mode status. This bit provides clock mode information. 0 = Clock is operating normally. 1 = Clock frequency has changed.
23−17	RSVD	R	Reserved. Bits 23−17 return 0s when read.
16	CLKCTRLEN	R/W	CardBus clock control enable. When bit 16 is set, bit 0 (CLKCTRL) is enabled. 0 = Clock control is disabled (default). 1 = Clock control is enabled.
15−1	RSVD	R	Reserved. Bits 15−1 return 0s when read.
0	CLKCTRL	R/W	CardBus clock control. This bit determines whether the CB $\overline{\text{CLKRUN}}$ protocol stops or slows the CB clock during idle states. Bit 16 (CLKCTRLEN) enables this bit. 0 = Allows CB $\overline{\text{CLKRUN}}$ protocol to stop the CB clock (default). 1 = Allows CB $\overline{\text{CLKRUN}}$ protocol to slow the CB clock by a factor of 16.

7 Distributed DMA (DDMA) Registers

The DMA base address, programmable in PCI configuration space at offset 98h, points to a 16-byte region in PCI I/O space where the DDMA registers reside. The names and locations of these registers are summarized in Table 7−1. These PCI4410 register definitions are identical in function, but differ in location, to the 8237 DMA controller. The similarity between the register models retains some level of compatibility with legacy DMA and simplifies the translation required by the master DMA device when it forwards legacy DMA writes to DMA channels.

While the DMA register definitions are identical to those in the 8237 of the same name, some register bits defined in the 8237 do not apply to distributed DMA in a PCI environment. In such cases, the PCI4410 implements these obsolete register bits as read-only nonfunctional bits. The reserved registers shown in Table 7−1 are implemented as read-only and return 0s when read. Write transactions to reserved registers have no effect.

Table 7−1. Distributed DMA Registers

TYPE	REGISTER NAME				DDMA BASE ADDRESS OFFSET
R	Reserved	Page	Current address		00h
W			Base address		
R	Reserved	Reserved	Current count		04h
W			Base count		
R	N/A	Reserved	N/A	Status	08h
W	Mode		Request	Command	
R	Multichannel	Reserved	N/A	Reserved	0Ch
W	Mask		Master clear		

7.1 DDMA Current Address/Base-Address Register

The DDMA current address/base-address register sets the starting (base) memory address of a DDMA transfer. Read transactions from this register indicate the current memory address of a direct memory transfer.

For the 8-bit DDMA transfer mode, the current address register contents are presented on AD15−AD0 of the PCI bus during the address phase. Bits 7−0 of the DDMA page register (see Section 7.2) are presented on AD23−AD16 of the PCI bus during the address phase.

For the 16-bit DDMA transfer mode, the current address register contents are presented on AD16−AD1 of the PCI bus during the address phase, and AD0 is driven to logic 0. Bits 7−1 of the DDMA page register (see Section 7.2) are presented on AD23−AD17 of the PCI bus during the address phase, and bit 0 is ignored.

Bit	15	14	13	12	11	10	9	8
Name	DDMA current address/base-address							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Name	DDMA current address/base-address							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **DDMA current address/base-address**
Type: Read/Write
Offset: DDMA base address + 00h
Default: 0000h
Size: Two bytes

7.2 DDMA Page Register

The DDMA page register sets the upper byte of the address of a DDMA transfer. Details of the address represented by this register are explained in Section 7.1, *DDMA Current Address/Base Address Register*.

Bit	7	6	5	4	3	2	1	0
Name	DDMA page							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **DDMA page**
Type: Read/Write
Offset: DDMA base address + 02h
Default: 00h
Size: One byte

7.3 DDMA Current Count/Base Count Register

The DDMA current count/base count register sets the total transfer count, in bytes, of a direct memory transfer. Read transactions to this register indicate the current count of a direct memory transfer. In the 8-bit transfer mode, the count is decremented by 1 after each transfer, and the count is decremented by 2 after each transfer in the 16-bit transfer mode.

Bit	15	14	13	12	11	10	9	8
Name	DDMA current count/base count							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
Name	DDMA current count/base count							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **DDMA current count/base count**
Type: Read/Write
Offset: DDMA base address + 04h
Default: 0000h
Size: Two bytes

7.4 DDMA Command Register

The DDMA command register enables and disables the DDMA controller. Bit 2 (DMAEN) defaults to 0 enabling the DDMA controller. All other bits are reserved. See Table 7−2 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	DDMA command							
Type	R	R	R	R	R	R/W	R	R
Default	0	0	0	0	0	0	0	0

Register: **DDMA command**
Type: Read-only, Read/Write
Offset: DDMA base address + 08h
Default: 00h
Size: One byte

Table 7−2. DDMA Command Register

BIT	SIGNAL	TYPE	FUNCTION
7−3	RSVD	R	Reserved. Bits 7−3 return 0s when read.
2	DMAEN	R/W	DDMA controller enable. Bit 2 enables and disables the distributed DMA slave controller in the PCI4410 and defaults to the enabled state. 0 = DDMA controller enabled (default) 1 = DDMA controller disabled
1−0	RSVD	R	Reserved. Bits 1 and 0 return 0s when read.

7.5 DDMA Status Register

The DDMA status register indicates the terminal count and DMA request ($\overline{\text{DREQ}}$) status. See Table 7−3 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	DDMA status							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **DDMA status**
Type: Read-only
Offset: DDMA base address + 08h
Default: 00h
Size: One byte

Table 7−3. DDMA Status Register

BIT	SIGNAL	TYPE	FUNCTION
7−4	DREQSTAT	R	Channel request. In the 8237, bits 7−4 indicate the status of $\overline{\text{DREQ}}$ of each DMA channel. In the PCI4410, these bits indicate the $\overline{\text{DREQ}}$ status of the single socket being serviced by this register. All four bits are set to 1 when the PC Card asserts $\overline{\text{DREQ}}$ and are reset to 0 when $\overline{\text{DREQ}}$ is deasserted. The status of bit 0 (MASKBIT) in the DDMA multichannel/mask register (see Section 7.9) has no effect on these bits.
3−0	TC	R	Channel terminal count. The 8327 uses bits 3−0 to indicate the TC status of each of its four DMA channels. In the PCI4410, these bits report information about a single DMA channel; therefore, all four of these register bits indicate the TC status of the single socket being serviced by this register. All four bits are set to 1 when the TC is reached by the DMA channel. These bits are reset to 0 when read or when the DMA channel is reset.

7.6 DDMA Request Register

The DDMA request register requests a DDMA transfer through software. Any write to this register enables software requests, and this register is to be used in block mode only.

Bit	7	6	5	4	3	2	1	0
Name	DDMA request							
Type	W	W	W	W	W	W	W	W
Default	0	0	0	0	0	0	0	0

Register: **DDMA request**
Type: Write-only
Offset: DDMA base address + 09h
Default: 00h
Size: One byte

7.7 DDMA Mode Register

The DDMA mode register sets the DDMA transfer mode. See Table 7−4 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	DDMA mode							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Default	0	0	0	0	0	0	0	0

Register: **DDMA mode**
Type: Read-only, Read/Write
Offset: DDMA base address + 0Bh
Default: 00h
Size: One byte

Table 7−4. DDMA Mode Register

BIT	SIGNAL	TYPE	FUNCTION
7−6	DMAMODE	R/W	Mode select. The PCI4410 uses bits 7 and 6 to determine the transfer mode. 00 = Demand mode select (default) 01 = Single mode select 10 = Block mode select 11 = Reserved
5	INCDEC	R/W	Address increment/decrement. The PCI4410 uses bit 5 to select the memory address in the DDMA current address/base address register to increment or decrement after each data transfer. This is in accordance with the 8237 use of this register bit and is encoded as follows: 0 = Addresses increment (default). 1 = Addresses decrement.
4	AUTOINIT	R/W	Auto initialization 0 = Auto initialization disabled (default) 1 = Auto initialization enabled
3−2	XFERTYPE	R/W	Transfer type. Bits 3 and 2 select the type of direct memory transfer to be performed. A memory write transfer moves data from the PCI4410 PC Card interface to memory and a memory read transfer moves data from memory to the PCI4410 PC Card interface. The field is encoded as: 00 = No transfer selected (default) 01 = Write transfer 10 = Read transfer 11 = Reserved
1−0	RSVD	R	Reserved. Bits 1 and 0 return 0s when read.

7.8 DDMA Master Clear Register

The DDMA master clear register resets the DDMA controller and all DDMA registers.

Bit	7	6	5	4	3	2	1	0
Name	DDMA master clear							
Type	W	W	W	W	W	W	W	W
Default	0	0	0	0	0	0	0	0

Register: **DDMA master clear**
Type: Write-only
Offset: DDMA base address + 0Dh
Default: 00h
Size: One byte

7.9 DDMA Multichannel/Mask Register

The PCI4410 uses only the least significant bit of this register to mask the PC Card DMA channel. The PCI4410 sets the mask bit to 1 when the PC Card is removed. Host software is responsible for either resetting the socket DMA controller or enabling the mask bit. See Table 7−5 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	DDMA multichannel/mask							
Type	R	R	R	R	R	R	R	R/W
Default	0	0	0	0	0	0	0	0

Register: **DDMA multichannel/mask**
Type: Read-only, Read/Write
Offset: DDMA base address + 0Fh
Default: 00h
Size: One byte

Table 7−5. DDMA Multichannel/Mask Register

BIT	SIGNAL	TYPE	FUNCTION
7−1	RSVD	R	Reserved. Bits 7−1 return 0s when read.
0	MASKBIT	R/W	Mask select. Bit 0 masks incoming $\overline{\text{DREQ}}$ signals from the PC Card. When set to 1, the socket ignores DMA requests from the card. When cleared (or reset to 0), incoming $\overline{\text{DREQ}}$ assertions are serviced normally. 0 = DDMA service provided on card $\overline{\text{DREQ}}$ 1 = Socket $\overline{\text{DREQ}}$ signal ignored (default)

8 OHCI-Lynx Controller Programming Model

This section describes the internal registers used to program the link function, including both PCI configuration registers and open HCI registers. All registers are detailed in the same format. A brief description is provided for each register, followed by the register offset and a bit table describing the reset state for each register.

A bit description table is typically included that indicates bit signal names, a detailed field description, and field access tags. Table 8−1 describes the field access tags.

Table 8−1. Bit Field Access Tag Descriptions

ACCESS TAG	NAME	MEANING
R	Read	Field may be read by software.
W	Write	Field may be written by software to any value.
S	Set	Field may be set to 1 by a write of 1. Writes of 0 have no effect.
C	Clear	Field may be reset to 0 by a write of 1. Writes of 0 have no effect.

8.1 PCI Configuration Registers

The PCI4410 link function configuration header is compliant with the PCI specification as a standard header. Table 8−2 illustrates the PCI configuration header which includes both the predefined portion of the configuration space and the user-definable registers. The registers that are labeled reserved are read-only, returning 0 when read, and are not applicable to the link function or have been reserved by the PCI specification for future use.

Table 8−2. PCI Configuration Register Map

REGISTER NAME				OFFSET
Device ID		Vendor ID		00h
Status		Command		04h
Class code			Revision ID	08h
BIST	Header type	Latency timer	Cache line size	0Ch
Open HCI registers base address				10h
TI extension registers base address				14h
Reserved				18h
Reserved				1Ch
Reserved				20h
Reserved				24h
Reserved				28h
Subsystem ID		Subsystem vendor ID		2Ch
Reserved				30h
Reserved			Capabilites pointer	34h
Reserved				38h
Max latency	Min grant	Interrupt pin	Interrupt line	3Ch
PCI OHCI control				40h
Power management capabilities		Next item pointer	Capability ID	44h
PM data	PMCSR_BSE	Power management CSR		48h
Reserved				4C−ECh
PCI miscellaneous configuration				F0h

Table 8−2. PCI Configuration Register Map (Continued)

REGISTER NAME				OFFSET
Link_enhancements				F4h
Subsystem ID alias		Subsystem vendor ID alias		F8h
GPIO3	GPIO2	GPIO1	GPIO0	FCh

8.2 Vendor ID Register

This 16-bit read-only register contains a value allocated by the PCI SIG and identifies the manufacturer of the PCI device. The vendor ID assigned to Texas Instruments is 104Ch.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Vendor ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	1	0	0	0	0	0	1	0	0	1	1	0	0

Register: **Vender ID**
Type: Read-only
Offset: 00h
Default: 104Ch

8.3 Device ID Register

This 16-bit read-only register contains a value assigned to the PCI4410 by Texas Instruments. The device identification for the PCI4410 OHCI controller function is 8017h.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Device ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

Register: **Device ID register**
Type: Read-only
Offset: 02h
Default: 8017h

8.4 PCI Command Register

The command register provides control over the PCI4410 link interface to the PCI bus. All bit functions adhere to the definitions in the *PCI Local Bus Specification*, as seen in the following bit descriptions. See Table 8−3 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCI command															
Type	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R/W	R/W	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **PCI command**
Type: Read-only, Read/Write
Offset: 04h
Default: 0000h

Table 8−3. PCI Command Register

BIT	SIGNAL	TYPE	FUNCTION
15−10	RSVD	R	Reserved. These bits return 0s when read.
9	FBB_ENB	R	Fast back-to-back enable. The PCI4410 will not generate fast back-to-back transactions; thus, this bit returns 0 when read.
8	SERR_ENB	R/W	$\overline{\text{SERR}}$ enable. When this bit is set to 1, the PCI4410 $\overline{\text{SERR}}$ driver is enabled. $\overline{\text{SERR}}$ can be asserted after detecting an address parity error on the PCI bus.
7	STEP_ENB	R	Address/data stepping control. The PCI4410 does not support address/data stepping, and this bit is hardwired to 0.
6	PERR_ENB	R/W	Parity error enable. When this bit is set to 1, the PCI4410 is enabled to drive $\overline{\text{PERR}}$ response to parity errors through the $\overline{\text{PERR}}$ signal.
5	VGA_ENB	R	VGA palette snoop enable. The PCI4410 does not feature VGA palette snooping. This bit returns 0 when read.
4	MWI_ENB	R/W	Memory write and invalidate enable. When this bit is set to 1, the PCI4410 is enabled to generate MWI PCI bus commands. If reset to 0, the PCI4410 will generate memory write commands instead.
3	SPECIAL	R	Special cycle enable. The PCI4410 does not respond to special cycle transactions. This bit returns 0 when read.
2	MASTER_ENB	R/W	Bus master enable. When this bit is set to 1, the PCI4410 is enabled to initiate cycles on the PCI bus.
1	MEMORY_ENB	R/W	Memory response enable. Setting this bit to 1 enables the PCI4410 to respond to memory cycles on the PCI bus. This bit must be set to 1 to access OHCI registers.
0	IO_ENB	R	I/O space enable. The PCI4410 link does not implement any I/O mapped functionality; thus, this bit returns 0 when read.

8.5 PCI Status Register

The PCI status register provides device information to the host system. Bits in this register may be read normally. A bit in the status register is reset when a 1 is written to that bit location; a 0 written to a bit location has no effect. All bit functions adhere to the definitions in the *PCI Local Bus Specification*. PCI bus status is shown through each function. See Table 8−4 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCI status															
Type	RCU	RCU	RCU	RCU	RCU	R	R	RCU	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0

Register: **PCI status**
Type: Read-only, Read/Clear/Update
Offset: 06h
Default: 0210h

Table 8−4. PCI Status Register

BIT	SIGNAL	TYPE	FUNCTION
15	PAR_ERR	RCU	Detected parity error. This bit is set to 1 when a parity error is detected, either address or data parity errors.
14	SYS_ERR	RCU	Signaled system error. This bit is set to 1 when $\overline{\text{SERR}}$ is enabled and the PCI4410 signaled a system error to the host.
13	MABORT	RCU	Received master abort. This bit is set to 1 when a cycle initiated by the PCI4410 on the PCI bus has been terminated by a master abort.
12	TABORT_REC	RCU	Received target abort. This bit is set to 1 when a cycle initiated by the PCI4410 on the PCI bus is terminated by a target abort.
11	TABORT_SIG	RCU	Signaled target abort. This bit is set to 1 by the PCI4410 when it terminates a transaction on the PCI bus with a target abort.
10−9	PCI_SPEED	R	DEVSEL timing. These bits encode the timing of $\overline{\text{DEVSEL}}$ and are hardwired 01b indicating that the PCI4410 asserts this signal at a medium speed on non-configuration cycle accesses.
8	DATAPAR	RCU	Data parity error detected. This bit is set to 1 when the following conditions have been met: a. $\overline{\text{PERR}}$ was asserted by any PCI device including the PCI4410. b. The PCI4410 was the bus master during the data parity error. c. The parity error response bit is set to 1 in the command register.
7	FBB_CAP	R	Fast back-to-back capable. The PCI4410 cannot accept fast back-to-back transactions; thus, this bit is hardwired to 0.
6	UDF	R	UDF supported. The PCI4410 does not support the user-definable features; thus, this bit is hardwired to 0.
5	66MHZ	R	66-MHz capable. The PCI4410 operates at a maximum PCLK frequency of 33 MHz; therefore, this bit is hardwired to 0.
4	CAPLIST	R	Capabilities list. This bit returns 1 when read, and indicates that capabilities additional to standard PCI are implemented. The linked list of PCI power management capabilities is implemented in this function.
3−0	RSVD	R	Reserved. These bits return 0s when read.

8.6 Class Code and Revision ID Register

This read-only register categorizes the PCI4410 as a serial bus controller (0Ch), controlling an IEEE1394 bus (00h), with an OHCI programming model (10h). Furthermore, the TI chip revision is indicated in the lower byte. See Table 8−5 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Class code and revision ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Class code and revision ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1

Register: **Class code and revision ID**
Type: Read-only
Offset: 08h
Default: 0C00 1001h

Table 8−5. Class Code and Revision ID Register

BIT	SIGNAL	TYPE	FUNCTION
31−24	BASECLASS	R	Base class. This field returns 0Ch when read, which broadly classifies the function as a serial bus controller.
23−16	SUBCLASS	R	Sub class. This field returns 00h when read, which specifically classifies the function as controlling an IEEE1394 serial bus.
15−8	PGMIF	R	Programming interface. This field returns 10h when read, which indicates that the programming model is compliant with the 1394 OHCI specification.
7−0	CHIPREV	R	Silicon revision. This field returns the silicon revision of the PCI4410.

8.7 Latency Timer and Class Cache Line Size Register

The latency timer and class cache line size register is programmed by host BIOS to indicate system cache line size and the latency timer associated with the PCI4410. See Table 8−6 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Latency timer and class cache line size															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Latency timer and class cache line size**
Type: Read/Write
Offset: 0Ch
Default: 0000h

Table 8−6. Latency Timer and Class Cache Line Size Register

BIT	SIGNAL	TYPE	FUNCTION
15−8	LATENCY_TIMER	R/W	PCI latency timer. The value in this register specifies the latency timer for the PCI4410, in units of PCI clock cycles. When the PCI4410 is a PCI bus initiator and asserts FRAME, the latency timer begins counting from zero. If the latency timer expires before the PCI4410 transaction has terminated, then the PCI4410 terminates the transaction when its GNT is deasserted.
7−0	CACHELINE_SZ	R/W	Cache line size. This value is used by the PCI4410 during memory write and invalidate, memory read line, and memory read multiple transactions.

8.8 Header Type and BIST Register

The header type and BIST register indicates that this function is part of a multifunction device, and has a standard PCI header type and no built-in self-test. See Table 8−7 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Header type and BIST															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Header type and BIST**
Type: Read-only
Offset: 0Eh
Default: 0000h

Table 8−7. Header Type and BIST Register

BIT	SIGNAL	TYPE	FUNCTION
15−8	BIST	R	Built-in self-test. The PCI4410 does not include a built-in self-test, and this field returns 00h when read.
7−0	HEADER_TYPE	R	PCI header type. The PCI4410 includes the standard PCI header, and this is communicated by returning 00h when this field is read.

8.9 Open HCI Registers Base Address Register

The open HCI registers base address register is programmed with a base address referencing the memory mapped OHCI control. When BIOS writes all 1s to this register, the value read back is FFFF F800h, indicating that at least 2K bytes of memory address space are required for the OHCI registers. See Table 8−8 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Open HCI registers base address															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Open HCI registers base address															
Type	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Open HCI registers base address**
Type: Read-only**,** Read/Write
Offset: 10h
Default: 0000 0000h

Table 8−8. Open HCI Registers Base Address Register

BIT	SIGNAL	TYPE	FUNCTION
31−11	OHCIREG_PTR	R/W	Open HCI register pointer. Specifies the upper 21 bits of the 32-bit OHCI register base address.
10−4	OHCI_SZ	R	Open HCI register size. This field returns 0s when read, and indicates that the OHCI registers require a 2-Kbyte region of memory.
3	OHCI_PF	R	OHCI register prefetch. This bit returns 0, indicating the OHCI registers are nonprefetchable.
2−1	OHCI_MEMTYPE	R	Open HCI memory type. This field returns 0s when read, and indicates that the base register is 32 bits wide and mapping can be done anywhere in the 32-bit memory space.
0	OHCI_MEM	R	OHCI memory indicator. This bit returns 0, indicating the OHCI registers are mapped into system memory space.

8.10 TI Extension Base-Address Register

The TI extension base-address register is programmed with a base address referencing the memory-mapped TI extension registers. See Table 8−9 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TI extension base-address															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TI extension base-address															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **TI extension base-address**
Type: Read-only
Offset: 14h
Default: 0000 0000h

Table 8−9. TI Extension Base-Address Register

BIT	SIGNAL	TYPE	FUNCTION
31−11	TI_EXTREG_PTR	R	TI extension register pointer. Specifies the upper 20 bits of the 32-bit TI extension register base address.
10−4	TI_SZ	R	TI extension register size. This field returns 0s when read, and indicates that the TI extension registers require a 2-Kbyte region of memory.
3	TI_PF	R	TI extension register prefetch. This bit returns 0, indicating the TI extension registers are nonprefetchable.
2−1	TI_MEMTYPE	R	TI memory type. This field returns 0s when read, and indicates that the base register is 32 bits wide and mapping can be done anywhere in the 32-bit memory space.
0	TI_MEM	R	TI memory indicator. This bit returns 0, indicating the TI extension registers are mapped into system memory space.

8.11 PCI Subsystem Identification Register

The PCI subsystem identification register is used for subsystem and option card identification purposes. This register can be initialized from the serial EEPROM or can be written using the subsystem access register. See Table 8−10 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCI subsystem identification															
Type	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCI subsystem identification															
Type	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **PCI subsystem identification**
Type: Read/Update
Offset: 2Ch
Default: 0000 0000h

Table 8−10. PCI Subsystem Identification Register

BIT	SIGNAL	TYPE	FUNCTION
31−16	OHCI_SSID	RU	Subsystem device ID. This field indicates the subsystem device ID.
15−0	OHCI_SSVID	RU	Subsystem vendor ID. This field indicates the subsystem vendor ID.

8.12 PCI Power Management Capabilities Pointer Register

The PCI power management capabilities pointer register provides a pointer into the PCI configuration header where the PCI power management register block resides. PCI4410 configuration header doublewords at 44h and 48h provide the power management registers. This register is read-only and returns 44h when read.

Bit	7	6	5	4	3	2	1	0
Name	PCI power management capabilities pointer							
Type	R	R	R	R	R	R	R	R
Default	0	1	0	0	0	1	0	0

Register: **PCI power management capabilities pointer**
Type: Read-only
Offset: 34h
Default: 44h

8.13 Interrupt Line and Interrupt Pin Registers

The interrupt line and interrupt pin registers are used to communicate interrupt line routing information. See Table 8−11 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Interrupt line and interrupt pin															
Type	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Registers: **Interrupt line and interrupt pin**
Type: Read-only, Read/Write
Offset: 3Ch
Default: 0200h

Table 8−11. Interrupt Line and Interrupt Pin Registers

BIT	SIGNAL	TYPE	FUNCTION
15−8	INTR_PIN	R	Interrupt pin register. This register returns 01h or 02h when read, indicating that the PCI4410 link function signals interrupts on the INTA or INTB terminal, respectively. If TIE_INTB_INTA (offset 80h, bit 29) is set to 1, then INTR_PIN byte reads 0000 0001b,which indicates the OHCI function is signaling on INTA.
7−0	INTR_LINE	R/W	Interrupt line register. This register is programmed by the system and indicates to the software which interrupt line the PCI4410 INTA is connected to.

8.14 MIN_GNT and MAX_LAT Registers

These registers are used to communicate to the system the desired setting of the latency timer register. If a serial ROM is detected, then the contents of this register are loaded through the serial ROM interface after a PCI reset. If no serial ROM is detected, then these registers return a default value that corresponds to MIN_GNT = 3, MAX_LAT = 4. See Table 8−12 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MIN_GNT and MAX_LAT															
Type	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU
Default	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1

Registers: **MIN_GNT and MAX_LAT**
Type: Read/Update
Offset: 3Eh
Default: 0403h

Table 8−12. MIN_GNT and MAX_LAT Registers

BIT	SIGNAL	TYPE	FUNCTION
15−8	MAX_LAT	RU	Maximum latency. The contents of this register may be used by host BIOS to assign an arbitration priority level to the PCI4410. The default for this register indicates that the PCI4410 may need to access the PCI bus as often as every 1/4 μs; thus, an extremely high priority level is requested. The contents of this field may also be loaded through the serial ROM.
7−0	MIN_GNT	RU	Minimum grant. The contents of this register may be used by host BIOS to assign a latency timer register value to the PCI4410. The default for this register indicates that the PCI4410 may need to sustain burst transfers for nearly 64 μs, thus requesting a large value be programmed in the PCI4410 latency timer register.

8.15 PCI OHCI Control Register

The PCI OHCI control register contains IEEE1394 Open HCI specific control bits. All bits in this register are read-only and return 0s, because no OHCI-specific control bits have been implemented.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCI OHCI control															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **PCI OHCI control**
Type: Read-only
Offset: 40h
Default: 0000h

8.16 Capability ID and Next Item Pointer Registers

The capability ID and next item pointer registers identify the linked list capability item, and provide a pointer to the next capability item, respectively. See Table 8−13 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Capability ID and next item pointer															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Register: **Capability ID and next item pointer**
Type: Read-only
Offset: 44h
Default: 0001h

Table 8−13. Capability ID and Next Item Pointer Registers

BIT	SIGNAL	TYPE	FUNCTION
15−8	NEXT_ITEM	R	Next item pointer. The PCI4410 supports only one additional capability that is communicated to the system through the extended capabilities list; thus, this field returns 00h when read.
7−0	CAPABILITY_ID	R	Capability identification. This field returns 01h when read, which is the unique ID assigned by the PCI SIG for PCI power management capability.

8.17 Power Management Capabilities Register

The power management capabilities register indicates the capabilities of the PCI4410 related to PCI power management. In summary, the D0, D2, and $D3_{hot}$ device states are supported. See Table 8−14 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Power management capabilities															
Type	RU	RU	RU	RU	RU	R	R	R	R	R	R	R	R	R	R	R
Default	0	1	1	0	0	1	0	0	0	0	0	1	0	0	0	1

Register: **Power management capabilities**
Type: Read/Update
Offset: 46h
Default: 6411h

Table 8−14. Power Management Capabilities Register

BIT	SIGNAL	TYPE	FUNCTION
15	PME_D3COLD	RU	$\overline{PME}$ support from $D3_{cold}$. When this bit is set to 1, the PCI4410 generates a $\overline{PME}$ wake event from $D3_{cold}$. This bit state is dependent upon PCI4410 V_{aux} implementation and may be configured by host software using the PCI miscellaneous configuration register.
14−11	PME_SUPPORT	RU	$\overline{PME}$ support. This four-bit field indicates the power states from which the PCI4410 may assert $\overline{PME}$. These four bits return a value of 1100b by default, indicating that $\overline{PME}$ may be asserted from the $D3_{hot}$ and D2 power states. Bit 13 may be modified by host software using the PCI miscellaneous configuration register.
10	D2_SUPPORT	R	D2 support. This bit returns a 1 when read, indicating that the PCI4410 supports the D2 power state.
9	D1_SUPPORT	R	D1 support. This bit returns a 0 when read, indicating that the PCI4410 does not support the D1 power state.
8	DYN_DATA	R	Dynamic data support. This bit returns a 0 when read, indicating that the PCI4410 does not report dynamic power consumption data.
7−6	RSVD	R	Reserved. These bits return 0s when read.
5	DSI	R	Device-specific initialization. This bit returns 0 when read, indicating that the PCI4410 does not require special initialization beyond the standard PCI configuration header before a generic class driver is able to use it.
4	AUX_PWR	R	Auxiliary power source. Since the PCI4410 supports $\overline{PME}$ generation in the $D3_{cold}$ device state and requires V_{aux}, this bit returns 1 when read.
3	PME_CLK	R	$\overline{PME}$ clock. This bit returns 0 when read indicating that no host bus clock is required for the PCI4410 to generate $\overline{PME}$.
2−0	PM_VERSION	R	Power management version. This field returns 001b when read, indicating that the PCI4410 is compatible with the registers described in the revision 1.0 *PCI Bus Power Management Specification*.

8.18 Power Management Control and Status Register

The power management control and status register implements the control and status of the PCI power management function. This register is not affected by the internally generated reset caused by the transition from the $D3_{hot}$ to D0 state. See Table 8−15 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Power management control and status															
Type	RC	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Power management control and status**
Type: Read-only, Read/Write, Read/Clear
Offset: 48h
Default: 0000h

Table 8−15. Power Management Control and Status Register

BIT	SIGNAL	TYPE	FUNCTION
15	PME_STS	RC	This bit is set to 1 when the PCI4410 would normally be asserting the $\overline{PME}$ signal, independent of the state of bit 8 (PME_ENB). This bit is cleared by a writeback of 1, and this also clears the $\overline{PME}$ signal driven by the PCI4410. Writing a 0 to this bit has no effect.
14−9	DYN_CTRL	R	Dynamic data control. This bit field returns 0s when read because the PCI4410 does not report dynamic data.
8	PME_ENB	R/W	$\overline{PME}$ enable. This bit enables the function to assert $\overline{PME}$. If the bit is reset to 0, assertion of $\overline{PME}$ is disabled.
7−5	RSVD	R	Reserved. These bits return 0s when read.
4	DYN_DATA	R	Dynamic data. This bit returns 0 when read because the PCI4410 does not report dynamic data.
3−2	RSVD	R	Reserved. These bits return 0s when read.
1−0	PWR_STATE	R/W	Power state. This two-bit field is used to set the PCI4410 device power state, and is encoded as follows: 00 = Current power state is D0. 01 = Current power state is D1. 10 = Current power state is D2. 11 = Current power state is $D3_{hot}$.

8.19 Power Management Extension Register

The power management extension register provides extended power management features not applicable to the PCI4410; thus, it is read-only and returns 0 when read. See Table 8−16 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Power management extension															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Power management extension**
Type: Read-only
Offset: 4Ah
Default: 0000h

Table 8−16. Power Management Extension Register

BIT	SIGNAL	TYPE	FUNCTION
15−8	PM_DATA	R	Power management data. This bit field returns 0s when read because the PCI4410 does not report dynamic data.
7−0	PMCSR_BSE	R	Power management CSR − bridge support extensions. This field returns 0s because the PCI4410 does not provide P-to-P bridging.

8.20 PCI Miscellaneous Configuration Register

The PCI miscellaneous configuration register provides miscellaneous PCI-related configuration. See Table 8−17 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCI miscellaneous configuration															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PCI miscellaneous configuration															
Type	R/W	R	R/W	R	R	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W
Default	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0

Register: **PCI miscellaneous configuration**
Type: Read-only, Read/Write
Offset: F0h
Default: 0000 2400h

Table 8−17. PCI Miscellaneous Configuration Register

BIT	SIGNAL	TYPE	FUNCTION
31−16	RSVD	R	Reserved. These bits return 0s when read.
15	PME_D3COLD	R/W	$\overline{\text{PME}}$ support from $D3_{cold}$. This bit is used to program the corresponding read-only value read from power management capabilities. This bit retains state through PCI reset and D3–D0 transitions.
14	RSVD	R	Reserved. This bit returns 0 when read.
13	PME_SUPPORT_D2	R/W	$\overline{\text{PME}}$ support. This bit is used to program the corresponding read-only value read from power management capabilities. If wake up from the D2 power state implemented in PCI4410 is not desired, then this bit may be reset to 0 to indicate to power management software that wake up from D2 is not supported. This bit retains state through PCI reset and D3 – D0 transitions.
12−11	RSVD	R	Reserved. These bits return 0s when read.
10	D2_SUPPORT	R/W	D2 support. This bit is used to program the corresponding read-only value read from power management capabilities. If the D2 power state implemented in PCI4410 is not desired, then this bit may be reset to 0 to indicate to power management software that D2 is not supported. This bit retains state through PCI reset and D3–D0 transitions.
9−5	RSVD	R	Reserved. Bits 9−5 return 0s when read.
4	DISABLE_PCI_ TARGET_ABORT	R/W	When bit 4 is set to 1, the OSCI function returns indeterminate data instead of signaling target abort. The default (0) allows the OSCI function to signal target abort.
3	RSVD	R/W	Reserved. This bit defaults to 0.
2	DISABLE_SCLKGATE	R/W	When this bit is set to 1, the internal SCLK runs identically with the chip input.
1	DISABLE_PCIGATE	R/W	When this bit is set to 1, the internal PCI clock runs identically with the chip input.
0	KEEP_PCLK	R/W	When this bit is set to 1, the PCI clock is always kept running through the $\overline{\text{CLKRUN}}$ protocol. When reset to 0, the PCI clock may be stopped using $\overline{\text{CLKRUN}}$.

8.21 Link Enhancement Control Register

The link enhancement control register implements TI proprietary bits that are initialized by software or by a serial EEPROM if present. After these bits are set to 1, their functionality is enabled only if the APHYENHANCEENABLE bit (bit 22) in the host controller control register (offset 50h/54h, see Section 9.16) is set to 1. See Table 8−18 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Link enhancement control															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Link enhancement control															
Type	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R
Default	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Link enhancement control**
Type: Read-only, Read/Write
Offset: F4h
Default: 0000 1000h

Table 8−18. Link Enhancement Control Register

BIT	SIGNAL	TYPE	FUNCTION
31−14	RSVD	R	Reserved. These bits return 0 when read.
13−12	ATX_THRESH	R/W	This bit field sets the initial AT threshold value, which is used until the AT FIFO is underrun. When PCI4410 retries the packet, it uses a 2K-byte threshold resulting in store-and-forward operation. 00 = Threshold ~ 2 Kbytes resulting in store-and-forward operation 01 = Threshold ~ 1.7 Kbytes (default) 10 = Threshold ~ 1 K 11 = Threshold ~ 512 bytes
11−10	RSVD	R	Reserved. This bit returns 0 when read.
9	ENAB_AUDIO_TS	R/W	Enable audio/music CIP timestamp enhancement. When this bit is set to 1, the enhancement is enabled for audio/music CIP transmit streams (FMT = 10h).
8	ENAB_DV_TS	R/W	Enable DV CIP timestamp enhancement. When this bit is set to 1, the enhancement is enabled for DV CIP transmit streams (FMT = 00h).
7	ENAB_UNFAIR	R/W	Enable asynchronous priority requests. OHCI-Lynx (TSB12LV22) compatible.
6	RSVD	R	This reserved field will not be assigned in PCI4410 follow-on products since this bit location loaded by the serial ROM from the *enhancements* field corresponds to HCControl.programPhyEnable in open HCI register space.
5−3	RSVD	R	Reserved. These bits return 0 when read.
2	ENAB_INSERT_IDLE	R/W	Enable insert idle. OHCI-Lynx (TSB12LV22) compatible.
1	ENAB_ACCEL	R/W	Enable acceleration enhancements. OHCI-Lynx (TSB12LV22) compatible.
0	RSVD	R	Reserved. This bit returns 0 when read.

8.22 Subsystem Access Identification Register

The subsystem access identification register is used for system and option card identification purposes. The contents of this register are aliased to subsystem identification register at address 2Ch. See Table 8−19 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Subsystem access identification															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Subsystem access identification															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Subsystem access identification**
Type: Read/Write
Offset: F8h
Default: 0000 0000h

Table 8−19. Subsystem Access Identification Register

BIT	SIGNAL	TYPE	FUNCTION
31−16	SUBDEV_ID	R/W	Subsystem device ID. This field indicates the subsystem device ID.
15−0	SUBVEN_ID	R/W	Subsystem vendor ID. This field indicates the subsystem vendor ID.

8.23 GPIO Control Register

The GPIO control register has the control and status bits for GPIO0, GPIO1, GPIO2 and GPIO3 ports. Upon reset, GPIO0 and GPIO1 default to bus manager contender (BMC) and link power status terminals, respectively. The BMC terminal can be configured as GPIO0 by setting bit 7 (DISABLE_BMC) to 1. The LPS terminal can be configured as GPIO1 by setting bit 15 (DISABLE_LPS) to 1. See Table 8−20 for a complete description of the register contents.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	GPIO control															
Type	R	R	R/W	R/W	R	R	R	R/W	R	R	R/W	R/W	R	R	R	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	GPIO control															
Type	R/W	R	R/W	R/W	R	R	R	R/W	R/W	R	R/W	R/W	R	R	R	R/W
Default	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0

Register: **GPIO control**
Type: Read-only, Read/Write
Offset: FCh
Default: 0000 1010h

Table 8−20. GPIO Control Register

BIT	SIGNAL	TYPE	FUNCTION
31−30	RSVD	R	Reserved. These bits return 0s when read.
29	GPIO_INV3	R/W	GPIO3 polarity invert. This bit controls the input/output polarity control of GPIO3. 0 = Noninverted (default) 1 = Inverted
28	GPIO_ENB3	R/W	GPIO3 enable control. This bit controls the output enable for GPIO3. 0 = High-impedance output (default) 1 = Output enabled
27−25	RSVD	R	Reserved. These bits return 0s when read.
24	GPIO_DATA3	R/W	GPIO3 data. When GPIO3 output is enabled, the value written to this bit represents the logical data driven to the GPIO3 terminal.
23−22	RSVD	R	Reserved. These bits return 0s when read.
21	GPIO_INV2	R/W	GPIO2 polarity invert. This bit controls the input/output polarity control of GPIO2. 0 = Noninverted (default) 1 = Inverted
20	GPIO_ENB2	R/W	GPIO2 enable control. This bit controls the output enable for GPIO2. 0 = High-impedance output (default) 1 = Output enabled
19−17	RSVD	R	Reserved. These bits return 0s when read.
16	GPIO_DATA2	R/W	GPIO2 data. When GPIO2 output is enabled, the value written to this bit represents the logical data driven to the GPIO2 terminal.
15	DISABLE_LPS	R/W	Disable link power status (LPS). This bit configures this terminal as 0 = LPS (default) 1 = GPIO1
14	RSVD	R	Reserved. This bit returns 0 when read.
13	GPIO_INV1	R/W	GPIO1 polarity invert. When DISABLE_LPS bit is set to 1, this bit controls the input/output polarity control of GPIO1. 0 = Noninverted (default) 1 = Inverted

Table 8−20. GPIO Control Register (Continued)

BIT	SIGNAL	TYPE	FUNCTION
12	GPIO_ENB1	R/W	GPIO1 enable control. When the DISABLE_LPS bit is set to 1, this bit controls the output enable for GPIO1 0 = High-impedance output 1 = Output enabled (default)
11−9	RSVD	R	Reserved. These bits return 0s when read.
8	GPIO_DATA1	R/W	GPIO1 data. When the DISABLE_LPS bit is set to 1 and GPIO1 output is enabled, the value written to this bit represents the logical data driven to the GPIO1 terminal.
7	DISABLE_BMC	R/W	Disable bus manager contender (BMC). This bit configures this terminals as bus master contender or GPIO0. 0 = BMC (default) 1 = GPIO0
6	RSVD	R	Reserved. This bit returns 0 when read.
5	GPIO_INV0	R/W	GPIO0 polarity invert. When bit 7 (DISABLE_BMC) is set to 1, this bit controls the input/output polarity control of for GPIO0. 0 = Non-inverted (default) 1 = Inverted
4	GPIO_ENB0	R/W	GPIO0 enable control. When DISABLE_BMC bit is set to 1, this bit controls the output enable for GPIO0 0 = High-impedance output 1 = Output enabled (default)
3−1	RSVD	R	Reserved. These bits return 0s when read.
0	GPIO_DATA0	R/W	GPIO0 data. When the DISABLE_BMC bit is set to 1 and GPIO0 output is enabled, the value written to this bit represents the logical data driven to the GPIO0 terminal.

9 Open HCI Registers

The open HCI registers defined by the *IEEE1394 Open HCI Specification* are memory-mapped into a 2-Kbyte region of memory pointed to by the OHCI base address register at offset 10h in PCI configuration space. These registers are the primary interface for controlling the PCI4410 IEEE1394 link function.

This section provides the register interface and bit descriptions. There are several set and clear register pairs in this programming model, which are implemented to solve various issues with typical read-modify-write control registers. There are two addresses for a set/clear register: RegisterSet and RegisterClear. See Table 9−1 for a register listing. A 1 written to RegisterSet causes the corresponding bit in the set/clear register to be set to 1, whereas a 0 leaves the corresponding bit unaffected. A 1 written to RegisterClear causes the corresponding bit in the set/clear register to be reset to 0, whereas a 0 leaves the corresponding bit in the set/clear register unaffected.

Typically, a read from either RegisterSet or RegisterClear returns the value of the set/clear register. However, sometimes reading the RegisterClear provides a masked version of the set/clear register. The interrupt event register is an example of this behavior.

Table 9−1. Open HCI Register Map

DMA CONTEXT	REGISTER NAME	ABBREVIATION	OFFSET
—	OHCI version	Version	00h
	Global unique ID ROM	GUID_ROM	04h
	Asynchronous transmit retries	ATRetries	08h
	CSR data	CSRData	0Ch
	CSR compare data	CSRCompareData	10h
	CSR control	CSRControl	14h
	Configuration ROM header	ConfigROMhdr	18h
	Bus identification	BusID	1Ch
	Bus options	BusOptions	20h
	Global unique ID high	GUIDHi	24h
	Global unique ID low	GUIDLo	28h
	Reserved	—	2Ch
	Reserved	—	30h
	Configuration ROM map	ConfigROMmap	34h
	Posted write address low	PostedWriteAddressLo	38h
	Posted write address high	PostedWriteAddressHi	3Ch
	Vendor identification	VendorID	40h
	Reserved	—	44h − 4Ch

Table 9−1. Open HCI Register Map (Continued)

DMA CONTEXT	REGISTER NAME	ABBREVIATION	OFFSET
—	Host controller control	HCControlSet	50h
		HCControlClr	54h
	Reserved	—	58h
	Reserved	—	5Ch
Self ID	Reserved	—	60h
	Self ID buffer	SelfIDBuffer	64h
	Self ID count	SelfIDCount	68h
	Reserved	—	6Ch
—	Isochronous receive channel mask high	IRChannelMaskHiSet	70h
		IRChannelMaskHiClear	74h
	Isochronous receive channel mask low	IRChannelMaskLoSet	78h
		IRChannelMaskLoClear	7Ch
	Interrupt event	IntEventSet	80h
		IntEventClear	84h
	Interrupt mask	IntMaskSet	88h
		IntMaskClear	8Ch
	Isochronous transmit interrupt event	IsoXmitIntEventSet	90h
		IsoXmitIntEventClear	94h
	Isochronous transmit interrupt mask	IsoXmitIntMaskSet	98h
		IsoXmitIntMaskClear	9Ch
—	Isochronous receive interrupt event	IsoRecvIntEventSet	A0h
		IsoRecvIntEventClear	A4h
	Isochronous receive interrupt mask	IsoRecvIntMaskSet	A8h
		IsoRecvIntMaskClear	ACh
	Reserved		B0−D8h
	Fairness control	FairnessControl	DCh
	Link control	LinkControlSet	E0h
		LinkControlClear	E4h
	Node identification	NodeID	E8h
	PHY layer control	PhyControl	ECh
	Isochronous cycle timer	IsoCycleTimer	F0h
	Reserved	—	F4h − FCh
	Asynchronous request filter high	AsyncRequestFilterHiSet	100h
		AsyncRequestFilterHiClear	104h
	Asynchronous request filter low	AsyncRequestFilterLoSet	108h
		AsyncRequestFilterloClear	10Ch
	Physical request filter high	PhysicalRequestFilterHiSet	110h
		PhysicalRequestFilterHiClear	114h
	Physical request filter low	PhysicalRequestFilterLoSet	118h
		PhysicalRequestFilterloClear	11Ch
	Physical upper bound	PhysicalUpperBound	120h
	Reserved	—	124h − 17Ch

Table 9−1. Open HCI Register Map (Continued)

DMA CONTEXT	REGISTER NAME	ABBREVIATION	OFFSET
Asynchronous request transmit [ATRQ]	Context control	ContextControlSet	180h
		ContextControlClear	184h
	Reserved	—	188h
	Command pointer	CommandPtr	18Ch
Asynchronous response transmit [ATRS]	Reserved	—	190h − 19Ch
	Context control	ContextControlSet	1A0h
		ContextControlClear	1A4h
	Reserved	—	1A8h
	Command pointer	CommandPtr	1ACh
Asynchronous request receive [ARRQ]	Reserved	—	1B0h − 1BCh
	Context control	ContextControlSet	1C0h
		ContextControlClear	1C4h
	Reserved	—	1C8h
	Command pointer	CommandPtr	1CCh
Asynchronous response receive [ARRS]	Reserved	—	1D0h − 1DCh
	Context control	ContextControlSet	1E0h
		ContextControlClear	1E4h
	Reserved	—	1E8h
	Command pointer	CommandPtr	1ECh
Isochronous transmit context n n = 0, 1, 2, 3, … 7	Reserved	—	1F0h − 1FCh
	Context control	ContextControlSet	200h + 16*n
		ContextControlClear	204h + 16*n
	Reserved	—	208h + 16*n
Isochronous receive context n n = 0, 1, 2, 3, 4	Command pointer	CommandPtr	20Ch + 16*n
	Context control	ContextControlSet	400h + 32*n
		ContextControlClear	404h + 32*n
	Reserved	—	408h + 32*n
	Command pointer	CommandPtr	40Ch + 32*n
	Context match	ContextMatch	410h + 32*n

9.1 OHCI Version Register

This register indicates the OHCI version support, and whether or not the serial ROM is present. See Table 9−2 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OHCI version															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	OHCI version															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Register: **OHCI version**
Type: Read-only
Offset: 00h
Default: 0001 0000h

Table 9−2. OHCI Version Register

BIT	SIGNAL	TYPE	FUNCTION
31−25	RSVD	R	Reserved. Bits 31−25 return 0s when read.
24	GUID_ROM	R	The PCI4410 sets this bit to 1 if the serial ROM is detected. If the serial ROM is present, then the Bus_Info_Block is automatically loaded on hardware reset.
23−16	VERSION	R	Major version of the open HCI. The PCI4410 is compliant with the OHCI specification version 1.00; thus, this field reads 01h.
15−8	RSVD	R	Reserved. Bits 15−8 return 0s when read.
7−0	REVISION	R	Minor version of the open HCI. The PCI4410 is compliant with the OHCI specification version 1.00; thus, this field reads 00h.

9.2 GUID ROM Register

This register is used to access the serial ROM and is only applicable if the GUID_ROM bits are set to 1. See Table 9−3 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	GUID ROM															
Type	RSU	R	R	R	R	R	RSU	R	RU	RU	RU	RU	RU	RU	RU	RU
Default	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X
Xit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	GUID ROM															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **GUID ROM**
Type: Read-only, Read/Set/Update, Read/Update
Offset: 04h
Default: 00XX 0000h

Table 9−3. GUID ROM Register

BIT	SIGNAL	TYPE	FUNCTION
31	ADDRRESET	RSU	Software sets this bit to 1 to reset the GUID ROM address to 0. When the PCI4410 completes the reset, it clears this bit. The PCI4410 does not automatically fill bits 23−16 (RDDATA field) with the 0th byte.
30−26	RSVD	R	Reserved. Bits 30−26 return 0s when read.
25	RDSTART	RSU	A read of the currently addressed byte is started when this bit is set to 1. This bit is automatically cleared when the PCI4410 completes the read of the currently addressed GUID ROM byte.
24	RSVD	R	Reserved. Bit 24 returns 0 when read.
23−16	RDDATA	RU	This field represents the data read from the GUID ROM.
15−0	RSVD	R	Reserved. Bits 15−0 return 0s when read.

9.3 Asynchronous Transmit Retries Register

This register indicates the number of times the PCI4410 will attempt a retry for asynchronous DMA request transmit and for asynchronous physical and DMA response transmit. See Table 9−4 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Asynchronous transmit retries															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Asynchronous transmit retries															
Type	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Asynchronous transmit retries**
Type: Read-only, Read/Write
Offset: 08h
Default: 0000 0000h

Table 9−4. Asynchronous Transmit Retries Register

BIT	SIGNAL	TYPE	FUNCTION
31−29	SECONDLIMIT	R	The second limit field returns 0 when read, because outbound dual-phase retry is not implemented.
28−16	CYCLELIMIT	R	The cycle limit field returns 0 when read, because outbound dual-phase retry is not implemented.
15−12	RSVD	R	Reserved. Bits 15−12 return 0 when read.
11−8	MAXPHYSRESPRETRIES	R/W	The MAXPHYSRESPRETRIES field tells the physical response unit how many times to attempt to retry the transmit operation for the response packet when a busy acknowledge or ack_data_error is received from the target node.
7−4	MAXATRESPRETRIES	R/W	The MAXATRESPRETRIES field tells the asynchronous transmit response unit how many times to attempt to retry the transmit operation for the response packet when a busy acknowledge or ack_data_error is received from the target node.
3−0	MAXATREQRETRIES	R/W	The MAXATREQRETRIES field tells the asynchronous transmit DMA request unit how many times to attempt to retry the transmit operation for the response packet when a busy acknowledge or ack_data_error is received from the target node.

9.4 CSR Data Register

This register is used to access the bus management CSR registers from the host through compare-swap operations. This register contains the data to be stored in a CSR if the compare is successful.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CSR data															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CSR data															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **CSR data**
Type: Read-only
Offset: 0Ch
Default: 0000 0000h

9.5 CSR Compare Register

This register is used to access the bus management CSR registers from the host through compare-swap operations. This register contains the data to be compared with the existing value of the CSR resource.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CSR compare															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CSR compare															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **CSR compare**
Type: Read-only
Offset: 10h
Default: 0000 0000h

9.6 CSR Control Register

This register is used to access the bus management CSR registers from the host through compare-swap operations. This register is used to control the compare-swap operation and select the CSR resource. See Table 9−5 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CSR control															
Type	R/C	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CSR control															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **CSR control**
Type: Read-only, Read/Update, Read/Write
Offset: 14h
Default: 0000 0000h

Table 9−5. CSR Control Register

BIT	SIGNAL	TYPE	FUNCTION
31	CSRDONE	R/C	This bit is set to 1 by the PCI4410 when a compare-swap operation is complete. It is reset to 0 whenever this register is written.
30−2	RSVD	R	Reserved. Bits 30−2 return 0s when read.
1−0	CSRSEL	R/W	This field selects the CSR resource as follows: 00 = BUS_MANAGER_ID 01 = BANDWIDTH_AVAILABLE 10 = CHANNELS_AVAILABLE_HI 11 = CHANNELS_AVAILABLE_LO

9.7 Configuration ROM Header Register

This register externally maps to the first quadlet of the 1394 configuration ROM, offset FFFF F000 0400h. See Table 9−6 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Configuration ROM header															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Configuration ROM header															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **Configuration ROM header**
Type: Read/Write
Offset: 18h
Default: 0000 XXXXh

Table 9−6. Configuration ROM Header Register

BIT	SIGNAL	TYPE	FUNCTION
31−24	INFO_LENGTH	R/W	IEEE1394 bus management field. Must be valid when HCControl.linkEnable bit is set to 1.
23−16	CRC_LENGTH	R/W	IEEE1394 bus management field. Must be valid when HCControl.linkEnable bit is set to 1.
15−0	ROM_CRC_VALUE	R/W	IEEE1394 bus management field. Must be valid at any time the HCControl.linkEnable bit is set to 1. The reset value is undefined if no serial ROM is present. If a serial ROM is present, then this field is loaded from the serial ROM.

9.8 Bus Identification Register

This register externally maps to the first quadlet in the Bus_Info_Block, and contains the constant 3133 3934h, which is the ASCII value of 1394.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Bus identification															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Bus identification															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	1	1	1	0	0	1	0	0	1	1	0	1	0	0

Register: **Bus identification**
Type: Read-only
Offset: 1Ch
Default: 3133 3934h

9.9 Bus Options Register

This register externally maps to the second quadlet of the Bus_Info_Block. See Table 9−7 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Bus options															
Type	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	X	X	X	X	0	0	0	0	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Bus options															
Type	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R	R	R	R	R	R
Default	1	0	1	0	0	0	0	0	X	X	0	0	0	0	1	0

Register: **Bus options**
Type: Read-only, Read/Write
Offset: 20h
Default: X0XX A0X2h

Table 9−7. Bus Options Register

BIT	SIGNAL	TYPE	FUNCTION
31	IRMC	R/W	Isochronous resource manager capable. IEEE1394 bus management field. Must be valid when HCControl.linkEnable bit is set to 1.
30	CMC	R/W	Cycle master capable. IEEE1394 bus management field. Must be valid when HCControl.linkEnable bit is set to 1.
29	ISC	R/W	Isochronous support capable. IEEE1394 bus management field. Must be valid when HCControl.linkEnable bit is set to 1.
28	BMC	R/W	Bus manager capable. IEEE1394 bus management field. Must be valid when HCControl.linkEnable bit is set to 1.
27	PMC	R/W	IEEE1394 bus management field. Must be valid when HCControl.linkEnable bit is set to 1.
26−24	RSVD	R	Reserved. Bits 26−24 return 0 when read.
23−16	CYC_CLK_ACC	R/W	Cycle master clock accuracy in parts per million. IEEE1394 bus management field. Must be valid when HCControl.linkEnable bit is set to 1.
15−12	MAX_REC	R/W	IEEE 1394 bus management field. Hardware initializes this field to indicate the maximum number of bytes in a block request packet that is supported by the implementation. This value, max_rec_bytes must be 512 or greater, and is calculated by 2^(max_rec + 1). Software may change max_rec; however, this field must be valid at any time the HCControl.linkEnable bit is set to 1. A received block write request packet with a length greater than max_rec_bytes may generate an ack_type_error. This field is not affected by a soft reset and defaults to a value indicating 2048 bytes on hard reset.
11−8	RSVD	R	Reserved. Bits 11−8 return 0s when read.
7−6	G	R/W	Generation counter. This field is incremented if any portion of the configuration ROM has incremented since the prior bus reset.
5−3	RSVD	R	Reserved. Bits 5−3 return 0s when read.
2−0	LNK_SPD	R	Link speed. This field returns 010, indicating that the link speeds of 100, 200, and 400 Mbits/s are supported.

9.10 GUID High Register

This register represents the upper quadlet in a 64-bit global unique ID (GUID) which maps to the third quadlet in the Bus_Info_Block. This register contains node_vendor_ID and chip_ID_hi fields. This register initializes to 0s on a hardware reset, which is an illegal GUID value. If a serial ROM is detected, then the contents of this register are loaded through the serial ROM interface after a PCI reset. At that point, the contents of this register cannot be changed. If no serial ROM is detected, then this register may be written once to set the value of this register. At that point, the contents of this register cannot be changed.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	GUID high															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	GUID high															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **GUID high**
Type: Read-only
Offset: 24h
Default: 0000 0000h

9.11 GUID Low Register

This register represents the lower quadlet in a 64-bit global unique ID (GUID) which maps to chip_ID_lo in the Bus_Info_Block. This register initializes to 0s on a hardware reset and behaves identically to the GUID high register.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	GUID low															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	GUID low															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **GUID low**
Type: Read-only
Offset: 28h
Default: 0000 0000h

9.12 Configuration ROM Mapping Register

This register contains the start address within system memory that will map to the start address of 1394 configuration ROM for this node. See Table 9−8 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Configuration ROM mapping															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Configuration ROM mapping															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Configuration ROM mapping**
Type: Read-only, Read/Write
Offset: 34h
Default: 0000 0000h

Table 9−8. Configuration ROM Mapping Register

BIT	SIGNAL	TYPE	FUNCTION
31−10	CONFIGROMADDR	R/W	If a quadlet read request to 1394 offset FFFF F000 0400h through offset FFFF F000 07FFh is received, then the low order 10 bits of the offset are added to this register to determine the host memory address of the read request.
9−0	RSVD	R	Reserved. Bits 9−0 return 0s when read.

9.13 Posted Write Address Low Register

This register is used to communicate error information if a write request is posted and an error occurs while the posted data packet is being written. See Table 9−9 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Posted write address low															
Type	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Posted write address low															
Type	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **Posted write address low**
Type: Read/Update
Offset: 38h
Default: XXXX XXXXh

Table 9−9. Posted Write Address Low Register

BIT	SIGNAL	TYPE	FUNCTION
31−0	OFFSETLO	RU	The lower 32 bits of the 1394 destination offset of the write request that failed

9.14 Posted Write Address High Register

This register is used to communicate error information if a write request is posted and an error occurs while the posted data packet is being written. See Table 9−10 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Posted write address high															
Type	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Posted write address high															
Type	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **Posted write address high**
Type: Read/Update
Offset: 3Ch
Default: XXXX XXXXh

Table 9−10. Posted Write Address High Register

BIT	SIGNAL	TYPE	FUNCTION
31−16	SOURCEID	RU	This bus and node number of the node that issued the write request that failed
15−0	OFFSETHI	RU	The upper 16 bits of the 1394 destination offset of the write request that failed

9.15 Vendor ID Register

The vendor ID register holds the company ID of an organization that specifies any vendor-unique registers. The PCI4410 does not implement Texas Instruments unique behavior with regards to open HCI. Thus this register is read-only and returns 0s when read.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Vendor ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Vendor ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Vendor ID**
Type: Read-only
Offset: 40h
Default: 0000 0000h

9.16 Host Controller Control Register

This set/clear register pair provides flags for controlling the PCI4410 link function. See Table 9−11 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Host controller control															
Type	R	RSC	R	R	R	R	R	R	RC	RSC	R	R	RSC	RSC	RSC	RSCU
Default	0	X	0	0	0	0	0	0	0	0	0	0	0	X	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Host controller control															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Host controller control**
Type: Read/Set/Clear/Update
Offset: 50h set register
54h clear register
Default: X00X 0000h

Table 9−11. Host Controller Control Register

BIT	SIGNAL	TYPE	FUNCTION
31	RSVD	R	Reserved. Bit 31 returns 0 when read.
30	NOBYTESWAPDATA	RSC	This bit is used to control whether physical accesses to locations outside the PCI4410 itself as well as any other DMA data accesses should be swapped.
29−24	RSVD	R	Reserved. Bits 29−24 return 0s when read.
23	PROGRAMPHYENABLE	RC	This bit informs upper level software that lower level software has consistently configured the p1394a enhancements in the Link and PHY. When this bit is 1, generic software such as the OHCI driver is responsible for configuring p1394a enhancements in the PHY and the APHYENHANCEENABLE bit in the PCI4410. When this bit is 0, the generic software may not modify the p1394a enhancements in the PCI4410 or PHY and cannot interpret the setting of APHYENHANCEENABLE. This bit can be initialized from serial EEPROM.
22	APHYENHANCEENABLE	RSC	When bits 23 (PROGRAMPHYENABLE) and 17 (LINKENABLE) are 1, the OHCI driver can set this bit to 1 to use all p1394a enhancements. When bit 23 (PROGRAMPHYENABLE) is 0, the software does not change PHY enhancements or the APHYENHANCEENABLE bit.
21−20	RSVD	R	Reserved. Bits 21 and 20 return 0s when read.
19	LPS	RSC	This bit is used to control the link power status. Software must set this bit to 1 to permit the link-PHY communication. A 0 prevents link-PHY communication.
18	POSTEDWRITEENABLE	RSC	This bit is used to enable (1) or disable (0) posted writes. Software should change this bit only when bit 17 (LINKENABLE) is 0.
17	LINKENABLE	RSC	This bit is cleared to 0 by a hardware reset or software reset. Software must set this bit to 1 when the system is ready to begin operation and then force a bus reset. This bit is necessary to keep other nodes from sending transactions before the local system is ready. When this bit is cleared, the PCI4410 is logically and immediately disconnected from the 1394 bus, no packets are received or processed, and no packets are transmitted.
16	SOFTRESET	RSCU	When this bit is set to 1, all PCI4410 states are reset, all FIFO's are flushed, and all OHCI registers are set to their hardware reset values unless otherwise specified. PCI registers are not affected by this bit. This bit remains set to 1 while the soft reset is in progress and reverts back to 0 when the reset has completed.
15−0	RSVD	R	Reserved. Bits 15−0 return 0s when read.

9.17 Self ID Buffer Pointer Register

This register points to the 2-Kbyte aligned base address of the buffer in host memory where the self ID packets will be stored during bus initialization. Bits 31−11 are read/write accessible.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Self ID buffer pointer															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Self ID buffer pointer															
Type	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Self ID buffer pointer**
Type: Read-only, Read/Write
Offset: 64h
Default: 0000 0000h

9.18 Self ID Count Register

This register keeps a count of the number of times the bus self ID process has occurred, flags self ID packet errors, and keeps a count of the amount of self ID data in the self ID buffer. See Table 9−12 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Self ID count															
Type	RU	R	R	R	R	R	R	R	RU	RU	RU	RU	RU	RU	RU	RU
Default	X	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Self ID count															
Type	R	R	R	R	R	RU	RU	RU	RU	RU	RU	RU	RU	RU	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Self ID count**
Type: Read/Update
Offset: 68h
Default: X0XX 0000h

Table 9−12. Self ID Count Register

BIT	SIGNAL	TYPE	FUNCTION
31	SELFIDERROR	RU	When this bit is 1, an error was detected during the most recent self ID packet reception. The contents of the self ID buffer are undefined. This bit is cleared after a self ID reception in which no errors are detected. Note that an error can be a hardware error or a host bus write error.
30−24	RSVD	R	Reserved. Bits 30−24 return 0s when read.
23−16	SELFIDGENERATION	RU	The value in this field increments each time a bus reset is detected. This field rolls over to 0 after reaching 255.
15−11	RSVD	R	Reserved. Bits 15−11 return 0s when read.
10−2	SELFIDSIZE	RU	This field indicates the number of quadlets that have been written into the self ID buffer for the current SELFIDGENERATION. This includes the header quadlet and the self ID data. This field is cleared to 0 when the self-ID reception begins.
1−0	RSVD	R	Reserved. Bits 1 and 0 return 0s when read.

9.19 ISO Receive Channel Mask High Register

This set/clear register is used to enable packet receives from the upper 32 isochronous data channels. A read from either the set register or clear register returns the value of the IRChannelMaskHi register. See Table 9−13 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ISO receive channel mask high															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ISO receive channel mask high															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **ISO receive channel mask high**
Type: Read/Set/Clear
Offset: 70h set register
74h clear register
Default: XXXX XXXXh

Table 9−13. ISO Receive Channel Mask High Register

BIT	SIGNAL	TYPE	FUNCTION
31	ISOCHANNEL63	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 63.
30	ISOCHANNEL62	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 62.
29	ISOCHANNEL61	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 61.
28	ISOCHANNEL60	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 60.
27	ISOCHANNEL59	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 59.
26	ISOCHANNEL58	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 58.
25	ISOCHANNEL57	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 57.
24	ISOCHANNEL56	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 56.
23	ISOCHANNEL55	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 55.
22	ISOCHANNEL54	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 54.
21	ISOCHANNEL53	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 53.
20	ISOCHANNEL52	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 52.
19	ISOCHANNEL51	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 51.
18	ISOCHANNEL50	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 50.
17	ISOCHANNEL49	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 49.
16	ISOCHANNEL48	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 48.
15	ISOCHANNEL47	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 47.
14	ISOCHANNEL46	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 46.
13	ISOCHANNEL45	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 45.
12	ISOCHANNEL44	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 44.
11	ISOCHANNEL43	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 43.
10	ISOCHANNEL42	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 42.
9	ISOCHANNEL41	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 41.
8	ISOCHANNEL40	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 40.
7	ISOCHANNEL39	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 39.
6	ISOCHANNEL38	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 38.
5	ISOCHANNEL37	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 37.

Table 9−13. ISO Receive Channel Mask High Register (Continued)

BIT	SIGNAL	TYPE	FUNCTION
4	ISOCHANNEL36	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 36.
3	ISOCHANNEL35	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 35.
2	ISOCHANNEL34	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 34.
1	ISOCHANNEL33	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 33.
0	ISOCHANNEL32	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 32.

9.20 ISO Receive Channel Mask Low Register

This set/clear register is used to enable packet receives from the lower 32 isochronous data channels. See Table 9−14 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ISO receive channel mask low															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ISO receive channel mask low															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **ISO receive channel mask low**
Type: Read/Set/Clear
Offset: 78h set register
7Ch clear register
Default: XXXX XXXXh

Table 9−14. ISO Receive Channel Mask Low Register

BIT	SIGNAL	TYPE	FUNCTION
31	ISOCHANNEL31	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 31.
30	ISOCHANNEL30	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 30.
⋮	⋮	⋮	Bits 29 through 2 follow the same pattern
1	ISOCHANNEL1	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 1.
0	ISOCHANNEL0	RSC	When set to 1, the PCI4410 is enabled to receive from ISO channel number 0.

9.21 Interrupt Event Register

This set/clear register reflects the state of the various PCI4410 interrupt sources. The interrupt bits are set to 1 by an asserting edge of the corresponding interrupt signal, or by writing a 1 in the corresponding bit in the set register. The only mechanism to reset the bits in this register to 0 is to write a 1 to the corresponding bit in the clear register. See Table 9−15 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Interrupt event															
Type	R	R	R	R	R	RSCU	RSCU	RSCU	RSCU	RSCU	RSCU	RSCU	RSCU	R	RSCU	RSCU
Default	0	X	0	0	0	X	X	X	X	X	X	X	X	0	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Interrupt event															
Type	R	R	R	R	R	R	RSCU	RSCU	RU	RU	RSCU	RSCU	RSCU	RSCU	RSCU	RSCU
Default	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X

Register: **Interrupt event**
Type: Read/Set/Clear/Update
Offset: 80h set register
84h clear register (returns IntEvent and IntMask when read**)**
Default: XXXX 0XXXh

Table 9−15. Interrupt Event Register

BIT	SIGNAL	TYPE	FUNCTION
31	RSVD	R	Reserved. Bit 31 returns 0 when read.
30	VENDORSPECIFIC	R	Vendor defined.
29−27	RSVD	R	Reserved. Bits 29−27 return 0s when read.
26	PHYREGRCVD	RSCU	The PCI4410 has received a PHY register data byte which can be read from the PHY control register.
25	CYCLETOOLONG	RSCU	If LinkControl.cycleMaster is set to 1, this indicates that over 125 µs elapsed between the start of sending a cycle start packet and the end of a subaction gap. LinkControl.cycleMaster is cleared by this event.
24	UNRECOVERABLEERROR	RSCU	This event occurs when the PCI4410 encounters any error that forces it to stop operations on any or all of its subunits, for example, when a DMA context sets its dead bit to 1. While UNRECOVERABLEERROR is set to 1, all normal interrupts for the context(s) that caused this interrupt are blocked from being set to 1.
23	CYCLEINCONSISTENT	RSCU	A cycle start was received that had an isochronous cycleTimer.seconds and isochronous cycleTimer.count different from the value in the CycleTimer register.
22	CYCLELOST	RSCU	A lost cycle is indicated when no cycle_start packet is sent/received between two successive cycleSynch events. A lost cycle can be predicted when a cycle_start packet does not immediately follow the first subaction gap after the cycleSynch event or if an arbitration reset gap is detected after a cycleSynch event without an intervening cycle start. CYCLELOST may be set to 1 either when a lost cycle occurs or when logic predicts that it will occur.
21	CYCLE64SECONDS	RSCU	Indicates that the 7th bit of the cycle second counter has changed.
20	CYCLESYNCH	RSCU	Indicates that a new isochronous cycle has started and is set to 1 when the low order bit of the cycle count toggles.
19	PHY	RSCU	Indicates the PHY requests an interrupt through a status transfer.
18	RSVD	R	Reserved. Bit 18 returns 0 when read.
17	BUSRESET	RSCU	Indicates that the PHY chip has entered bus reset mode.
16	SELFDCOMPLETE	RSCU	A self ID packet stream has been received. It is generated at the end of the bus initialization process. This bit is turned off simultaneously when IntEvent.busReset is turned on.
15−10	RSVD	R	Reserved. Bits 15−10 return 0s when read.

Table 9−15. Interrupt Event Register (Continued)

BIT	SIGNAL	TYPE	FUNCTION
9	LOCKRESPERR	RSCU	Indicates that the PCI4410 sent a lock response for a lock request to a serial bus register, but did not receive an ack_complete.
8	POSTEDWRITEERR	RSCU	Indicates that a host bus error occurred while the PCI4410 was trying to write a 1394 write request, which had already been given an ack_complete, into system memory.
7	ISOCHRX	RU	Isochronous receive DMA interrupt. Indicates that one or more isochronous receive contexts have generated an interrupt. This is not a latched event; it is the OR'ing of all bits in (isoRecvIntEvent and isoRecvIntMask). The isoRecvIntEvent register indicates which contexts have interrupted.
6	ISOCHTX	RU	Isochronous transmit DMA interrupt. Indicates that one or more isochronous transmit contexts have generated an interrupt. This is not a latched event, it is the OR'ing of all bits in (isoXmitIntEvent and isoXmitIntMask). The isoXmitIntEvent register indicates which contexts have interrupted.
5	RSPKT	RSCU	Indicates that a packet was sent to an asynchronous receive response context buffer and the descriptor's xferStatus and resCount fields have been updated.
4	RQPKT	RSCU	Indicates that a packet was sent to an asynchronous receive request context buffer and the descriptor's xferStatus and resCount fields have been updated.
3	ARRS	RSCU	Asynchronous receive response DMA interrupt. This bit is conditionally set to 1 upon completion of an ARRS DMA context command descriptor.
2	ARRQ	RSCU	Asynchronous receive request DMA interrupt. This bit is conditionally set to 1 upon completion of an ARRQ DMA context command descriptor.
1	RESPTXCOMPLETE	RSCU	Asynchronous response transmit DMA interrupt. This bit is conditionally set to 1 upon completion of an ATRS DMA command.
0	REQTXCOMPLETE	RSCU	Asynchronous request transmit DMA interrupt. This bit is conditionally set to 1 upon completion of an ATRQ DMA command.

9.22 Interrupt Mask Register

This set/clear register is used to enable the various PCI4410 interrupt sources. Reads from either the set register or the clear register always return IntMask. In all cases except masterIntEnable (bit 31), the enables for each interrupt event align with the event register bits detailed in Table 9−15. See Table 9−16 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Interrupt mask															
Type	RSC	R	R	R	R	RSCU	RSCU	RSCU	RSCU	RSCU	RSCU	RSCU	RSCU	R	RSCU	RSCU
Default	0	X	0	0	0	X	X	X	X	X	X	X	X	0	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Interrupt mask															
Type	R	R	R	R	R	R	RSCU	RSCU	RU	RU	RSCU	RSCU	RSCU	RSCU	RSCU	RSCU
Default	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X

Register: **Interrupt mask**
Type: Read/Set/Clear/Update, Read-only
Offset: 88h set register
8Ch clear register
Default: XXXX 0XXXh

Table 9−16. Interrupt Mask Register

BIT	SIGNAL	TYPE	FUNCTION
31	MASTERINTENABLE	RSC	When this bit is set to 1, external interrupts are generated in accordance with the IntMask register. If this bit is reset to 0, no external interrupts are generated.
30−0			See Table 9−15.

9.23 Isochronous Transmit Interrupt Event Register

This set/clear register reflects the interrupt state of the isochronous transmit contexts. An interrupt is generated on behalf of an isochronous transmit context if an OUTPUT_LAST command completes and its interrupt bits are set to 1. Upon determining that the IntEvent.isochTx interrupt has occurred, software can check this register to determine which context(s) caused the interrupt. The interrupt bits are set to 1 by an asserting edge of the corresponding interrupt signal, or by writing a 1 in the corresponding bit in the set register. The only mechanism to reset the bits in this register to 0 is to write a 1 to the corresponding bit in the clear register. See Table 9−17 for a complete description of the register contents.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	Isochronous transmit interrupt event															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Isochronous transmit interrupt event															
Type	R	R	R	R	R	R	R	R	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X

Register: **Isochronous transmit interrupt event**
Type: Read/Set/Clear**,** Read-only
Offset: 90h set register
84h clear register (returns IsoXmitEvent and IsoXmitMask when read**)**
Default: 0000 00XXh

Table 9−17. Isochronous Transmit Interrupt Event Register

BIT	SIGNAL	TYPE	FUNCTION
31−8	RSVD	R	Reserved. Bits 31−8 return 0s when read.
7	ISOXMIT7	RSC	Isochronous transmit channel 7 caused the isochTx interrupt.
6	ISOXMIT6	RSC	Isochronous transmit channel 6 caused the isochTx interrupt.
5	ISOXMIT5	RSC	Isochronous transmit channel 5 caused the isochTx interrupt.
4	ISOXMIT4	RSC	Isochronous transmit channel 4 caused the isochTx interrupt.
3	ISOXMIT3	RSC	Isochronous transmit channel 3 caused the isochTx interrupt.
2	ISOXMIT2	RSC	Isochronous transmit channel 2 caused the isochTx interrupt.
1	ISOXMIT1	RSC	Isochronous transmit channel 1 caused the isochTx interrupt.
0	ISOXMIT0	RSC	Isochronous transmit channel 0 caused the isochTx interrupt.

9.24 Isochronous Transmit Interrupt Mask Register

This set/clear register is used to enable the isochTx interrupt source on a per channel basis. Reads from either the set register or the clear register always return IsoXmitIntMask. In all cases the enables for each interrupt event align with the event register bits detailed in Table 9−17.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous transmit interrupt mask															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous transmit interrupt mask															
Type	R	R	R	R	R	R	R	R	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X

Register: **Isochronous transmit interrupt mask**
Type: Read/Set/Clear, Read-only
Offset: 98h set register
9Ch clear register
Default: 0000 00XXh

9.25 Isochronous Receive Interrupt Event Register

This set/clear register reflects the interrupt state of the isochronous receive contexts. An interrupt is generated on behalf of an isochronous receive context if an INPUT_* command completes and its interrupt bits are set to 1. Upon determining that the IntEvent.isochRx interrupt has occurred, software can check this register to determine which context(s) caused the interrupt. The interrupt bits are set to 1 by an asserting edge of the corresponding interrupt signal, or by writing a 1 in the corresponding bit in the set register. The only mechanism to reset the bits in this register to 0 is to write a 1 to the corresponding bit in the clear register. See Table 9−18 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous receive interrupt event															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous receive interrupt event															
Type	R	R	R	R	R	R	R	R	R	R	R	R	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X

Register: **Isochronous receive interrupt event**
Type: Read/Set/Clear, Read-only
Offset: A0h set register
A4h clear register (returns IsoRecvEvent and IsoRecvMask when read**)**
Default: 0000 000Xh

Table 9−18. Isochronous Receive Interrupt Event Register

BIT	SIGNAL	TYPE	FUNCTION
31−4	RSVD	R	Reserved. These bits return 0s when read.
3	ISORECV3	RSC	Isochronous receive channel 3 caused the isochRx interrupt.
2	ISORECV2	RSC	Isochronous receive channel 2 caused the isochRx interrupt.
1	ISORECV1	RSC	Isochronous receive channel 1 caused the isochRx interrupt.
0	ISORECV0	RSC	Isochronous receive channel 0 caused the isochRx interrupt.

9.26 Isochronous Receive Interrupt Mask Register

This set/clear register is used to enable the isochRx interrupt source on a per channel basis. Reads from either the set register or the clear register always return IsoRecvIntMask. In all cases the enables for each interrupt event align with the event register bits detailed in Table 9−18.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous receive interrupt mask															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous receive interrupt mask															
Type	R	R	R	R	R	R	R	R	R	R	R	R	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X

Register: **Isochronous receive interrupt mask**
Type: Read/Set/Clear, Read-only
Offset: A8h set register
ACh clear register
Default: 0000 000Xh

9.27 Fairness Control Register (Optional Register)

This register provides a mechanism by which software can direct the host controller to transmit multiple asynchronous requests during a fairness interval. See Table 9−19 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Fairness control															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Fairness control															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0

Register: **Fairness control**
Type: Read-only
Offset: DCh
Default: XXXX XX00h

Table 9−19. Fairness Control Register

BIT	SIGNAL	TYPE	FUNCTION
31−8	RSVD	R	Reserved.
7−0	PRI_REQ	R	This field specifies the maximum number of priority arbitration requests for asynchronous request packets that the link is permitted to make of the PHY during fairness interval.

9.28 Link Control Register

This set/clear register provides the control flags that enable and configure the link core protocol portions of the PCI4410. It contains controls for the receiver and cycle timer. See Table 9−20 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Link control															
Type	R	R	R	R	R	R	R	R	R	RSC	RSCU	RSC	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	X	X	X	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Link control															
Type	R	R	R	R	R	RSC	RSC	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	X	X	0	0	0	0	0	0	0	0	0

Register: **Link control**
Type: Read/Set/Clear/Update, Read/Set/Clear, Read-only
Offset: E0h set register
E4h clear register
Default: 00X0 0X00h

Table 9−20. Link Control Register

BIT	SIGNAL	TYPE	FUNCTION
31−23	RSVD	R	Reserved. Bits 31−23 return 0s when read.
22	CYCLESOURCE	RSC	When this bit is 1, the cycle timer uses an external source (CYCLEIN) to determine when to roll over the cycle timer. When this bit is 0, the cycle timer rolls over when the timer reaches 3072 cycles of the 24.576-MHz clock (125 μs).
21	CYCLEMASTER	RSCU	When this bit is set to 1 and the PHY has notified the PCI4410 that it is root, the PCI4410 generates a cycle start packet every time the cycle timer rolls over, based on the setting of bit 22 (CYCLESOURCE). When this bit is 0, the OHCILynx accepts received cycle start packets to maintain synchronization with the node which is sending them. This bit is automatically reset to 0 when the cycleTooLong event occurs and cannot be set to 1 until the IntEvent.cycleTooLong bit is cleared.
20	CYCLETIMERENABLE	RSC	When this bit is 1, the cycle timer offset counts cycles of the 24.576-MHz clock and rolls over at the appropriate time based on the settings of the above bits. When this bit is 0, the cycle timer offset does not count.
19−11	RSVD	R	Reserved. Bits 19−11 return 0s when read.
10	RCVPHYPKT	RSC	When this bit is 1, the receiver accepts incoming PHY packets into the AR request context if the AR request context is enabled. This does not control receipt of self-identification packets.
9	RCVSELFID	RSC	When this bit is 1, the receiver accepts incoming self-identification packets. Before setting this bit to 1, software must ensure that the self ID buffer pointer register contains a valid address.
8−0	RSVD	R	Reserved. Bits 8−0 return 0s when read.

9.29 Node Identification Register

This register contains the address of the node on which the OHCILynx chip resides, and indicates the valid node number status. The 16-bit combination of busNumber and NodeNumber is referred to as the node ID. See Table 9−21 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Node identification															
Type	RU	RU	R	R	RU	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Node identification															
Type	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RU	RU	RU	RU	RU	RU
Default	1	1	1	1	1	1	1	1	1	1	X	X	X	X	X	X

Register: **Node identification**
Type: Read/Write/Update, Read/Update, Read-only
Offset: E8h
Default: 0000 11XXh

Table 9−21. Node Identification Register

BIT	SIGNAL	TYPE	FUNCTION
31	IDVALID	RU	This bit indicates whether or not the PCI4410 has a valid node number. It is reset to 0 when a 1394 bus reset is detected and set to 1 when the PCI4410 receives a new node number from the PHY.
30	ROOT	RU	This bit is set to 1 during the bus reset process if the attached PHY is root.
29−28	RSVD	R	Reserved. Bits 29 and 28 return 0s when read.
27	CPS	RU	This bit is set to 1 if the PHY is reporting that cable power status is OK (VP 8V).
26−16	RSVD	R	Reserved. Bits 26−16 return 0s when read.
15−6	BUSNUMBER	RWU	This number is used to identify the specific 1394 bus the PCI4410 belongs to when multiple 1394-compatible buses are connected via a bridge.
5−0	NODENUMBER	RU	This number is the physical node number established by the PHY during self-identification. It is automatically set to the value received from the PHY after the self-identification phase. If the PHY sets the nodeNumber to 63, then software should not set ContextControl.run for either of the AT DMA contexts.

9.30 PHY Control Register

This register is used to read or write a PHY register. See Table 9−22 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PHY control															
Type	RU	R	R	R	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU	RU
Default	X	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PHY control															
Type	RWU	RWU	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X

Register: **PHY control**
Type: Read/Write/Update, Read/Update, Read-only
Offset: ECh
Default: XXXX 0XXXh

Table 9−22. PHY Control Register

BIT	SIGNAL	TYPE	FUNCTION
31	RDDONE	RU	This bit is cleared to 0 by the PCI4410 when either bit 15 (rdReg) or bit 14 (wrReg) is set to 1. This bit is set to 1 when a register transfer is received from the PHY.
30−28	RSVD	R	Reserved. Bits 30−28 return 0s when read.
27−24	RDADDR	RU	This is the address of the register most recently received from the PHY.
23−16	RDDATA	RU	This field is the contents of a PHY register which has been read.
15	RDREG	RWU	This bit is set to 1 by software to initiate a read request to a PHY register, and is cleared by hardware when the request has been sent. The wrReg and rdReg bits must be used exclusively.
14	WRREG	RWU	This bit is set to 1 by software to initiate a write request to a PHY register, and is reset to 0 by hardware when the request has been sent. The wrReg and rdReg bits must be used exclusively.
13−12	RSVD	R	Reserved. Bits 13 and 12 return 0 when read.
11−8	REGADDR	R/W	This field is the address of the PHY register to be written or read.
7−0	WRDATA	R/W	This field is the data to be written to a PHY register, and is ignored for reads.

9.31 Isochronous Cycle Timer Register

This read/write register indicates the current cycle number and offset. When the PCI4410 is cycle master, this register is transmitted with the cycle start message. When the PCI4410 is not cycle master, this register is loaded with the data field in an incoming cycle start. In the event that the cycle start message is not received, the fields can continue incrementing on their own (if programmed) to maintain a local time reference. See Table 9−23 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous cycle timer															
Type	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous cycle timer															
Type	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **Isochronous cycle timer**
Type: Read/Write/Update
Offset: F0h
Default: XXXX XXXXh

Table 9−23. Isochronous Cycle Timer Register

BIT	SIGNAL	TYPE	FUNCTION
31−25	CYCLESECONDS	RWU	This field counts seconds (cycleCount rollovers) modulo 128.
24−12	CYCLECOUNT	RWU	This field counts cycles (cycleOffset rollovers) modulo 8000.
11−0	CYCLEOFFSET	RWU	This field counts 24.576-MHz clocks modulo 3072, that is, 125 μs. If an external 8-kHz clock configuration is being used, then CYCLEOFFSET must be reset to 0 at each tick of the external clock.

9.32 Asynchronous Request Filter High Register

This set/clear register is used to enable asynchronous receive requests on a per node basis, and handles the upper node IDs. When a packet is destined for either the physical request context or the ARRQ context, the source node ID is examined. If the bit corresponding to the node ID is not set to 1 in this register, then the packet is not acknowledged and the request is not queued. The node ID comparison is done if the source node is on the same bus as the PCI4410. All nonlocal bus-sourced packets are not acknowledged unless bit 31 in this register is set to 1. See Table 9−24 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Asynchronous request filter high															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Asynchronous request filter high															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Asynchronous request filter high**
Type: Read/Set/Clear
Offset: 100h set register
104h clear register
Default: 0000 0000h

Table 9−24. Asynchronous Request Filter High Register

BIT	SIGNAL	TYPE	FUNCTION
31	ASYNREQALLBUSES	RSC	If set to 1, all asynchronous requests received by the PCI4410 from nonlocal bus nodes are accepted.
30	ASYNREQRESOURCE62	RSC	If set to 1 for local bus node number 62, asynchronous requests received by the PCI4410 from that node are accepted.
29	ASYNREQRESOURCE61	RSC	If set to 1 for local bus node number 61, asynchronous requests received by the PCI4410 from that node are accepted.
28	ASYNREQRESOURCE60	RSC	If set to 1 for local bus node number 60, asynchronous requests received by the PCI4410 from that node are accepted.
27	ASYNREQRESOURCE59	RSC	If set to 1 for local bus node number 59, asynchronous requests received by the PCI4410 from that node are accepted.
26	ASYNREQRESOURCE58	RSC	If set to 1 for local bus node number 58, asynchronous requests received by the PCI4410 from that node are accepted.
25	ASYNREQRESOURCE57	RSC	If set to 1 for local bus node number 57, asynchronous requests received by the PCI4410 from that node are accepted.
24	ASYNREQRESOURCE56	RSC	If set to 1 for local bus node number 56, asynchronous requests received by the PCI4410 from that node are accepted.
23	ASYNREQRESOURCE55	RSC	If set to 1 for local bus node number 55, asynchronous requests received by the PCI4410 from that node are accepted.
22	ASYNREQRESOURCE54	RSC	If set to 1 for local bus node number 54, asynchronous requests received by the PCI4410 from that node are accepted.
21	ASYNREQRESOURCE53	RSC	If set to 1 for local bus node number 53, asynchronous requests received by the PCI4410 from that node are accepted.
20	ASYNREQRESOURCE52	RSC	If set to 1 for local bus node number 52, asynchronous requests received by the PCI4410 from that node are accepted.

Table 9−24. Asynchronous Request Filter High Register (Continued)

BIT	SIGNAL	TYPE	FUNCTION
19	ASYNREQRESOURCE51	RSC	If set to 1 for local bus node number 51, asynchronous requests received by the PCI4410 from that node are accepted.
18	ASYNREQRESOURCE50	RSC	If set to 1 for local bus node number 50, asynchronous requests received by the PCI4410 from that node are accepted.
17	ASYNREQRESOURCE49	RSC	If set to 1 for local bus node number 49, asynchronous requests received by the PCI4410 from that node are accepted.
16	ASYNREQRESOURCE48	RSC	If set to 1 for local bus node number 48, asynchronous requests received by the PCI4410 from that node are accepted.
15	ASYNREQRESOURCE47	RSC	If set to 1 for local bus node number 47, asynchronous requests received by the PCI4410 from that node are accepted.
14	ASYNREQRESOURCE46	RSC	If set to 1 for local bus node number 46, asynchronous requests received by the PCI4410 from that node are accepted.
13	ASYNREQRESOURCE45	RSC	If set to 1 for local bus node number 45, asynchronous requests received by the PCI4410 from that node are accepted.
12	ASYNREQRESOURCE44	RSC	If set to 1 for local bus node number 44, asynchronous requests received by the PCI4410 from that node are accepted.
11	ASYNREQRESOURCE43	RSC	If set to 1 for local bus node number 43, asynchronous requests received by the PCI4410 from that node are accepted.
10	ASYNREQRESOURCE42	RSC	If set to 1 for local bus node number 42, asynchronous requests received by the PCI4410 from that node are accepted.
9	ASYNREQRESOURCE41	RSC	If set to 1 for local bus node number 41, asynchronous requests received by the PCI4410 from that node are accepted.
8	ASYNREQRESOURCE40	RSC	If set to 1 for local bus node number 40, asynchronous requests received by the PCI4410 from that node are accepted.
7	ASYNREQRESOURCE39	RSC	If set to 1 for local bus node number 39, asynchronous requests received by the PCI4410 from that node are accepted.
6	ASYNREQRESOURCE38	RSC	If set to 1 for local bus node number 38, asynchronous requests received by the PCI4410 from that node are accepted.
5	ASYNREQRESOURCE37	RSC	If set to 1 for local bus node number 37, asynchronous requests received by the PCI4410 from that node are accepted.
4	ASYNREQRESOURCE36	RSC	If set to 1 for local bus node number 36, asynchronous requests received by the PCI4410 from that node are accepted.
3	ASYNREQRESOURCE35	RSC	If set to 1 for local bus node number 35, asynchronous requests received by the PCI4410 from that node are accepted.
2	ASYNREQRESOURCE34	RSC	If set to 1 for local bus node number 34, asynchronous requests received by the PCI4410 from that node are accepted.
1	ASYNREQRESOURCE33	RSC	If set to 1 for local bus node number 33, asynchronous requests received by the PCI4410 from that node are accepted.
0	ASYNREQRESOURCE32	RSC	If set to 1 for local bus node number 32, asynchronous requests received by the PCI4410 from that node are accepted.

9.33 Asynchronous Request Filter Low Register

This set/clear register is used to enable asynchronous receive requests on a per node basis, and handles the lower node IDs. Other than filtering different node IDs, this register behaves identically to the asynchronous request filter high register. See Table 9−25 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Asynchronous request filter low															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Asynchronous request filter low															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Asynchronous request filter low**
Type: Read/Set/Clear
Offset: 108h set register
10Ch clear register
Default: 0000 0000h

Table 9−25. Asynchronous Request Filter Low Register

BIT	SIGNAL	TYPE	FUNCTION
31	ASYNREQRESOURCE31	RSC	If set to 1 for local bus node number 31, asynchronous requests received by the PCI4410 from that node are accepted.
30	ASYNREQRESOURCE30	RSC	If set to 1 for local bus node number 30, asynchronous requests received by the PCI4410 from that node are accepted.
⋮	⋮	⋮	Bits 29 through 2 follow the same pattern.
1	ASYNREQRESOURCE1	RSC	If set to 1 for local bus node number 1, asynchronous requests received by the PCI4410 from that node are accepted.
0	ASYNREQRESOURCE0	RSC	If set to 1 for local bus node number 0, asynchronous requests received by the PCI4410 from that node are accepted.

9.34 Physical Request Filter High Register

This set/clear register is used to enable physical receive requests on a per node basis, and handles the upper node IDs. When a packet is destined for the physical request context, and the node ID has been compared against the ARRQ registers, then the comparison is done again with this register. If the bit corresponding to the node ID is not set to 1 in this register, then the request is handled by the ARRQ context instead of the physical request context. See Table 9−26 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Physical request filter high															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Physical request filter high															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Physical request filter high**
Type: Read/Set/Clear
Offset: 110h set register
114h clear register
Default: 0000 0000h

Table 9−26. Physical Request Filter High Register

BIT	SIGNAL	TYPE	FUNCTION
31	PHYSREQALLBUSSES	RSC	If set to 1, then all asynchronous requests received by the PCI4410 from nonlocal bus nodes are accepted.
30	PHYSREQRESOURCE62	RSC	If set to 1 for local bus node number 62, then physical requests received by the PCI4410 from that node are handled through the physical request context.
29	PHYSREQRESOURCE61	RSC	If set to 1 for local bus node number 61, then physical requests received by the PCI4410 from that node are handled through the physical request context.
28	PHYSREQRESOURCE60	RSC	If set to 1 for local bus node number 60, then physical requests received by the PCI4410 from that node are handled through the physical request context.
27	PHYSREQRESOURCE59	RSC	If set to 1 for local bus node number 59, then physical requests received by the PCI4410 from that node are handled through the physical request context.
26	PHYSREQRESOURCE58	RSC	If set to 1 for local bus node number 58, then physical requests received by the PCI4410 from that node are handled through the physical request context.
25	PHYSREQRESOURCE57	RSC	If set to 1 for local bus node number 57, then physical requests received by the PCI4410 from that node are handled through the physical request context.
24	PHYSREQRESOURCE56	RSC	If set to 1 for local bus node number 56, then physical requests received by the PCI4410 from that node are handled through the physical request context.
23	PHYSREQRESOURCE55	RSC	If set to 1 for local bus node number 55, then physical requests received by the PCI4410 from that node are handled through the physical request context.
22	PHYSREQRESOURCE54	RSC	If set to 1 for local bus node number 54, then physical requests received by the PCI4410 from that node are handled through the physical request context.
21	PHYSREQRESOURCE53	RSC	If set to 1 for local bus node number 53, then physical requests received by the PCI4410 from that node are handled through the physical request context.
20	PHYSREQRESOURCE52	RSC	If set to 1 for local bus node number 52, then physical requests received by the PCI4410 from that node are handled through the physical request context.
19	PHYSREQRESOURCE51	RSC	If set to 1 for local bus node number 51, then physical requests received by the PCI4410 from that node are handled through the physical request context.
18	PHYSREQRESOURCE50	RSC	If set to 1 for local bus node number 50, then physical requests received by the PCI4410 from that node are handled through the physical request context.

Table 9−26. Physical Request Filter High Register (Continued)

BIT	SIGNAL	TYPE	FUNCTION
17	PHYSREQRESOURCE49	RSC	If set to 1 for local bus node number 49, then physical requests received by the PCI4410 from that node are handled through the physical request context.
16	PHYSREQRESOURCE48	RSC	If set to 1 for local bus node number 48, then physical requests received by the PCI4410 from that node are handled through the physical request context.
15	PHYSREQRESOURCE47	RSC	If set to 1 for local bus node number 47, then physical requests received by the PCI4410 from that node are handled through the physical request context.
14	PHYSREQRESOURCE46	RSC	If set to 1 for local bus node number 46, then physical requests received by the PCI4410 from that node are handled through the physical request context.
13	PHYSREQRESOURCE45	RSC	If set to 1 for local bus node number 45, then physical requests received by the PCI4410 from that node are handled through the physical request context.
12	PHYSREQRESOURCE44	RSC	If set to 1 for local bus node number 44, then physical requests received by the PCI4410 from that node are handled through the physical request context.
11	PHYSREQRESOURCE43	RSC	If set to 1 for local bus node number 43, then physical requests received by the PCI4410 from that node are handled through the physical request context.
10	PHYSREQRESOURCE42	RSC	If set to 1 for local bus node number 42, then physical requests received by the PCI4410 from that node are handled through the physical request context.
9	PHYSREQRESOURCE41	RSC	If set to 1 for local bus node number 41, then physical requests received by the PCI4410 from that node are handled through the physical request context.
8	PHYSREQRESOURCE40	RSC	If set to 1 for local bus node number 40, then physical requests received by the PCI4410 from that node are handled through the physical request context.
7	PHYSREQRESOURCE39	RSC	If set to 1 for local bus node number 39, then physical requests received by the PCI4410 from that node are handled through the physical request context.
6	PHYSREQRESOURCE38	RSC	If set to 1 for local bus node number 38, then physical requests received by the PCI4410 from that node are handled through the physical request context.
5	PHYSREQRESOURCE37	RSC	If set to 1 for local bus node number 37, then physical requests received by the PCI4410 from that node are handled through the physical request context.
4	PHYSREQRESOURCE36	RSC	If set to 1 for local bus node number 36, then physical requests received by the PCI4410 from that node are handled through the physical request context.
3	PHYSREQRESOURCE35	RSC	If set to 1 for local bus node number 35, then physical requests received by the PCI4410 from that node are handled through the physical request context.
2	PHYSREQRESOURCE34	RSC	If set to 1 for local bus node number 34, then physical requests received by the PCI4410 from that node are handled through the physical request context.
1	PHYSREQRESOURCE33	RSC	If set to 1 for local bus node number 33, then physical requests received by the PCI4410 from that node are handled through the physical request context.
0	PHYSREQRESOURCE32	RSC	If set to 1 for local bus node number 32, then physical requests received by the PCI4410 from that node are handled through the physical request context.

9.35 Physical Request Filter Low Register

This set/clear register is used to enable physical receive requests on a per node basis, and handles the lower node IDs. When a packet is destined for the physical request context, and the node ID has been compared against the asynchronous request filter registers, then the node ID comparison is done again with this register. If the bit corresponding to the node ID is not set to 1 in this register, then the request is handled by the asynchronous request context instead of the physical request context. See Table 9−27 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Physical request filter low															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Physical request filter low															
Type	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Physical request filter low**
Type: Read/Set/Clear
Offset: 118h set register
11Ch clear register
Default: 0000 0000h

Table 9−27. Physical Request Filter Low Register

BIT	SIGNAL	TYPE	FUNCTION
31	PHYSREQRESOURCE31	RSC	If set to 1 for local bus node number 31, then physical requests received by the PCI4410 from that node are handled through the physical request context.
30	PHYSREQRESOURCE30	RSC	If set to 1 for local bus node number 30, then physical requests received by the PCI4410 from that node are handled through the physical request context.
⋮	⋮	⋮	Bits 29 through 2 follow the same pattern.
1	PHYSREQRESOURCE1	RSC	If set to 1 for local bus node number 1, then physical requests received by the PCI4410 from that node are handled through the physical request context.
0	PHYSREQRESOURCE0	RSC	If set to 1 for local bus node number 0, then physical requests received by the PCI4410 from that node are handled through the physical request context.

9.36 Physical Upper Bound Register (Optional Register)

This register is an optional register and is not implemented. This register is read-only and returns all 0s.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Physical upper bound															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Physical upper bound															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Physical upper bound**
Type: Read-only
Offset: 120h
Default: 0000 0000h

9.37 Asynchronous Context Control Register

This set/clear register controls the state and indicates status of the DMA context. See Table 9−28 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Asynchronous context control															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Asynchronous context control															
Type	RSCU	R	R	RSU	RU	RU	R	R	RU	RU	RU	RU	RU	RU	RU	RU
Default	0	0	0	X	0	0	0	0	X	X	X	X	X	X	X	X

Register: **Asynchronous context control**
Type: Read/Set/Clear/Update, Read/Set/Update, Read/Update, Read-only
Offset:
180h set register [ATRQ]
184h clear register [ATRQ]
1A0h set register [ATRS]
1A4h clear register [ATRS]
1C0h set register [ARRQ]
1C4h clear register [ARRQ]
1E0h set register [ATRS]
1E4h clear register [ATRS]
Default: 0000 X0XXh

Table 9−28. Asynchronous Context Control Register

BIT	SIGNAL	TYPE	FUNCTION
31−16	RSVD	R	Reserved. Bits 31−16 return 0s when read.
15	RUN	RSCU	This bit is set to 1 by software to enable descriptor processing for the context and cleared by software to stop descriptor processing. The PCI4410 will only change this bit on a hardware or software reset.
14−13	RSVD	R	Reserved. Bits 14 and 13 return 0s when read.
12	WAKE	RSU	Software sets this bit to 1 to cause the PCI4410 to continue or resume descriptor processing. The PCI4410 will reset this bit to 0 on every descriptor fetch.
11	DEAD	RU	The PCI4410 sets this bit to 1 when it encounters a fatal error and resets the bit to 0 when software resets the RUN bit to 0.
10	ACTIVE	RU	The PCI4410 sets this bit to 1 when it is processing descriptors.
9−8	RSVD	R	Reserved. Bits 9 and 8 return 0s when read.
7−5	SPD	RU	This field indicates the speed at which a packet was received or transmitted, and only contains meaningful information for receive contexts. This field is encoded as: 000b = 100 Mbits/sec 001b = 200 Mbits/sec 010b = 400 Mbits/sec. All other values are reserved.
4−0	EVENTCODE	RU	This field holds the acknowledge sent by the link core for this packet or an internally generated error code if the packet was not transferred successfully.

9.38 Asynchronous Context Command Pointer Register

This register contains a pointer to the address of the first descriptor block that the PCI4410 will access when software enables the context by setting the ContextControl.run bit to 1. See Table 9−29 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Asynchronous context command pointer															
Type	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Asynchronous context command pointer															
Type	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU	RWU
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **Asynchronous context command pointer**
Type: Read/Write/Update
Offset: 19Ch [ATRQ]
1ACh [ATRS]
1CCh [ATRQ]
1ECh [ATRS]
Default: XXXX XXXXh

Table 9−29. Asynchronous Context Command Pointer Register

BIT	SIGNAL	TYPE	FUNCTION
31−4	DESCRIPTORADDRESS	RWU	Contains the upper 28 bits of the address of a 16-byte aligned descriptor block.
3−0	Z	RWU	Indicates the number of contiguous descriptors at the address pointed to by the descriptor address. If Z is 0, it indicates that the descriptorAddress is not valid.

9.39 Isochronous Transmit Context Control Register

This set/clear register controls options, state, and status for the isochronous transmit DMA contexts. The n value in the following register addresses indicates the context number (n = 0, 1, 2, 3, …). See Table 9−30 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous transmit context control															
Type	RSCU	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC	RSC
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous transmit context control															
Type	RSC	R	R	RSU	RU	RU	R	R	RU	RU	RU	RU	RU	RU	RU	RU
Default	0	0	0	X	0	0	0	0	X	X	X	X	X	X	X	X

Register: **Isochronous transmit context control**
Type: Read/Set/Clear/Update, Read/Set/Clear, Read-only, Read/Update
Offset: 200h + (16 * n) set register
204h + (16 * n) clear register
Default: XXXX X0XXh

Table 9−30. Isochronous Transmit Context Control Register

BIT	SIGNAL	TYPE	FUNCTION
31	CYCLEMATCHENABLE	RSCU	When set to 1, processing will occur such that the packet described by the first descriptor block of the context is transmitted in the cycle whose number is specified in the CYCLEMATCH field of this register. The 13-bit CYCLEMATCH field must match the 13-bit cycleCount field in the cycle start packet that is sent or received immediately before isochronous transmission begins.
30−16	CYCLEMATCH	RSC	Contains a 15-bit value, corresponding to the lower order 2 bits of cycleSeconds and 13-bit cycleCount field. If CYCLEMATCHENABLE is set to 1, then this IT DMA context becomes enabled for transmits when the bus cycleCount value equals the CYCLEMATCH value.
15	RUN	RSC	This bit is set to 1 by software to enable descriptor processing for the context and cleared by software to stop descriptor processing. The PCI4410 only changes this bit on a hardware or software reset.
14−13	RSVD	R	Reserved. Bits 14 and 13 return 0s when read.
12	WAKE	RSU	Software sets this bit to 1 to cause the PCI4410 to continue or resume descriptor processing. The PCI4410 resets this bit to 0 on every descriptor fetch.
11	DEAD	RU	The PCI4410 sets this bit to 1 when it encounters a fatal error, and resets the bit to 0 when software resets the RUN bit to 0.
10	ACTIVE	RU	The PCI4410 sets this bit to 1 when it is processing descriptors.
9−8	RSVD	R	Reserved. Bits 9 and 8 return 0 when read.
7−5	SPD	RU	This field is not meaningful for isochronous transmit contexts.
4−0	EVENT CODE	RU	Following an OUTPUT_LAST* command, the error code is indicated in this field. Possible values are: ack_complete, evt_descriptor_read, evt_data_read, and evt_unknown.

9.40 Isochronous Transmit Context Command Pointer Register

This register contains a pointer to the address of the first descriptor block that the PCI4410 will access when software enables an ISO transmit context by setting the ContextControl.run bit to 1. The n value in the following register addresses indicates the context number (n = 0, 1, 2, 3, …).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous transmit context command pointer															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous transmit context command pointer															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **Isochronous transmit context command pointer**
Type: Read-only
Offset: 20Ch + (16 * n)
Default: XXXX XXXh

9.41 Isochronous Receive Context Control Register

This set/clear register controls options, state, and status for the isochronous receive DMA contexts. The n value in the following register addresses indicates the context number (n = 0, 1, 2, 3, …). See Table 9−31 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous receive context control															
Type	RSC	RSC	RSCU	RSC	R	R	R	R	R	R	R	R	R	R	R	R
Default	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous receive context control															
Type	RSCU	R	R	RSU	RU	RU	R	R	RU	RU	RU	RU	RU	RU	RU	RU
Default	0	0	0	X	0	0	0	0	X	X	X	X	X	X	X	X

Register: **Isochronous receive context control**
Type: Read/Set/Clear/Update, Read/Set/Clear, Read/Update, Read-only
Offset: 400h + (32 * n) set register
404h + (32 * n) clear register
Default: X000 X0XXh

Table 9−31. Isochronous Receive Context Control Register

BIT	SIGNAL	TYPE	FUNCTION
31	BUFFERFILL	RSC	When set to 1, received packets are placed back to back to completely fill each receive buffer. When reset to 0, each received packet is placed in a single buffer. If the MULTICHANMODE bit is set to 1, this bit must also be set to 1. The value of BUFFERFILL must not be changed while ACTIVE or RUN is set to 1.
30	ISOCHHEADER	RSC	When set to 1, received isochronous packets will include the complete 4-byte isochronous packet header seen by the link layer. The end of the packet is marked with xferStatus in the first doublet, and a 16-bit timeStamp indicating the time of the most recently received (or sent) cycleStart packet. When clear, the packet header is stripped from received isochronous packets. The packet header, if received, immediately precedes the packet payload. The value of isochHeader must not be changed while ACTIVE or RUN is set to 1.
29	CYCLEMATCHENABLE	RSCU	When set to 1, the context begins running only when the 13-bit cycleMatch field in the contextMatch register matches the 13-bit cycleCount in the cycleStart packet. The effects of this bit, however, are impacted by the values of other bits in this register. Once the context has become active, hardware clears the CYCLEMATCHENABLE bit. The value of CYCLEMATCHENABLE must not be changed while ACTIVE or RUN is set to 1.
28	MULTICHANMODE	RSC	When set to 1, the corresponding isochronous receive DMA context receives packets for all isochronous channels enabled in the IRChannelMaskHi and IRChannelMaskLo registers. The isochronous channel number specified in the IRDMA context match register is ignored. When 0, the IRDMA context receives packets for that single channel. Only one IRDMA context may use the IRChannelMask registers. If more that one IRDMA context control register has the multiChanMode bit set to 1, then results are undefined. The value of MULTICHANMODE must not be changed while ACTIVE or RUN is set to 1.
27−16	RSVD	R	Reserved. Bits 27−16 return 0s when read.
15	RUN	RSCU	This bit is set by software to enable descriptor processing for the context and cleared by software to stop descriptor processing. The PCI4410 only changes this bit on a hardware or software reset.
14−13	RSVD	R	Reserved. Bits 14 and 13 return 0s when read.
12	WAKE	RSU	Software sets this bit to cause the PCI4410 to continue or resume descriptor processing. The PCI4410 clears this bit on every descriptor fetch.
11	DEAD	RU	The PCI4410 sets this bit to 1 when it encounters a fatal error, and resets the bit to 0 when software resets the RUN bit to 0.
10	ACTIVE	RU	The PCI4410 sets this bit to 1 when it is processing descriptors.

Table 9−31. Isochronous Receive Context Control Register (Continued)

BIT	SIGNAL	TYPE	FUNCTION
9−8	RSVD	R	Reserved. Bits 9 and 8 return 0 when read.
7−5	SPD	RU	This field indicates the speed at which the packet was received. 000b = 100 Mbits/sec 001b = 200 Mbits/sec, 010b = 400 Mbits/sec. All other values are reserved.
4−0	EVENT CODE	RU	Following an INPUT* command, the error code is indicated in this field.

9.42 Isochronous Receive Context Command Pointer Register

This register contains a pointer to the address of the first descriptor block that the PCI4410 will access when software enables an ISO receive context by setting the ContextControl.run bit. The n value in the following register addresses indicates the context number (n = 0, 1, 2, 3, …).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous receive context command pointer															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name																
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register: **Isochronous receive context command pointer**
Type: Read-only
Offset: 40Ch + (32 * n)
Default: XXXX XXXXh

9.43 Isochronous Receive Context Match Register

This register is used to start an isochronous receive context running on a specified cycle number, to filter incoming isochronous packets based on tag values, and to wait for packets with a specified sync value. The n value in the following register addresses indicates the context number (n = 0, 1, 2, 3, …). See Table 9−32 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous receive context match															
Type	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	X	X	X	X	0	0	0	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	X	X	X	X	X	X	X	X	0	X	X	X	X	X	X	X

Register: **Isochronous receive context match**
Type: Read/Write, Read-only
Offset: 410Ch + (32 * n)
Default: XXXX XXXXh

Table 9−32. Isochronous Receive Context Match Register

BIT	SIGNAL	TYPE	FUNCTION
31	TAG3	R/W	If this bit is set, then this context will match on ISO receive packets with a tag field of 11b.
30	TAG2	R/W	If this bit is set, then this context will match on ISO receive packets with a tag field of 10b.
29	TAG1	R/W	If this bit is set, then this context will match on ISO receive packets with a tag field of 01b.
28	TAG0	R/W	If this bit is set, then this context will match on ISO receive packets with a tag field of 00b.
27−25	RSVD	R	Reserved. Bits 27−25 return 0s when read.
24−12	CYCLEMATCH	R/W	Contains a 13-bit value, corresponding to the 13-bit cycleCount field in the cycleStart packet. If cycleMatchEnable is set, then this context is enabled for receives when the bus cycleCount value equals the cycleMatch value.
11−8	SYNC	R/W	This 4-bit field is compared to the sync field of each iso packet for this channel when the command descriptor's w field is set to 11b.
7	RSVD	R	Reserved. Bit 7 returns 0 when read.
6	TAG1SYNCFILTER	R/W	If this bit and bit 29 (TAG1) are set , then packets with tag 01b are accepted into the context if the two most significant bits of the packets sync field are 00b. Packets with tag values other than 01b are filtered according to TAG0, TAG2 and TAG3 without any additional restrictions. If clear, this context will match on isochronous receive packets as specified in the TAG0−3 bits with no additional restrictions.
5−0	CHANNELNUMBER	R/W	This 6-bit field indicates the isochronous channel number for which this IR DMA context accepts packets.

10 Electrical Characteristics

10.1 Absolute Maximum Ratings Over Operating Temperature Ranges†

Supply voltage range, V_{CC} . . . . . −0.5 V to 4.6 V
Clamping voltage range, V_{CCCB}, V_{CCI}, V_{CCL} V_{CCP}, ... −0.5 V to 6 V
Input voltage range, V_I: PCI . . . . . −0.5 V to V_{CCP} + 0.5 V
Card A . . . . . −0.5 V to V_{CCA} + 0.5 V
ZV . . . . . −0.5 V to V_{CC} + 0.5 V
TTL . . . . . −0.5 V to V_{CC} + 0.5 V
Fail safe . . . . . −0.5 V to V_{CC} + 0.5 V
Miscellaneous and PHY I/F . . . . . −0.5 V to V_{CC} + 0.5 V
Output voltage range, V_O: PCI . . . . . −0.5 V to V_{CC} + 0.5 V
Card A . . . . . −0.5 V to V_{CCA} + 0.5 V
ZV . . . . . −0.5 V to V_{CC} + 0.5 V
TTL . . . . . −0.5 V to V_{CC} + 0.5 V
Fail safe . . . . . −0.5 V to V_{CC} + 0.5 V
Miscellaneous and PHY I/F . . . . . −0.5 V to V_{CC} + 0.5 V
Input clamp current, I_{IK} (V_I < 0 or V_I > V_{CC}) (see Note 1) . . . . . ±20 mA
Output clamp current, I_{OK} (V_O < 0 or V_O > V_{CC}) (see Note 2) . . . . . ±20 mA
Storage temperature range, T_{stg} . . . . . −65°C to 150°C
Virtual junction temperature, T_J . . . . . 150°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. Applies for external input and bidirectional buffers. $V_I > V_{CC}$ does not apply to fail-safe terminals. PCI terminals are measured with respect to V_{CCP} instead of V_{CC}. PC Card terminals are measured with respect to V_{CCCB}. Miscellaneous signals are measured with respect to V_{CCI}. The limit specified applies for a dc condition.

2. Applies for external output and bidirectional buffers. $V_O > V_{CC}$ does not apply to fail-safe terminals. PCI terminals are measured with respect to V_{CCP} instead of V_{CC}. PC Card terminals are measured with respect to V_{CCCB}. Miscellaneous signals are measured with respect to V_{CCI}. The limit specified applies for a dc condition.

10.2 Recommended Operating Conditions (see Note 3)

			OPERATION	MIN	NOM	MAX	UNIT
V_{CC}	Core voltage	Commercial	3.3 V	3	3.3	3.6	V
V_{CCP}	PCI I/O clamp voltage, ZV Port I/O voltage	Commercial	3.3 V	3	3.3	3.6	V
			5 V	4.75	5	5.25	
V_{CCCB} V_{CCI} V_{CCL}	PC Card I/O clamp voltage	Commercial	3.3 V	3	3.3	3.6	V
			5 V	4.75	5	5.25	
V_{IH}†	High-level input voltage	PCI	3.3 V	0.5 V_{CCP}		V_{CCP}	V
			5 V	2		V_{CCP}	
		PC Card	3.3 V	0.475 $V_{CCA/B}$		$V_{CCA/B}$	
			5 V	2.4		$V_{CCA/B}$	
		PHY I/F		2		V_{CC}	
		TTL‡		2		V_{CC}	V
		Fail safe§		2.4		V_{CC}	V
V_{IL}†	Low-level input voltage	PCI	3.3 V	0		0.3 V_{CCP}	V
			5 V	0		0.8	
		PC Card	3.3 V	0		0.325 $V_{CCA/B}$	
			5 V	0		0.8	
		PHY I/F		0		0.8	
		TTL‡		0		0.8	V
		Fail safe§		0		0.8	V
V_I	Input voltage	PCI	3.3 V	0		V_{CCP}	V
		PC Card	5 V	0		$V_{CCA/B}$	
		PHY I/F		0		V_{CC}	
		TTL‡		0		V_{CC}	V
		Fail safe§		0		V_{CC}	V
V_O¶	Output voltage	PCI	3.3 V	0		V_{CC}	V
		PC Card	5 V	0		V_{CC}	
		PHY I/F		0		V_{CC}	
		TTL‡		0		V_{CC}	V
		Fail safe§		0		V_{CC}	V
t_t	Input transition time (t_r and t_f)	PCI and PC Card		1		4	ns
		TTL and fail safe		0		6	
T_A	Operating ambient temperature range			0	25	70	°C
T_J#	Virtual junction temperature			0	25	115	°C

† Applies to external inputs and bidirectional buffers without hysteresis

‡ Miscellaneous terminals are 70, 62, 59, 60, 61, 64, 65, 67, 68, and 69 for the PGE packaged device; L11, M9, L8, K8, N9, K9, N10, L10, N11, and M11 for the GGU packaged device; and W12, U10, P9, W10, V10, P10, W11, U11, P11, and R11 for the GHK packaged device ($\overline{\text{SUSPEND}}$, SPKROUT, $\overline{\text{RI\_OUT}}$, multifunction terminals (MFUNC0−MFUNC6), and power switch control terminals).

§ Fail-safe terminals are 75, 117, 131, and 137 for the PGE packaged device; L12, D9, C6, and A4 for the GGU packaged device; and L19, E13, F11, and A9 for the GHK packaged device (card detect and voltage sense pins).

¶ Applies to external output buffers

These junction temperatures reflect simulation conditions. The customer is responsible for verifying junction temperature.

NOTE 3: Unused terminals (input or I/O) must be held high or low to prevent them from floating.

10.3 Electrical Characteristics Over Recommended Operating Conditions (unless otherwise noted)

PARAMETER		PINS	OPERATION	TEST CONDITIONS	MIN	MAX	UNIT
V_{OH}	High-level output voltage	PCI	3.3 V	I_{OH} = −0.5 mA	$0.9V_{CC}$		V
			5 V	I_{OH} = −2 mA	2.4		
		PC Card	3.3 V	I_{OH} = −0.15 mA	$0.9V_{CC}$		
			5 V	I_{OH} = −0.15 mA	2.4		
		PHY I/F	3.3 V	I_{OH} = −4 mA	2.8		
			3.3 V	I_{OH} = −8 mA	V_{CC}−0.6		
		TTL		I_{OH} = −4 mA	V_{CC}−0.6		
				I_{OH} = −8 mA	V_{CC}−0.6		
V_{OL}	Low-level output voltage	PCI	3.3 V	I_{OL} = 1.5 mA		$0.1V_{CC}$	V
			5 V	I_{OL} = 6 mA		0.55	
		PC Card	3.3 V	I_{OL} = 0.7 mA		$0.1V_{CC}$	
			5 V	I_{OL} = 0.7 mA		0.55	
		PHY I/F	3.3 V	I_{OL} = 4 mA		0.5	
			3.3 V	I_{OL} = 8 mA		0.5	
		TTL		I_{OL} = 4 mA		0.5	
				I_{OL} = 8 mA		0.5	
		$\overline{\text{SERR}}$		I_{OL} = 8 mA		0.5	
I_{OZL}	3-state output, high-impedance state output current (see Note 4)	Output terminals	3.6 V	$V_I = V_{CC}$		−1	µA
			5.25 V	$V_I = V_{CC}$		−1	
I_{OZH}	3-state output, high-impedance state output current	Output terminals	3.6 V	$V_I = V_{CC}$†		10	µA
			5.25 V	$V_I = V_{CC}$†		25	
I_{IL}	Low-level input current	Input terminals		V_I = GND		−1	µA
		I/O terminals		V_I = GND		−10	
I_{IH}	High-level input current	Input terminals	3.6 V	$V_I = V_{CC}$‡		10	µA
			5.25 V	$V_I = V_{CC}$‡		20	
		I/O terminals	3.6 V	$V_I = V_{CC}$‡		10	
			5.25 V	$V_I = V_{CC}$‡		25	
		Fail-safe terminals	3.6 V	$V_I = V_{CC}$		10	

† For PCI pins, $V_I = V_{CCP}$. For PC Card pins, $V_I = V_{CCCB}$. For miscellaneous pins, $V_I = V_{CCI}$.
‡ For I/O pins, input leakage (I_{IL} and I_{IH}) includes I_{OZ} leakage of the disabled output.

10.4 PCI Clock/Reset Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature

PARAMETER		ALTERNATE SYMBOL	TEST CONDITIONS	MIN	MAX	UNIT
t_c	Cycle time, PCLK	t_{cyc}		30		ns
t_{wH}	Pulse duration (width), PCLK high	t_{high}		11		ns
t_{wL}	Pulse duration (width), PCLK low	t_{low}		11		ns
$\Delta v/\Delta t$	Slew rate, PCLK	t_r, t_f		1	4	V/ns
t_w	Pulse duration (width), $\overline{\text{PRST}}$	t_{rst}		1		ms
t_{su}	Setup time, PCLK active at end of $\overline{\text{PRST}}$	$t_{rst-clk}$		100		µs

10.5 PCI Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature

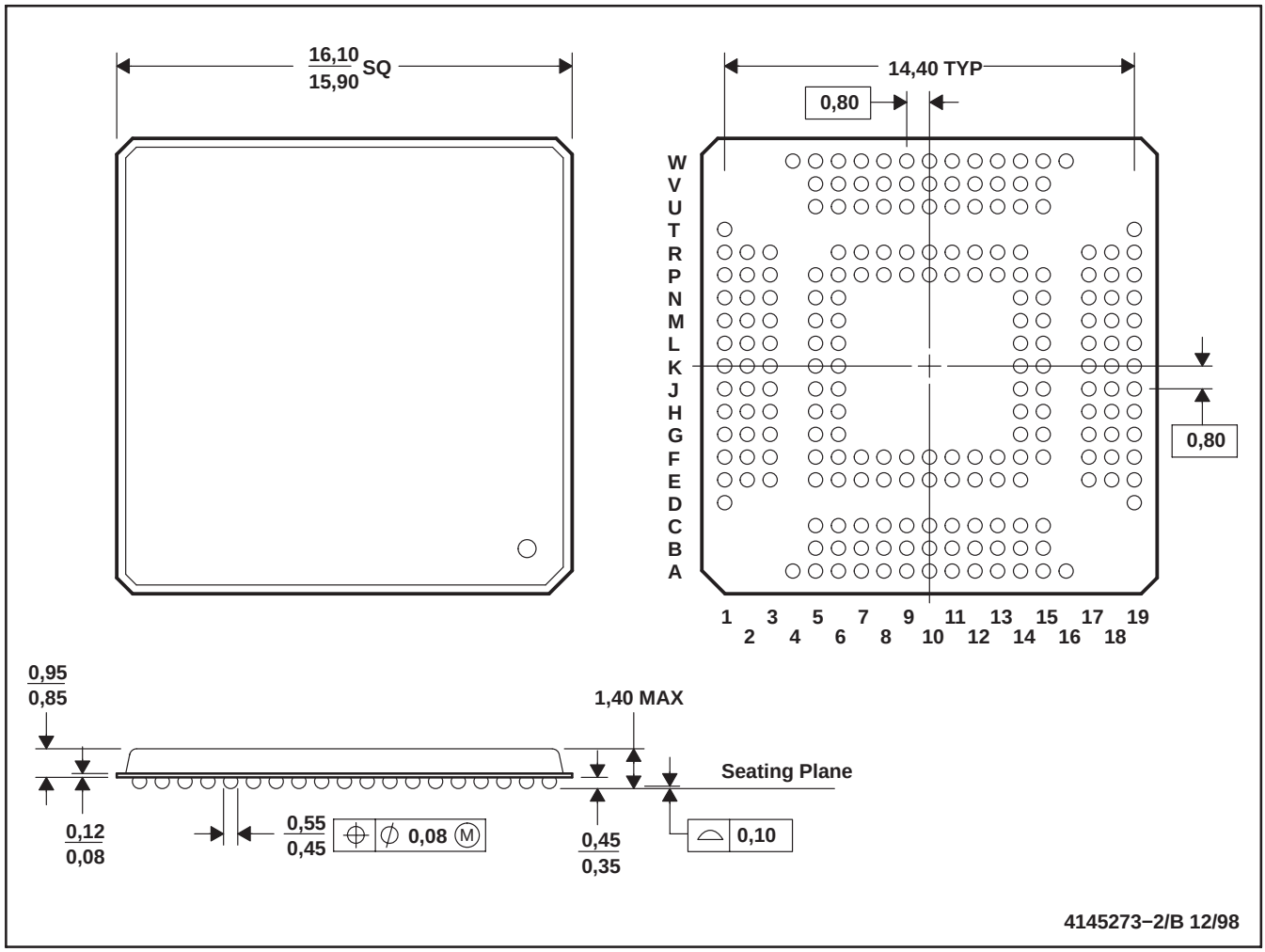
PARAMETER			ALTERNATE SYMBOL	TEST CONDITIONS	MIN	MAX	UNIT
t_{pd}	Propagation delay time, See Note 4	PCLK-to-shared signal valid delay time	t_{val}	C_L = 50 pF, See Note 4		11	ns
		PCLK-to-shared signal invalid delay time	t_{inv}		2		
t_{en}	Enable time, high impedance-to-active delay time from PCLK		t_{on}		2		ns
t_{dis}	Disable time, active-to-high impedance delay time from PCLK		t_{off}			28	ns
t_{su}	Setup time before PCLK valid		t_{su}		7		ns
t_h	Hold time after PCLK high		t_h		0		ns

NOTE 4: PCI shared signals are AD31−AD0, C/$\overline{\text{BE3}}$−C/$\overline{\text{BE0}}$, $\overline{\text{FRAME}}$, $\overline{\text{TRDY}}$, $\overline{\text{IRDY}}$, $\overline{\text{STOP}}$, IDSEL, $\overline{\text{DEVSEL}}$, and PAR.

11 Mechanical Information

The PCI4410 is packaged in either a 209-ball GHK MicroStar BGA or a 208-pin PDV package. The PCI4410 is a single-socket CardBus bridge with an integrated OHCI link. The following shows the mechanical dimensions for the GHK and PDV packages.

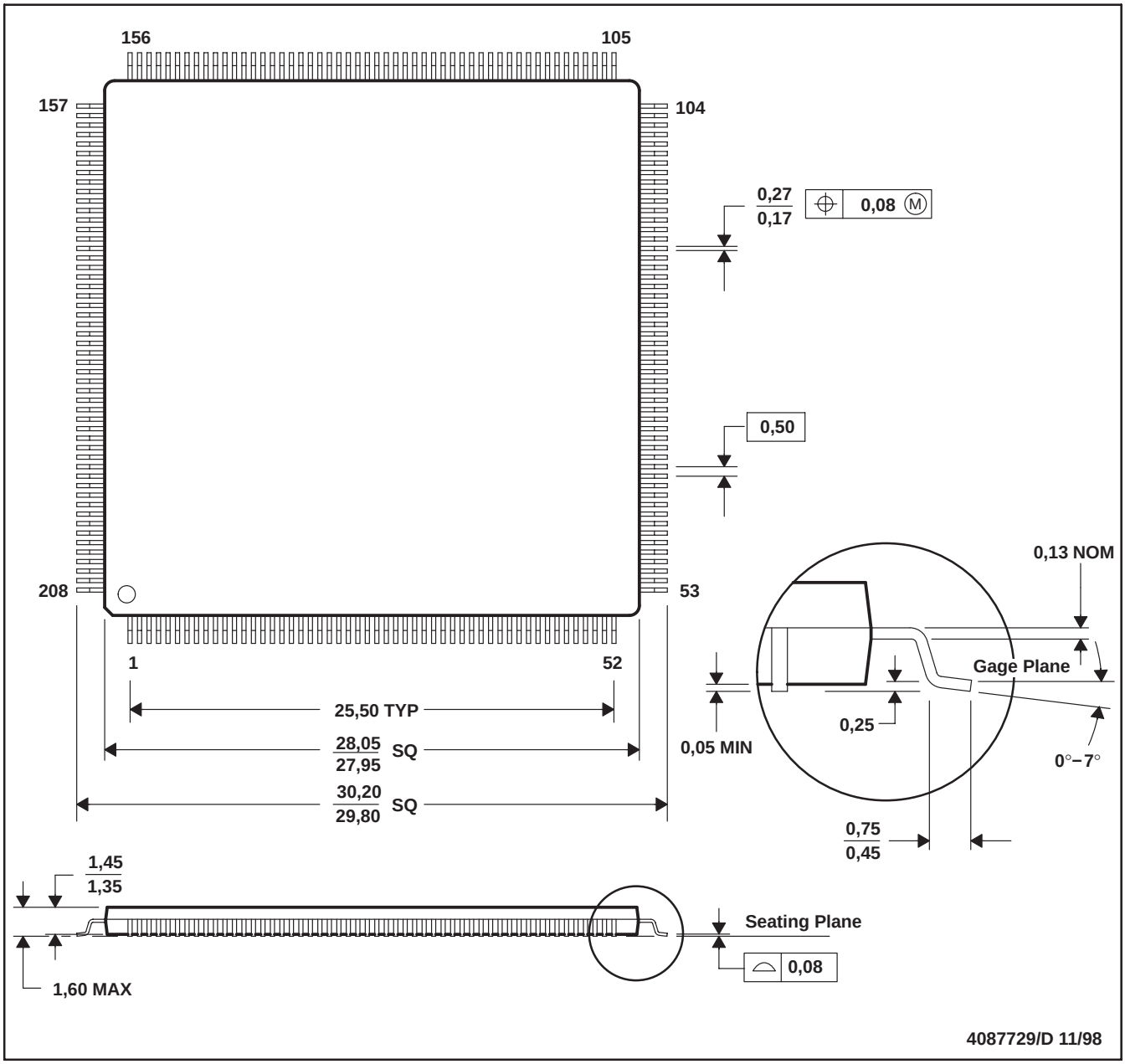
GHK (S-PBGA-N209) PLASTIC BALL GRID ARRAY



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Micro Star BGA configuration.

Micro Star is a trademark of Texas Instruments Incorporated.

PDV (S-PQFP-G208) PLASTIC QUAD FLATPACK



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-026