Texas Instruments

PCI4410A GHK/PDV

PC Card and OHCI Controller

Data Manual

2000

PCI Bus Solutions

Contents

List of Illustrations

List of Tables

1 Introduction

The Texas Instruments PCI4410A device 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 PCI4410A device is a dual-function PCI device compliant with the *PCI Local Bus Specification*. Function 0 provides the independent PC Card socket controller compliant with the *PC Card Standard*. The PCI4410A device 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 buffered internally to allow hot insertion and removal without external buffering. The PCI4410A device is compatible with the register of the Intel™ 82365SL–DF and 82365SL ExCA controllers. The PCI4410A 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 PCI4410A device can be programmed to accept posted writes to improve bus utilization.

Function 1 of the PCI4410A device is compatible with IEEE 1394a-2000 and the latest 1394 open host controller interface (OHCI) specifications. The chip provides the IEEE 1394 link function and is compatible with data rates of 100, 200, and 400 Mbits/s. Deep FIFOs are provided to buffer 1394 data and accommodate large host bus latencies. The PCI4410A device 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 PCI4410A device the best-in-class 1394 OHCI solution.

The PCI4410A device 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 can be programmed and controlled through the multifunction terminals, and an ACPI-compliant programming interface is included for the general-purpose inputs and outputs.

The PCI4410A device is compliant with the latest *PCI Bus Power Management Specification*, and provides several low-power modes that 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 PCI4410A device inputs must be pulled to a valid logic level using a 43-kΩ resistor.

1.2 Features

The PCI4410A device 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
- 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 130 Mbit/s sustained throughput from CardBus-to-PCI and from PCI-to-CardBus
- Interface to parallel single-slot PC Card power-switch interfaces like the TI™ TPS2211 device
- 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
- Compatibility with Intel 82365SL-DF and 82365SL registers
- 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 three outstanding transactions
- OHCI link function is compliant with IEEE 1394-1995 and compatible with IEEE 1394a-2000
- Supports serial bus data rates of 100, 200, and 400 Mbits/s
- 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 1.1)
- *PCI Local Bus Specification* (Revision 2.2)
- *PCI Mobile Design Guide* (Revision 1.0)
- *PCI14xx Implemenation Guide for D3 Wake-Up*
- *PC Card Standard*, Release 7
- *PC 98/99*
- *Serialized IRQ Support for PCI Systems* (Revision 6)
- *Serial Bus Protocol 2* (SBP-2)
- *1394 Open Host Controller Interface Specification* (Revision 1.0)
- *P1394 Standard for a High-Performance Serial Bus* (IEEE 1394-1995)
- *IEEE Standard for a High-Performance Serial Bus—Amendment 1* (IEEE 1394a-2000)

1.4 Trademarks

MicroStar BGA, OHCI-Lynx, and TI are trademarks of Texas Instruments.

Microsoft OnNow is a trademark of Microsoft Corporation.

Intel is a trademark of Intel Corporation.

Maxim is a trademark of Maxim Integrated Products, Inc.

Other trademarks are the property of their respective owners.

1.5 Ordering Information

ORDERING NUMBER	NAME	VOLTAGE	PACKAGE
PCI4410A	PC Card controller	3.3-V, 5-V tolerant I/Os	208-pin LQFP 209-ball PBGA

2 Terminal Descriptions

The PCI4410A device is packaged in either a 209-ball GHK MicroStar BGA™ or a 208-terminal PDV package. The PCI4410A device 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.

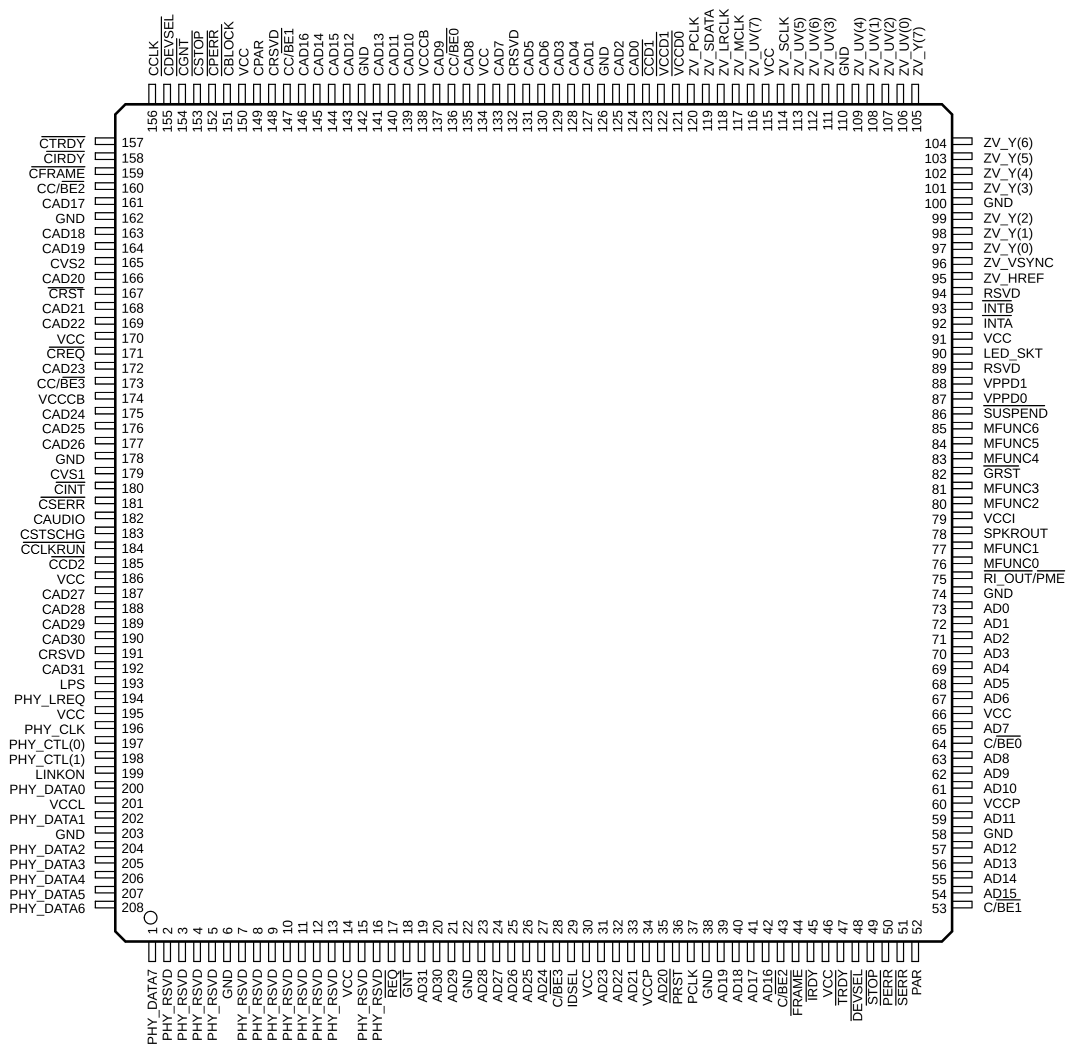


Figure 2–1. PCI-to-CardBus Terminal Diagram

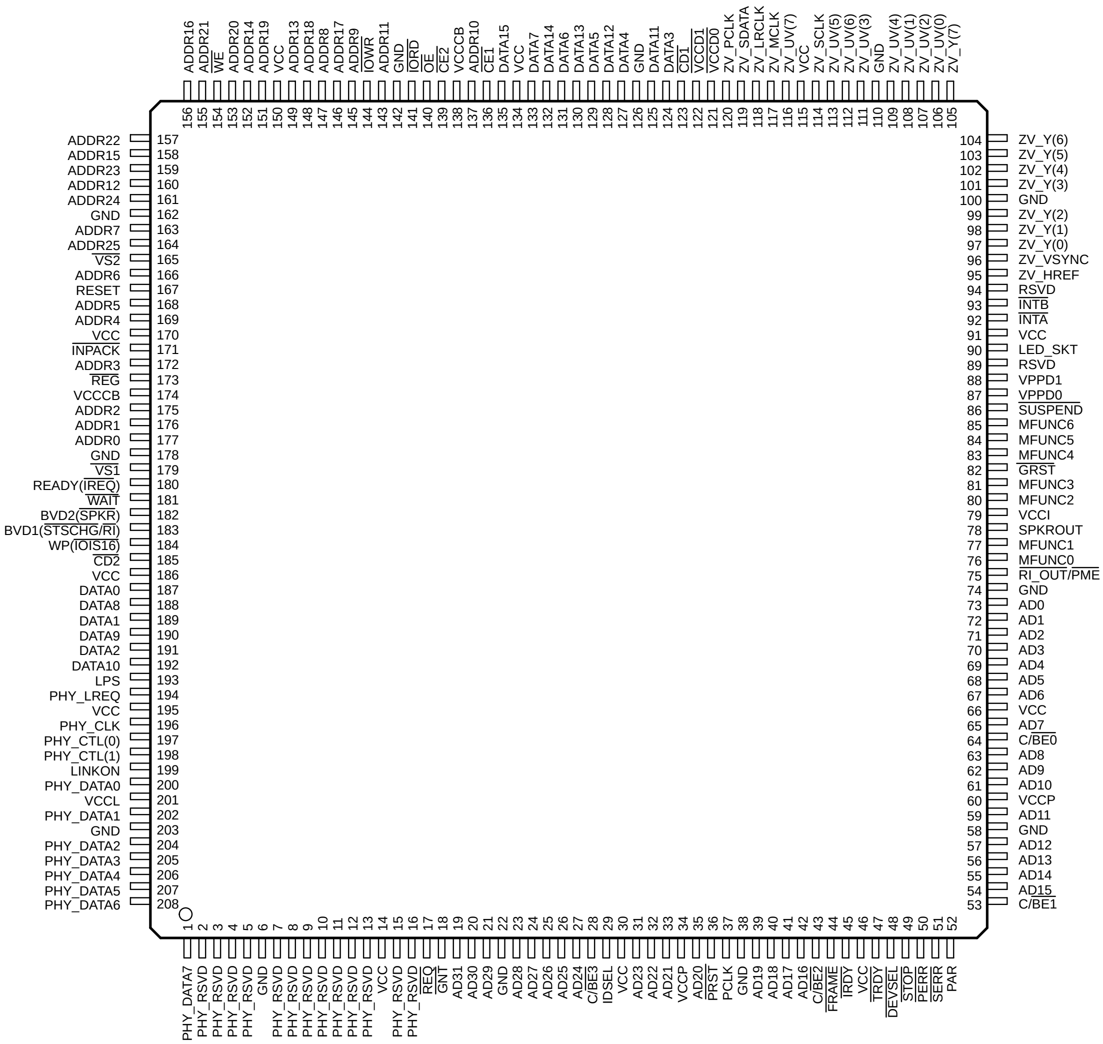


Figure 2–2. PCI-to-PC Card (16-Bit) Terminal Diagram

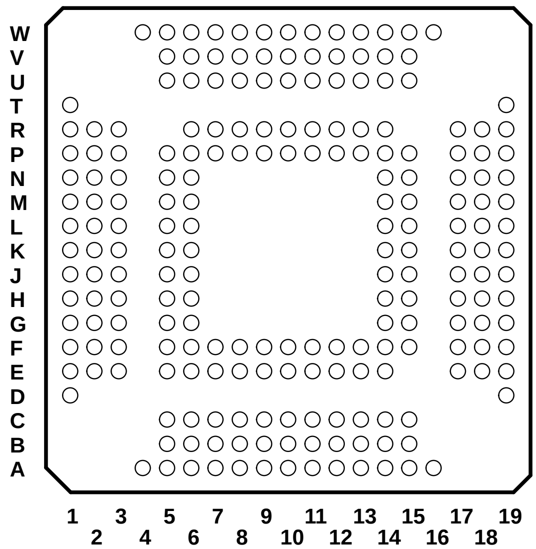


Figure 2–3. MicroStar BGA™ Ball Diagram

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	$\overline{CTRDY}$	ADDR22	184	$\overline{CCLKRUN}$	WP($\overline{IOIS16}$)
131	CAD5	DATA6	158	$\overline{CIRDY}$	ADDR15	185	$\overline{CCD2}$	$\overline{CD2}$
132	CRSVD	DATA14	159	$\overline{CFRAME}$	ADDR23	186	V_{CC}	V_{CC}
133	CAD7	DATA7	160	CC/$\overline{BE2}$	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/$\overline{BE0}$	$\overline{CE1}$	163	CAD18	ADDR7	190	CAD30	DATA9
137	CAD9	ADDR10	164	CAD19	ADDR25	191	CRSVD	DATA2
138	V_{CCCB}	V_{CCCB}	165	CVS2	$\overline{VS2}$	192	CAD31	DATA10
139	CAD10	$\overline{CE2}$	166	CAD20	ADDR6	193	LPS	LPS
140	CAD11	$\overline{OE}$	167	$\overline{CRST}$	RESET	194	PHY_LREQ	PHY_LREQ
141	CAD13	$\overline{IORD}$	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	$\overline{IOWR}$	171	$\overline{CREQ}$	$\overline{INPACK}$	198	PHY_CTL(1)	PHY_CTL(1)
145	CAD14	ADDR9	172	CAD23	ADDR3	199	LINKON	LINKON
146	CAD16	ADDR17	173	CC/$\overline{BE3}$	$\overline{REG}$	200	PHY_DATA0	PHY_DATA0
147	CC/$\overline{BE1}$	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	$\overline{CBLOCK}$	ADDR19	178	GND	GND	205	PHY_DATA3	PHY_DATA3
152	$\overline{CPERR}$	ADDR14	179	CVS1	$\overline{VS1}$	206	PHY_DATA4	PHY_DATA4
153	$\overline{CSTOP}$	ADDR20	180	$\overline{CINT}$	READY($\overline{IREQ}$)	207	PHY_DATA5	PHY_DATA5
154	$\overline{CGNT}$	$\overline{WE}$	181	$\overline{CSERR}$	$\overline{WAIT}$	208	PHY_DATA6	PHY_DATA6
155	$\overline{CDEVSEL}$	ADDR21	182	CAUDIO	BVD2($\overline{SPKR}$)			
156	CCLK	ADDR16	183	CSTSCHG	BVD1 ($\overline{STSCHG}$/$\overline{RI}$)			

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	$\overline{IORD}$
A5	GND	GND	E9	CAD29	DATA1	H15	GND	GND
A6	LINKON	LINKON	E10	CSTSCHG	BVD1 ($\overline{STSCHG}$/$\overline{RI}$)	H17	CAD11	$\overline{OE}$
A7	V_{CC}	V_{CC}	E11	GND	GND	H18	CAD10	$\overline{CE2}$
A8	CAD30	DATA9	E12	$\overline{CREQ}$	$\overline{INPACK}$	H19	V_{CCCB}	V_{CCCB}
A9	$\overline{CCD2}$	$\overline{CD2}$	E13	CVS2	$\overline{VS2}$	J1	AD31	AD31
A10	$\overline{CINT}$	READY($\overline{IREQ}$)	E14	$\overline{CFRAME}$	ADDR23	J2	AD30	AD30
A11	CAD24	ADDR2	E17	$\overline{CDEVSEL}$	ADDR21	J3	AD29	AD29
A12	V_{CCCB}	V_{CCCB}	E18	$\overline{CSTOP}$	ADDR20	J5	GND	GND
A13	V_{CC}	V_{CC}	E19	$\overline{CBLOCK}$	ADDR19	J6	AD28	AD28
A14	CAD20	ADDR6	F1	PHY_RSVD	PHY_RSVD	J14	CC/$\overline{BE0}$	$\overline{CE1}$
A15	GND	GND	F2	PHY_RSVD	PHY_RSVD	J15	CAD9	ADDR10
A16	$\overline{CTRDY}$	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	$\overline{CSERR}$	$\overline{WAIT}$	F10	$\overline{CCLKRUN}$	WP($\overline{IOIS16}$)	K5	AD24	AD24
B11	CAD25	ADDR1	F11	CVS1	$\overline{VS1}$	K6	C/$\overline{BE3}$	C/$\overline{BE3}$
B12	CC/$\overline{BE3}$	$\overline{REG}$	F12	$\overline{CRST}$	RESET	K14	CRSVD	DATA14
B13	CAD22	ADDR4	F13	CC/$\overline{BE2}$	ADDR12	K15	CAD5	DATA6
B14	CAD19	ADDR25	F14	$\overline{CPERR}$	ADDR14	K17	CAD6	DATA13
B15	CAD17	ADDR24	F15	$\overline{CGNT}$	$\overline{WE}$	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/$\overline{BE1}$	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($\overline{SPKR}$)	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	$\overline{CIRDY}$	ADDR15	G17	CAD14	ADDR9	L19	$\overline{CCD1}$	$\overline{CD1}$
D1	PHY_DATA7	PHY_DATA7	G18	CAD15	$\overline{IOWR}$	M1	V_{CCP}	V_{CCP}
D19	CCLK	ADDR16	G19	CAD12	ADDR11	M2	AD20	AD20
E1	GND	GND	H1	$\overline{GNT}$	$\overline{GNT}$	M3	$\overline{PRST}$	$\overline{PRST}$
E2	PHY_RSVD	PHY_RSVD	H2	$\overline{REQ}$	$\overline{REQ}$	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	VCCD0	VCCD0	P19	ZV_SCLK	ZV_SCLK	U15	ZV_Y(5)	ZV_Y(5)
M19	VCCD1	VCCD1	R1	TRDY	TRDY	V5	AD12	AD12
N1	AD19	AD19	R2	STOP	STOP	V6	V_{CCP}	V_{CCP}
N2	AD18	AD18	R3	SERR	SERR	V7	AD7	AD7
N3	AD17	AD17	R6	AD14	AD14	V8	AD4	AD4
N5	IRDY	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	GRST	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	INTA	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/BE2	C/BE2	R14	ZV_Y(4)	ZV_Y(4)	W4	C/BE1	C/BE1
P2	FRAME	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	PERR	PERR	R19	GND	GND	W7	V_{CC}	V_{CC}
P6	DEVSEL	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	RI_OUT/PME	RI_OUT/PME	U6	AD11	AD11	W11	MFUNC3	MFUNC3
P10	MFUNC2	MFUNC2	U7	C/BE0	C/BE0	W12	SUSPEND	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	INTB	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	$\overline{CRST}$	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	$\overline{CSERR}$	181	B10	PHY_DATA1	202	C6
AD5	68	U8	CAD16	146	G14	$\overline{CSTOP}$	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	$\overline{CTRDY}$	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	$\overline{DEVSEL}$	48	P6	PHY_DATA7	1	D1
AD11	59	U6	CAD22	169	B13	$\overline{FRAME}$	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/$\overline{BE0}$	64	U7	GND	178	E11	PHY_RSVD	13	G2
AD23	31	L3	C/$\overline{BE1}$	53	W4	GND	203	A5	PHY_RSVD	15	H5
AD24	27	K5	C/$\overline{BE2}$	43	P1	$\overline{GNT}$	18	H1	PHY_RSVD	16	H3
AD25	26	K3	C/$\overline{BE3}$	28	K6	$\overline{GRST}$	82	V11	$\overline{PRST}$	36	M3
AD26	25	K2	$\overline{CBLOCK}$	151	E19	IDSEL	29	L1	$\overline{REQ}$	17	H2
AD27	24	K1	CC/$\overline{BE0}$	136	J14	$\overline{INTA}$	92	V13	$\overline{RI\_OUT}$/$\overline{PME}$	75	P9
AD28	23	J6	CC/$\overline{BE1}$	147	F19	$\overline{INTB}$	93	U13	RSVD	89	P12
AD29	21	J3	CC/$\overline{BE2}$	160	F13	$\overline{IRDY}$	45	N5	RSVD	94	P13
AD30	20	J2	CC/$\overline{BE3}$	173	B12	LED_SKT	90	R12	$\overline{SERR}$	51	R3
AD31	19	J1	$\overline{CCD1}$	123	L19	LINKON	199	A6	SPKROUT	78	U10
CAD0	124	L18	$\overline{CCD2}$	185	A9	LPS	193	F8	$\overline{STOP}$	49	R2
CAD1	127	L14	CCLK	156	D19	MFUNC0	76	W10	$\overline{SUSPEND}$	86	W12
CAD2	125	L17	$\overline{CCLKRUN}$	184	F10	MFUNC1	77	V10	$\overline{TRDY}$	47	R1
CAD3	129	K18	$\overline{CDEVSEL}$	155	E17	MFUNC2	80	P10	V_{CC}	14	G1
CAD4	128	K19	$\overline{CFRAME}$	159	E14	MFUNC3	81	W11	V_{CC}	30	L2
CAD5	131	K15	$\overline{CGNT}$	154	F15	MFUNC4	83	U11	V_{CC}	46	P3
CAD6	130	K17	$\overline{CINT}$	180	A10	MFUNC5	84	P11	V_{CC}	66	W7
CAD7	133	J19	$\overline{CIRDY}$	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	$\overline{CPERR}$	152	F14	PCLK	37	M6	V_{CC}	134	J18
CAD10	139	H18	$\overline{CREQ}$	171	E12	$\overline{PERR}$	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
$\overline{\text{VCCD0}}$	121	M18	ZV_LRCLK	118	N19	ZV_UV(4)	109	R18	ZV_Y(4)	102	R14
$\overline{\text{VCCD1}}$	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	DEVSEL	48	P6	PHY_DATA2	204	F6
AD1	72	V9	ADDR11	143	G19	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	GNT	18	H1	PHY_RSVD	10	F1
AD15	54	U5	ADDR25	164	B14	GRST	82	V11	PHY_RSVD	11	H6
AD16	42	N6	BVD1 (STSCHG/RI)	183	E10	IDSEL	29	L1	PHY_RSVD	12	G3
AD17	41	N3	BVD2(SPKR)	182	C10	INPACK	171	E12	PHY_RSVD	13	G2
AD18	40	N2	C/BE0	64	U7	INTA	92	V13	PHY_RSVD	15	H5
AD19	39	N1	C/BE1	53	W4	INTB	93	U13	PHY_RSVD	16	H3
AD20	35	M2	C/BE2	43	P1	IRDY	45	N5	PRST	36	M3
AD21	33	L5	C/BE3	28	K6	IORD	141	H14	READY(IREQ)	180	A10
AD22	32	L6	CD1	123	L19	IOWR	144	G18	REG	173	B12
AD23	31	L3	CD2	185	A9	LED_SKT	90	R12	REQ	17	H2
AD24	27	K5	CE1	136	J14	LINKON	199	A6	RESET	167	F12
AD25	26	K3	CE2	139	H18	LPS	193	F8	RI_OUT/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	SERR	51	R3
AD29	21	J3	DATA3	124	L18	MFUNC3	81	W11	SPKROUT	78	U10
AD30	20	J2	DATA4	127	L14	MFUNC4	83	U11	STOP	49	R2
AD31	19	J1	DATA5	129	K18	MFUNC5	84	P11	SUSPEND	86	W12
ADDR0	177	C11	DATA6	131	K15	MFUNC6	85	R11	TRDY	47	R1
ADDR1	176	B11	DATA7	133	J19	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	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	$\overline{\text{VS1}}$	179	F11	ZV_SDATA	119	M15	ZV_Y(0)	97	R13
V_{CCCB}	174	A12	$\overline{\text{VS2}}$	165	E13	ZV_UV(0)	106	R17	ZV_Y(1)	98	U14
$\overline{\text{VCCD0}}$	121	M18	$\overline{\text{WAIT}}$	181	B10	ZV_UV(1)	108	N14	ZV_Y(2)	99	W15
$\overline{\text{VCCD1}}$	122	M19	$\overline{\text{WE}}$	154	F15	ZV_UV(2)	107	P15	ZV_Y(3)	101	V15
V_{CCI}	79	R10	WP($\overline{\text{IOIS16}}$)	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 also are listed for convenient reference.

Table 2–5. Power-Supply Terminals

TERMINAL			DESCRIPTION
NAME	NUMBER		
	PDV	GHK	
GND	6, 22, 38, 58, 74, 100, 110, 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, GRST, and 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, INTA, INTB LED_SKT, VCCD0, VCCD1, VPPD0, VPPD1

Table 2–6. PC Card Power-Switch Terminals

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

Table 2–7. PCI System Terminals

TERMINAL			I/O	DESCRIPTION
NAME	NUMBER			
	PDV	GHK		
GRST	82	V11	I	Global reset. When global reset is asserted, GRST causes the PCI4410A device to place all output buffers in a high-impedance state and reset all internal registers. When GRST is asserted, the device is completely in its default state. For systems that require wake-up from D3, GRST normally is asserted only during initial boot. 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, GRST should be tied to PRST. When the SUSPEND mode is enabled, the device is protected from 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.
PRST	36	M3	I	PCI bus reset. When the PCI bus reset is asserted, PRST causes the PCI4410A device to place all output buffers in a high-impedance state and reset internal registers. When PRST is asserted, the device is completely nonfunctional. After PRST is deasserted, the PCI4410A device is in a default state. When SUSPEND and PRST are asserted, the device is protected from 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			I/O	DESCRIPTION
NAME	NUMBER			
	PDV	GHK		
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/$\overline{\text{BE3}}$ C/$\overline{\text{BE2}}$ C/$\overline{\text{BE1}}$ C/$\overline{\text{BE0}}$	28 43 53 64	K6 P1 W4 U7	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/$\overline{\text{BE3}}$–C/$\overline{\text{BE0}}$ 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. C/$\overline{\text{BE0}}$ applies to byte 0 (AD7–AD0), C/$\overline{\text{BE1}}$ applies to byte 1 (AD15–AD8), C/$\overline{\text{BE2}}$ applies to byte 2 (AD23–AD16), and C/$\overline{\text{BE3}}$ applies to byte 3 (AD31–AD24).
PAR	52	T1	I/O	PCI bus parity. In all PCI bus read and write cycles, the PCI4410A device calculates even parity across the AD31–AD0 and C/$\overline{\text{BE3}}$–C/$\overline{\text{BE0}}$ buses. As an initiator during PCI cycles, the PCI4410A device 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 ($\overline{\text{PERR}}$).

Table 2–9. PCI Interface Control Terminals

TERMINAL			I/O	DESCRIPTION
NAME	NUMBER			
	PDV	GHK		
DEVSEL	48	P6	I/O	PCI device select. The PCI4410A device asserts DEVSEL to claim a PCI cycle as the target device. As a PCI initiator on the bus, the PCI4410A device monitors DEVSEL until a target responds. If no target responds before timeout occurs, the PCI4410A device terminates the cycle with an initiator abort.
FRAME	44	P2	I/O	PCI cycle frame. FRAME is driven by the initiator of a bus cycle. FRAME is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When FRAME is deasserted, the PCI bus transaction is in the final data phase.
GNT	18	H1	I	PCI bus grant. GNT is driven by the PCI bus arbiter to grant the PCI4410A device access to the PCI bus after the current data transaction has completed. 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 PCI4410A device during configuration space accesses. IDSEL can be connected to one of the upper 24 PCI address lines on the PCI bus.
IRDY	45	N5	I/O	PCI initiator ready. 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, when both IRDY and TRDY are asserted. Until IRDY and TRDY are both sampled asserted, wait states are inserted.
PERR	50	P5	I/O	PCI parity error indicator. PERR is driven by a PCI device to indicate that calculated parity does not match PAR when PERR is enabled through bit 6 (PERR_EN) of the command register (PCI offset 04h, see Section 4.4).
REQ	17	H2	O	PCI bus request. REQ is asserted by the PCI4410A device to request access to the PCI bus as an initiator.
SERR	51	R3	O	PCI system error. SERR is an output that is pulsed from the PCI4410A device when enabled through bit 8 (SERR_EN) of the command register (PCI offset 04h, see Section 4.4) indicating a system error has occurred. The PCI4410A device need not be the target of the PCI cycle to assert this signal. When SERR is enabled in the command register, this signal also pulses, indicating that an address parity error has occurred on a CardBus interface.
STOP	49	R2	I/O	PCI cycle stop signal. STOP is driven by a PCI target to request the initiator to stop the current PCI bus transaction. STOP is used for target disconnects and is commonly asserted by target devices that do not support burst data transfers.
TRDY	47	R1	I/O	PCI target ready. 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 IRDY and TRDY are asserted. Until both IRDY and TRDY are asserted, wait states are inserted.

Table 2–10. Multifunction and Miscellaneous Terminals

TERMINAL			I/O	DESCRIPTION
NAME	NUMBER			
	PDV	GHK		
INTA	92	V13	O	Parallel PCI interrupt. INTA
INTB	93	U13	O	Parallel PCI interrupt. 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 INTA, GPI0, GPO0, socket activity LED output, ZV switching outputs, CardBus audio PWM, 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, GPE, or a parallel IRQ. See Section 4.32, *Multifunction Routing Register*, for configuration details. Serial data (SDA). When VCCD0 and 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, GPE, 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 LOCK, GPI3, GPO3, socket activity LED output, ZV switching outputs, CardBus audio PWM, GPE, RI_OUT, or a parallel IRQ. See Section 4.32, *Multifunction Routing Register*, for configuration details. Serial clock (SCL). When VCCD0 and 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, 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 CLKRUN or a parallel IRQ. See Section 4.32, *Multifunction Routing Register*, for configuration details.
RI_OUT/PME	75	P9	O	Ring indicate out and power-management event output. Terminal provides an output for ring-indicate or PME signals.
SPKROUT	78	U10	O	Speaker output. SPKROUT is the output to the host system that can carry SPKR or CAUDIO through the PCI4410A device from the PC Card interface. SPKROUT is driven as the exclusive-OR combination of card SPKR//CAUDIO inputs.
SUSPEND	86	W12	I	Suspend. SUSPEND protects the internal registers from clearing when the GRST or 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	125	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 PCI4410A device 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 connected internally 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, 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 PCI4410A device 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 PCI4410A device 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 PCI4410A device 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 PCI4410A device asserts IOWR during transfers from host memory to the PC Card.
OE	140	H17	O	Output enable. OE is driven low by the PCI4410A device 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 PCI4410A device 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 generally is 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 PCI4410A device 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 also is 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 PCI4410A device 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, 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 PCI4410A device 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 PCI4410A device 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			I/O	DESCRIPTION
NAME	NUMBER			
	PDV	GHK		
CAD31 CAD30 CAD29 CAD28 CAD27 CAD26 CAD25 CAD24 CAD23 CAD22 CAD21 CAD20 CAD19 CAD18 CAD17 CAD16 CAD15 CAD14 CAD13 CAD12 CAD11 CAD10 CAD9 CAD8 CAD7 CAD6 CAD5 CAD4 CAD3 CAD2 CAD1 CAD0	192 190 189 188 187 177 176 175 172 169 168 166 164 163 161 146 144 145 141 143 140 139 137 135 133 130 131 128 129 125 127 124	C8 A8 E9 F9 C9 C11 B11 A11 C12 B13 C13 A14 B14 C14 B15 G14 G18 G17 H14 G19 H17 H18 J15 J17 J19 K17 K15 K19 K18 L17 L14 L18	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.
CC/$\overline{BE3}$ CC/$\overline{BE2}$ CC/$\overline{BE1}$ CC/$\overline{BE0}$	173 160 147 136	B12 F13 F19 J14	I/O	CardBus bus commands and byte enables. CC/$\overline{BE3}$–CC/$\overline{BE0}$ are multiplexed on the same CardBus terminals. During the address phase of a CardBus cycle, CC/$\overline{BE3}$–CC/$\overline{BE0}$ 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. CC/$\overline{BE0}$ applies to byte 0 (CAD7–CAD0), CC/$\overline{BE1}$ applies to byte 1 (CAD15–CAD8), CC/$\overline{BE2}$ applies to byte 2 (CAD23–CAD16), and CC/$\overline{BE3}$ applies to byte 3 (CAD31–CAD24).
CPAR	149	G15	I/O	CardBus parity. In all CardBus read and write cycles, the PCI4410A device calculates even parity across the CAD and CC/$\overline{BE}$ buses. As an initiator during CardBus cycles, the PCI4410A device 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 PCI4410A device supports the binary audio mode and outputs a binary signal from the card to SPKROUT.
CBLOCK	151	E19	I/O	CardBus lock. CBLOCK is used to gain exclusive access to a target.
CCD1 CCD2	123 185	L19 A9	I	CardBus detect 1 and CardBus detect 2. CCD1 and CCD2 are used in conjunction with CVS1 and CVS2 to identify card insertion and interrogate cards to determine the operating voltage and card type.
CDEVSEL	155	E17	I/O	CardBus device select. The PCI4410A device asserts CDEVSEL to claim a CardBus cycle as the target device. As a CardBus initiator on the bus, the PCI4410A device monitors CDEVSEL until a target responds. If no target responds before timeout occurs, the PCI4410A device terminates the cycle with an initiator abort.
CFRAME	159	E14	I/O	CardBus cycle frame. CFRAME is driven by the initiator of a CardBus bus cycle. CFRAME is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When CFRAME is deasserted, the CardBus bus transaction is in the final data phase.
CGNT	154	F15	O	CardBus bus grant. CGNT is driven by the PCI4410A device to grant a CardBus PC Card access to the CardBus bus after the current data transaction has been completed.
CINT	180	A10	I	CardBus interrupt. CINT is asserted low by a CardBus PC Card to request interrupt servicing from the host.
CIRDY	158	C15	I/O	CardBus initiator ready. 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 CIRDY and CTRDY are asserted. Until both CIRDY and CTRDY are sampled asserted, wait states are inserted.
CPERR	152	F14	I/O	CardBus parity error. 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.
CREQ	171	E12	I	CardBus request. CREQ indicates to the arbiter that the CardBus PC Card desires use of the CardBus bus as an initiator.
CSERR	181	B10	I	CardBus system error. CSERR reports address parity errors and other system errors that could lead to catastrophic results. CSERR is driven by the card synchronous to CCLK, but deasserted by a weak pullup, and may take several CCLK periods. The PCI4410A device can report CSERR to the system by assertion of SERR on the PCI interface.
CSTOP	153	E18	I/O	CardBus stop. CSTOP is driven by a CardBus target to request the initiator to stop the current CardBus transaction. 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.
CTRDY	157	A16	I/O	CardBus target ready. 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 CIRDY and 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 CCD1 and CCD2 to identify card insertion and interrogate cards to determine the operating voltage and card type.

Table 2–16. IEEE 1394 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 drive these terminals only 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
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 PCI4410A device. Figure 3–1 shows connections to the PCI4410A device. 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: SUSPEND, RI_OUT/PME (power-management control signal), and SPKROUT.

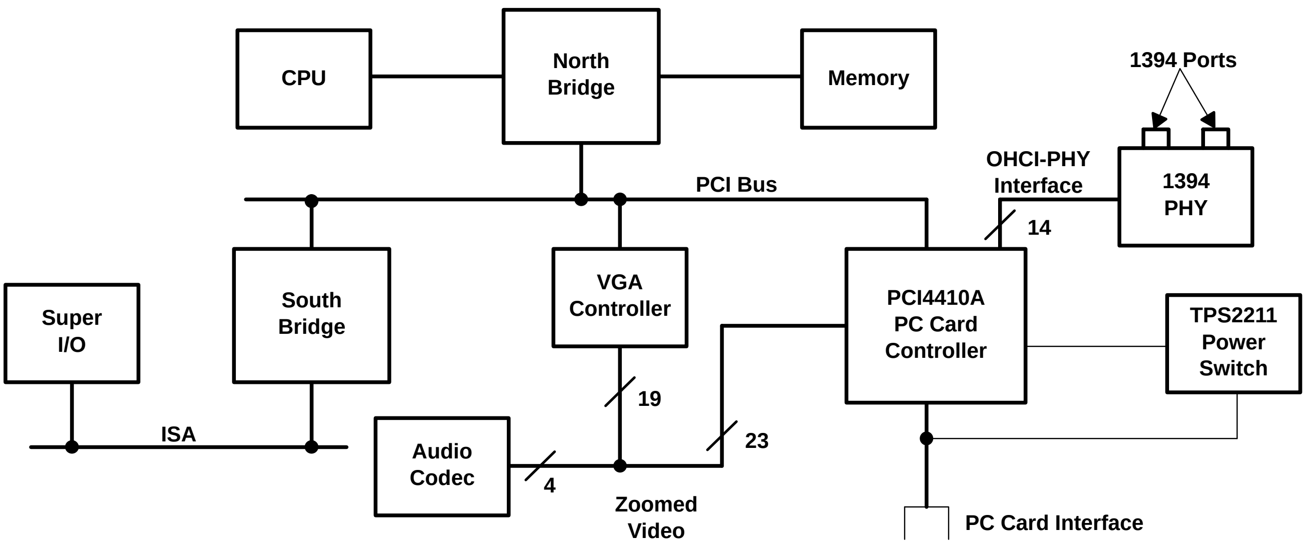


Figure 3–1. PCI4410A System Block Diagram

3.1 Power-Supply Sequencing

The PCI4410A device contains 3.3-V I/O buffers with 5-V tolerance, requiring a core power supply and clamp voltages. The core power supply always is 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 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 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 PCI4410A device meets the ac specifications of the *PC Card Standard* and the *PCI Local Bus Specification.*

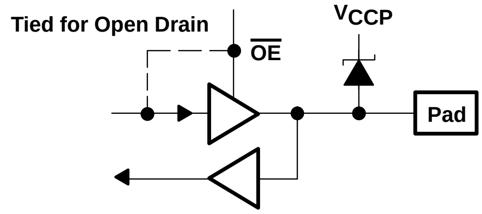


Figure 3–2. 3-State Bidirectional Buffer

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 PCI4410A device 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 always is 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 PCI4410A device 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, V_{CCP} can be connected to a 5-V power supply.

The PCI4410A device 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 PCI4410A device is fully compliant with the *PCI Local Bus Specification*. The PCI4410A device provides all required signals for PCI master or slave operation, and can 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 PCI4410A device provides the optional interrupt signal $\overline{INTA}$.

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

The bus-locking protocol defined in the *PCI Local Bus Specification* is not highly recommended, but is provided on the PCI4410A device as an additional compatibility feature. The PCI $\overline{LOCK}$ 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 $\overline{LOCK}$ is supported only by PCI-to-CardBus bridges in the downstream direction (away from the processor).

PCI $\overline{LOCK}$ indicates an atomic operation that may require multiple transactions to complete. When $\overline{LOCK}$ is asserted, nonexclusive transactions can proceed to an address that currently is not locked. A grant to start a transaction on the PCI bus does not guarantee control of $\overline{LOCK}$; control of $\overline{LOCK}$ is obtained under its own protocol. It is possible for different initiators to use the PCI bus while a single master retains ownership of $\overline{LOCK}$. Note that the CardBus signal for this protocol is $\overline{CBLOCK}$ 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 $\overline{LOCK}$ 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 $\overline{LOCK}$ protocol. In this scenario, the arbiter will not grant the bus to any other agent (other than the $\overline{LOCK}$ master) while $\overline{LOCK}$ 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 PCI4410A device supports all $\overline{LOCK}$ 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 devices such as PCI-to-PCI bridges are used. 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 LOCK.

3.4.2 Loading Subsystem Identification

The subsystem vendor ID register (PCI offset 40h, see Section 4.26) and subsystem ID register (PCI offset 42h, 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 PCI4410A device 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 (PCI offset 80h, see Section 4.29). 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 PCI4410A device loads the data from the serial EEPROM after a reset of the primary bus. Note that the SUSPEND input gates the PCI reset from the entire PCI4410A core, including the serial bus state machine (see Section 3.8.4, *Suspend Mode*, for details on using SUSPEND).

The PCI4410A device 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 PCI4410A device:

- 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 *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 *PC Card Standard* and in Table 3–1.

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

$\overline{\text{CD2}}$//$\overline{\text{CCD2}}$	$\overline{\text{CD1}}$//$\overline{\text{CCD1}}$	$\overline{\text{VS2}}$//CVS2	$\overline{\text{VS1}}$//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 $\overline{\text{CCD1}}$	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 $\overline{\text{CCD2}}$	Ground	LV	CardBus PC Card	3.3 V and X.X V
Connect to CVS1	Ground	Ground	Connect to $\overline{\text{CCD2}}$	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 $\overline{\text{CCD2}}$	Open	LV	CardBus PC Card	Y.Y V
Ground	Connect to CVS2	Connect to $\overline{\text{CCD1}}$	Open	LV	CardBus PC Card	X.X V and Y.Y V
Connect to CVS1	Ground	Open	Connect to $\overline{\text{CCD2}}$	LV	CardBus PC Card	Y.Y V
Ground	Connect to CVS1	Ground	Connect to $\overline{\text{CCD1}}$	Reserved		
Ground	Connect to CVS2	Connect to $\overline{\text{CCD1}}$	Ground	Reserved		

3.5.2 P²C Power-Switch Interface (TPS2211)

The PCI4410A device provides a P²C (PCMCIA peripheral control) interface for control of the PC Card power switch. The $\overline{\text{VCCD}}$ and VPPD terminals are used with the TI TPS2211 single-slot PC Card power-switch interface to provide power-switch support. Figure 3–3 shows terminal assignments for the TPS2211 power-switch interface. Figure 3–4 illustrates a typical application, where the PCI4410A device represents the PC Card controller.

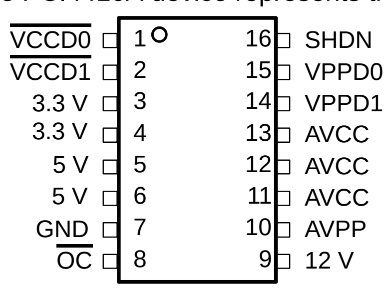


Figure 3–3. TPS2211 Terminal Assignments

The PCI4410A device also includes support for the Maxim™ 1602 and Micrel MIC2562A single-channel CardBus power switches. Application of these power switches is similar to that of the TPS2211 power-switch interface.

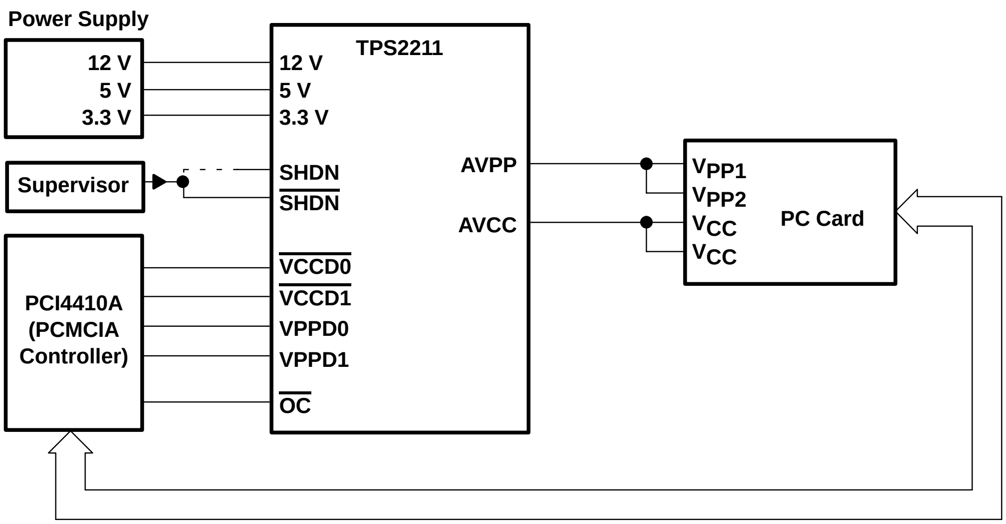


Figure 3–4. TPS2211 Typical Application

3.5.3 Zoomed-Video Support

The zoomed-video (ZV) port on the PCI4410A device provides an internally buffered 16-bit ZV PC Card data path. This internal routing is programmed through the card control register (PCI offset 91h, bits 5 and 6). Figure 3–5 summarizes the ZV subsystem implemented in the PCI4410A device, and details the bit functions found in the card control register.

When ZV PORT_ENABLE is enabled, the ZV output terminals are enabled and allow the PCI4410A device to route the ZV data. However, no data is transmitted unless ZVENABLE (PCI offset 91h, bit 6) is enabled. If ZVENABLE is set to low, the ZV output port drives a logic 0 on the ZV bus of the PCI4410A device.

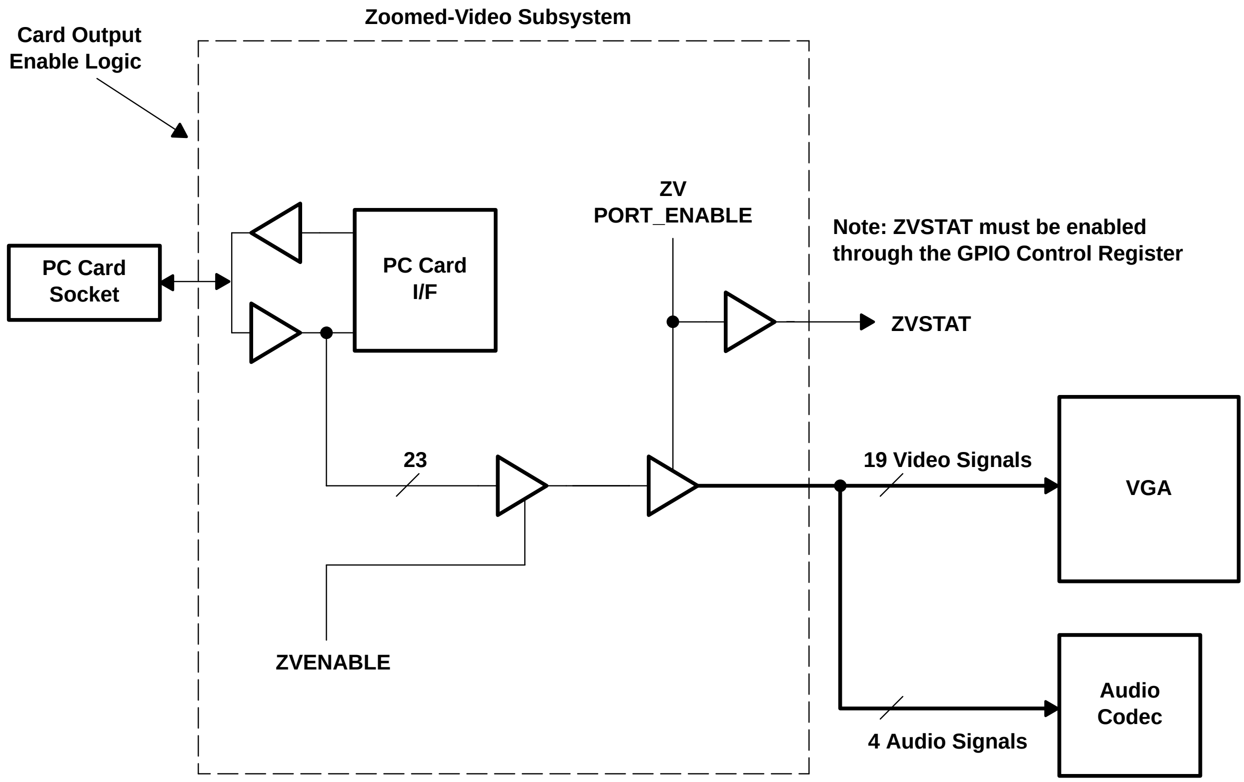


Figure 3–5. Zoomed-Video Subsystem

3.5.4 Ultra Zoomed Video

Ultra zoomed video is an enhancement to the PCI4410A DMA engine and is intended to improve the 16-bit bandwidth for MPEG I and MPEG II decoder PC Cards. This enhancement allows the PCI4410A device to fetch 32 bits of data from memory, versus the PCI11XX/12XX 16-bit fetch capability. This enhancement allows a higher sustained throughput to the 16-bit PC Card because the PCI4410A device prefetches an extra 16 bits (32 bits total) during each PCI read transaction. If the PCI bus becomes busy, the PCI4410A device 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 PCI11XX and PCI12XX 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 PCI14XX series CardBus controllers.

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

Additional functionality for the PCI4410A device versus the PCI12xx 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 PCI4410A device so that neither the PCI clock nor an external clock is required for the PCI4410A device 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 (PCI offset 80h, see Section 4.29) to a 1. This function is disabled by default.

3.5.7 Integrated Pullup Resistors for PC Card Interface

The *PC Card Standard* requires pullup resistors on various terminals to support both CardBus and 16-bit card configurations. Unlike the PCI1210 or PCI1211 device, which required external pullup resistors, the PCI4410A device has integrated all of these pullup resistors on the terminals shown in Table 3–2, except for the CCLKRUN/WP(IOIS16) pullup resistor.

Table 3–2. Integrated Pullup Resistors

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

† This terminal requires pullup, but the PCI4410A 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 SPKR. This terminal also is used in CardBus binary audio applications, and is referred to as CAUDIO. SPKR passes a TTL-level digital audio signal to the PCI4410A device. 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 PCI4410A device to produce SPKROUT. This output is enabled by bit 1 (SPKROUTEN) in the card control register (PCI offset 91h, see Section 4.34).

Older controllers support CAUDIO in binary or PWM mode, but use the same terminal (SPKROUT). Some audio chips may not support both modes on one terminal and may have a separate terminal for binary and PWM. The PCI4410A 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.

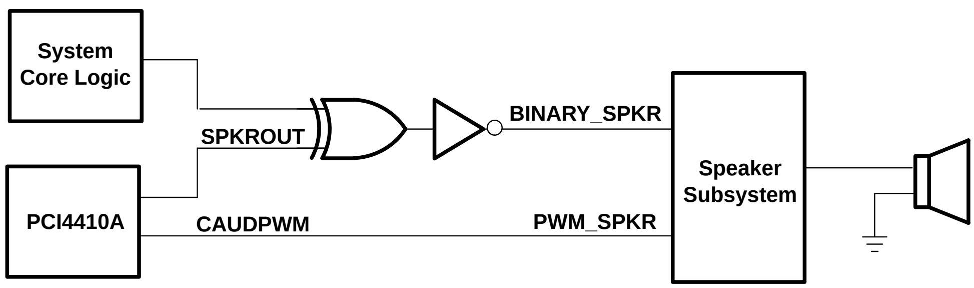


Figure 3–6. Sample Application of SPKROUT and CAUDPWM

3.5.9 LED Socket Activity Indicators

The socket activity LEDs indicate when a PC Card is being accessed. The LED_SKT signal can be routed to the multifunction terminals and also is 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/IREQ is low. For CardBus cards, the LED activity signal is pulsed if CFRAME, CIRDY, or CREQ is active.

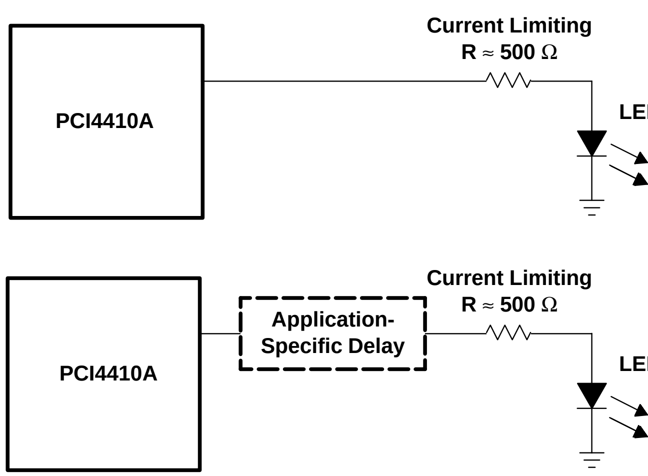


Figure 3–7. Two Sample LED Circuits

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 SUSPEND signal is asserted, when the PCI clock is to be stopped during the clock run protocol, or in the D2 or D1 power state.

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

3.5.10 PC Card-16 Distributed DMA Support

The PCI4410A device 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 3–3 provides the DDMA register configuration.

Two socket function-dependent PCI configuration header registers that are critical for DDMA are the socket DMA register 0 (PCI offset 94h, see Section 4.37) and the socket DMA register 1 (PCI offset 98h, 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 3–3. Distributed DMA Registers

<table>
<tr><th>TYPE</th><th colspan="4">REGISTER NAME</th><th>DDMA
BASE ADDRESS
OFFSET</th></tr>
<tr><td>R</td><td rowspan="2">Reserved</td><td rowspan="2">Page</td><td colspan="2">Current address</td><td rowspan="2">00h</td></tr>
<tr><td>W</td><td colspan="2">Base address</td></tr>
<tr><td>R</td><td rowspan="2">Reserved</td><td rowspan="2">Reserved</td><td colspan="2">Current count</td><td rowspan="2">04h</td></tr>
<tr><td>W</td><td colspan="2">Base count</td></tr>
<tr><td>R</td><td>N/A</td><td rowspan="2">Reserved</td><td>N/A</td><td>Status</td><td rowspan="2">08h</td></tr>
<tr><td>W</td><td>Mode</td><td>Request</td><td>Command</td></tr>
<tr><td>R</td><td>Multichannel</td><td rowspan="2">Reserved</td><td>N/A</td><td rowspan="2">Reserved</td><td rowspan="2">0Ch</td></tr>
<tr><td>W</td><td>Mask</td><td>Master clear</td></tr>
</table>

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 PCI4410A device implements these obsolete register bits as read-only, nonfunctional bits. The reserved registers shown in Table 3–3 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 PCI4410A device 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 PCI4410A device 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 PCI4410A device 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 PCI4410A device 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 PCI4410A device asserts $\overline{\text{REQ}}$ to acquire the PCI bus. Once the bus is granted in an idle state, the PCI4410A device 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 on to the PC Card. After terminating the PC Card cycle, the PCI4410A device requests access to the PCI bus again, until the transfer count has expired.

The PCI4410A 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 PCI4410A device 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 PCI4410A device supports demand mode transfers. The PCI4410A device 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 chipsets provide a way for legacy I/O devices to do DMA transfers on the PCI bus. In the PC/PCI DMA protocol, the PCI4410A device acts as a PCI target device to certain DMA-related I/O addresses. The PCI4410A $\overline{\text{PCREQ}}$ 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 PCI4410A device) 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 (PCI offset 80h, 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 3–4.

Table 3–4. 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 (PCI offset 94h, 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 PCI4410A device decodes the I/O addresses listed in Table 3–5 and performs actions dependent upon the address.

Table 3–5. 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 often is referred to as centralized DMA for this reason.

3.5.12 CardBus Socket Registers

The PCI4410A device contains all registers for compatibility with the *PC Card Standard*. These registers exist as the CardBus socket registers and are listed in Table 3–6.

Table 3–6. 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 PCI4410A device 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^2C. See Section 3.5.2, *P^2C Power-Switch Interface (TPS2211)*, for details. The PCI4410A serial bus interface is compatible with various I^2C and SMBus components.

3.6.1 Serial Bus-Interface Implementation

The PCI4410A device 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 PCI4410A device implements a two-pin serial interface with one clock signal (SCL) and one data signal (SDA). When pullup resistors are provided on the $\overline{\text{VCCD0}}$ and $\overline{\text{VCCD1}}$ terminals, the SCL signal is mapped to the MFUNC4 terminal and the SDA signal is mapped to the MFUNC1 terminal. The PCI4410A device drives SCL at nearly 100 kHz during data transfers, which is the maximum specified frequency for standard-mode I^2C. The serial EEPROM must be located at address A0h. Figure 3–8 illustrates an example application implementing the two-wire serial bus.

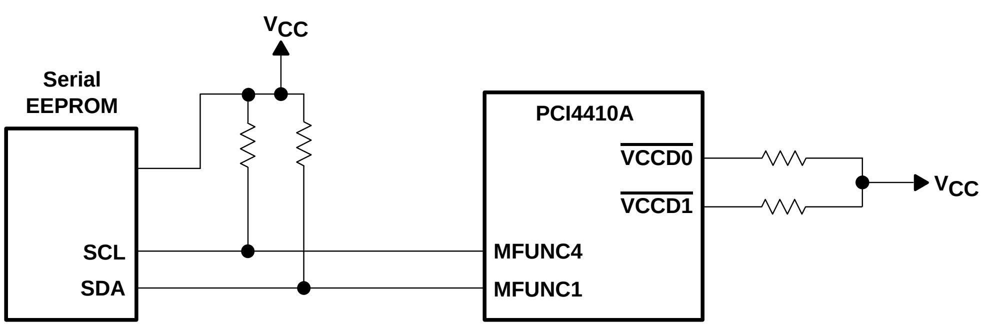


Figure 3–8. Serial EEPROM Application

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 PCI4410A device supports up to 100 Kb/s data transfer rate and is compatible with standard-mode I^2C 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 signaled 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, because changes on the SDA signal during the high state of SCL are interpreted as control signals, that is, a start or a stop condition.

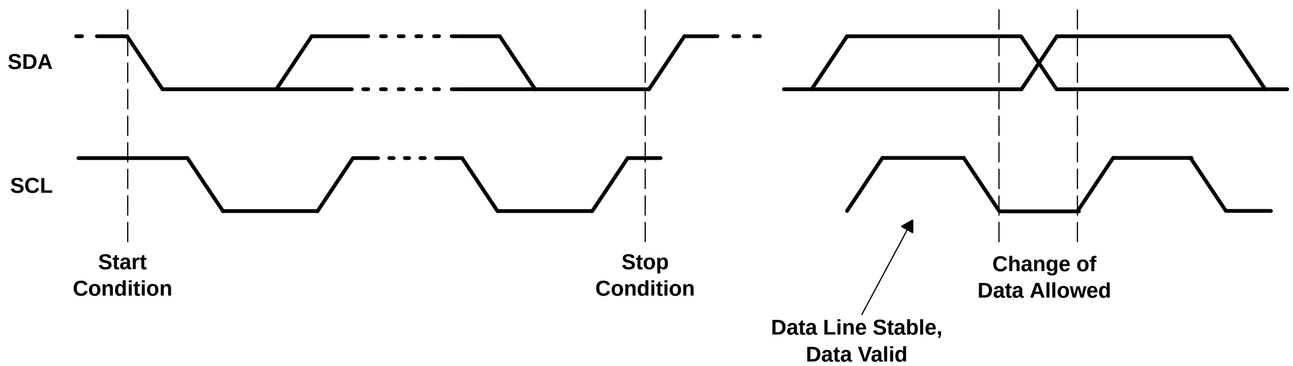


Figure 3–9. Serial Bus Start/Stop Conditions and Bit Transfers

Data is transferred serially in 8-bit bytes. The number of bytes that can 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.

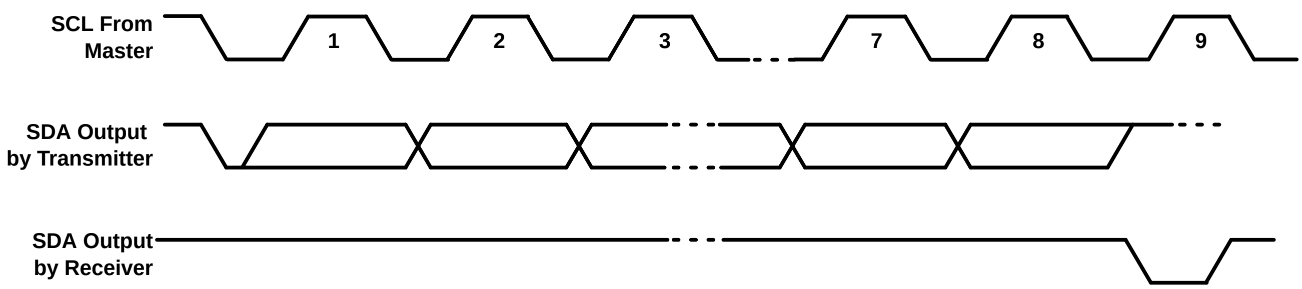


Figure 3–10. Serial Bus-Protocol Acknowledge

The PCI4410A device is a serial bus master; all other devices connected to the serial bus external to the PCI4410A device are slave devices. As the bus master, the PCI4410A device 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 PCI4410A device 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 PCI4410A device automatically loads the subsystem identification and other register defaults through a serial bus EEPROM.

Figure 3–11 illustrates a byte write. The PCI4410A device issues a start condition and sends the 7-bit slave device address and the command bit 0. 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 PCI4410A device and another slave acknowledgment is expected. The PCI4410A device then delivers the data byte, MSB first, and expects a final acknowledgment before issuing the stop condition.

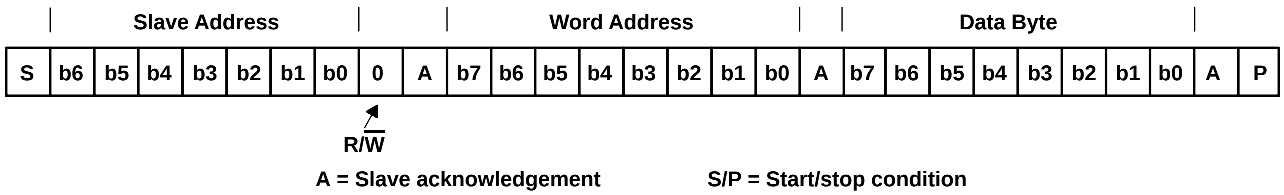


Figure 3–11. Serial Bus Protocol – Byte Write

Figure 3–12 illustrates a byte read. The read protocol is very similar to the write protocol, except the R/W command bit must be set to 1 to indicate a read-data transfer. In addition, the PCI4410A 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 PCI4410A master.

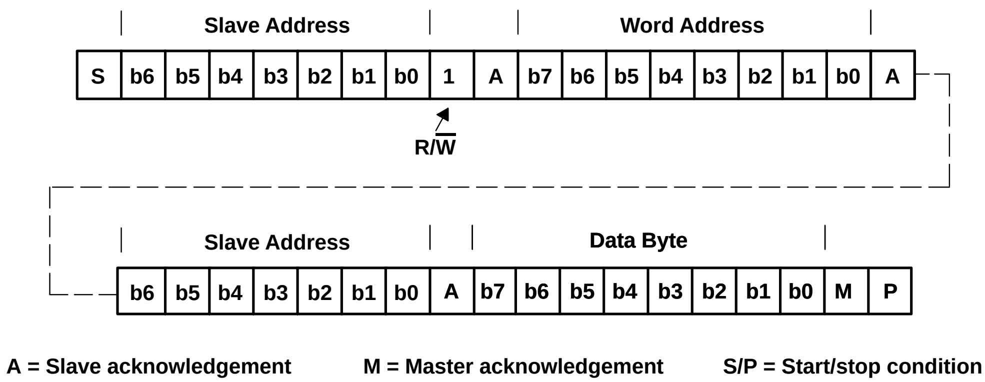


Figure 3–12. Serial Bus Protocol – Byte Read

Figure 3–13 illustrates EEPROM interface doubleword data-collection protocol.

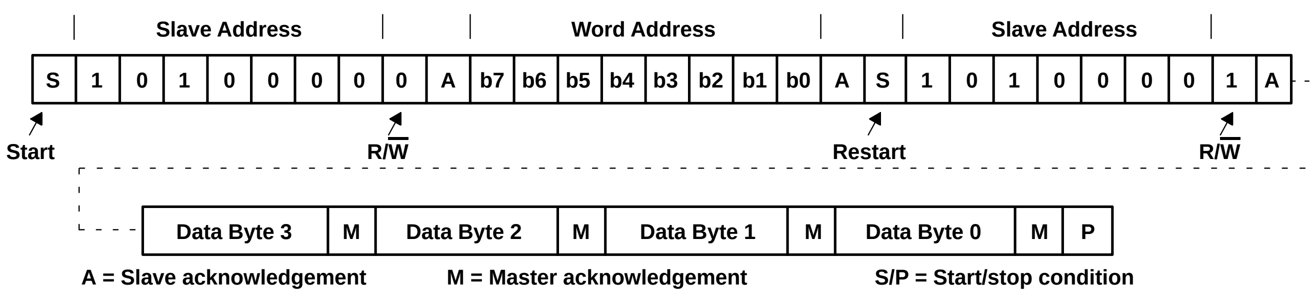


Figure 3–13. EEPROM Interface Doubleword Data Collection

3.6.3 Serial Bus EEPROM Application

When the PCI bus is reset and the serial bus interface is detected, the PCI4410A device 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 3–7.

Table 3–7. 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 Section 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 PCI4410A device to properly load initializations from a serial EEPROM.

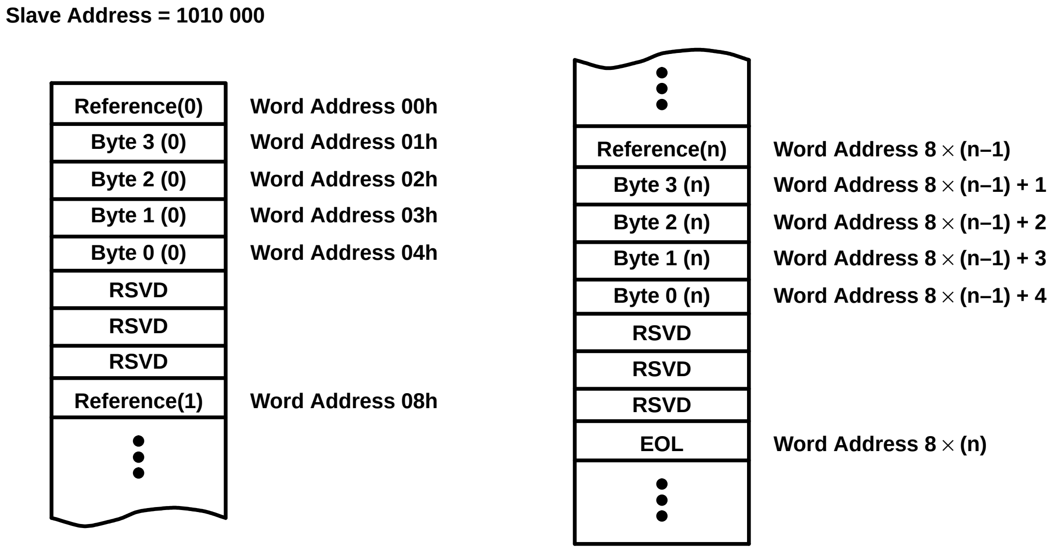


Figure 3–14. EEPROM Data Format

The byte at the EEPROM word address 00h must contain either a valid offset reference, as listed in Table 3–7, or an end-of-list (EOL) indicator. The EOL indicator has 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 101 0000b by the PCI4410A device. 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 PCI4410A device continues to load data from the serial EEPROM until an end-of-list indicator is read. Three reserved bytes are stuffed to maintain 8-byte data structures.

Note, the 8-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, the registers may be loaded incorrectly.

3.6.4 Accessing Serial Bus Devices Through Software

The PCI4410A device 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 PCI4410A device. The PCI4410A device 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 PCI4410A device is, therefore, backward compatible with existing interrupt control register definitions, and new registers have been defined where required.

The PCI4410A device 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 PCI4410A device, PC Card interrupts are classified as either card status change (CSC) or as functional interrupts.

The method by which any type of PCI4410A interrupt is communicated to the host interrupt controller varies from system to system. The PCI4410A device 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 PCI4410A device, 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 3–8 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 3–8. 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 3–9 describes the PC Card interrupt events.

Table 3–9. 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 *PC Card Standard* describes the power-up sequence that must be followed by the PCI4410A device 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 PCI4410A interrupt scheme can be used to notify the host system (see Table 3–9), denoted by the power cycle complete event. This interrupt source is considered a PCI4410A 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 3–9 by setting the appropriate bits in the PCI4410A device. By individually masking the interrupt sources listed, software can control those events that cause a PCI4410A interrupt. Host software has some control over the system interrupt the PCI4410A device asserts by programming the appropriate routing registers. The PCI4410A device 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 PCI4410A device, the interrupt service routine must determine which of the events listed in Table 3–8 caused the interrupt. Internal registers in the PCI4410A device 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 3–8 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 PCI4410A device 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 never should be a card interrupt that does not require service after proper initialization.

Table 3–8 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 (CardBus offset 00h, 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 PCI4410A device 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 (PCI offset 92h, see Section 4.35) 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 $\overline{\text{INTA}}$ also must be routed to the programmable interrupt controller (PIC), or to some circuitry that provides parallel PCI interrupts to the host.

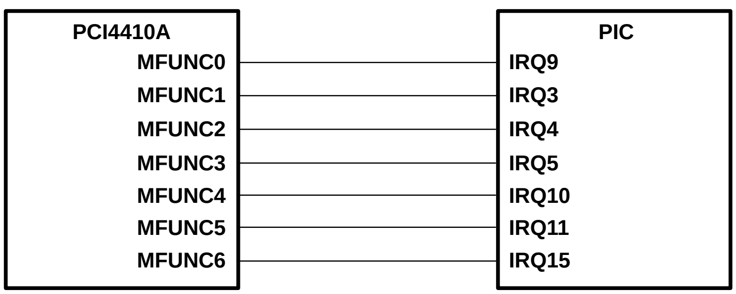


Figure 3–15. IRQ Implementation

Power-on software is responsible for programming the multifunction routing register to reflect the IRQ configuration of a system implementing the PCI4410A device. 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 PCI4410A device 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 PCI4410A device 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 PCI4410A Device

The PCI4410A device 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 PCI4410A device, when enabled, after a write cycle to either the socket control register (CardBus offset 10h, see Section 6.5) of the CardBus register set or the ExCA power control register (ExCA offset 02h, see Section 5.3).

The SMI control is programmed through three bits in the system control register (PCI offset 80h, see Section 4.29). These bits are SMIROUTE (bit 26), SMISTATUS (bit 25), and SMIENB (bit 24). Table 3–10 describes the SMI control bits function.

Table 3–10. SMI Control

BIT NAME	FUNCTION
SMIROUTE	This shared bit controls whether the SMI interrupts are sent as a CSC interrupt or as IRQ2.
SMISTATUS	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, the SMI interrupt is sent as the CSC. The CSC interrupt can be either level or edge mode, depending upon bit 1 (CSCMODE) in the ExCA global control register (ExCA offset 1Eh, see Section 5.22).

If IRQ2 is selected by SMIROUTE, 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 (PCI offset 8Ch, see Section 4.32).

3.8 Power-Management Overview

In addition to the low-power CMOS technology process used for the PCI4410A device, 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 PCI4410A device. $\overline{\text{CLKRUN}}$ 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 PCI4410A device 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 (PCI offset 80h, see Section 4.29) is set.
- The PC Card-16 resource manager is busy.
- The PCI4410A CardBus master state machine is busy. A cycle may be in progress on CardBus.
- The PCI4410A master is busy. There may be posted data from CardBus to PCI in the PCI4410A device.
- Interrupts are pending.
- The CardBus CCLK for either socket has not been stopped by the PCI4410A $\overline{\text{CCLKRUN}}$ manager.

The PCI4410A device 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 PCI4410A device 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

Bit 7 (COE) in the ExCA power control register (ExCA offset 02h, see Section 5.3) and bit 0 (PWRDWN) in the ExCA global control register (ExCA offset 1Eh, Section 5.22) 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 SUSPEND signal, provided for backward compatibility, gates the PRST (PCI reset) signal and the GRST (global reset) signal from the PCI4410A device. Besides gating PRST and GRST, SUSPEND also gates PCLK inside the PCI4410A device to minimize power consumption.

Gating PCLK does not create any issues with respect to the power switch interface in the PCI4410A device. This is because the PCI4410A device 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 PCI4410A device:

- Use an external clock to the PCI4410A CLOCK terminal
- Use the internal oscillator

It also should be noted that asynchronous signals, such as card status change interrupts and 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, the PCI clock must be restarted to pass the interrupt, because neither the internal oscillator nor an external clock is routed to the serial interrupt state machine. Figure 3–16 shows the suspend functional implementation diagram.

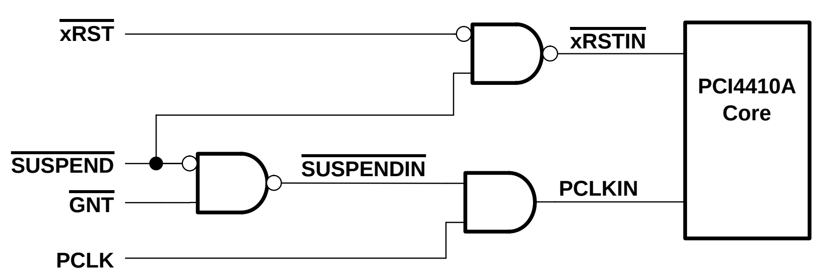


Figure 3–16. Suspend Functional Implementation

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

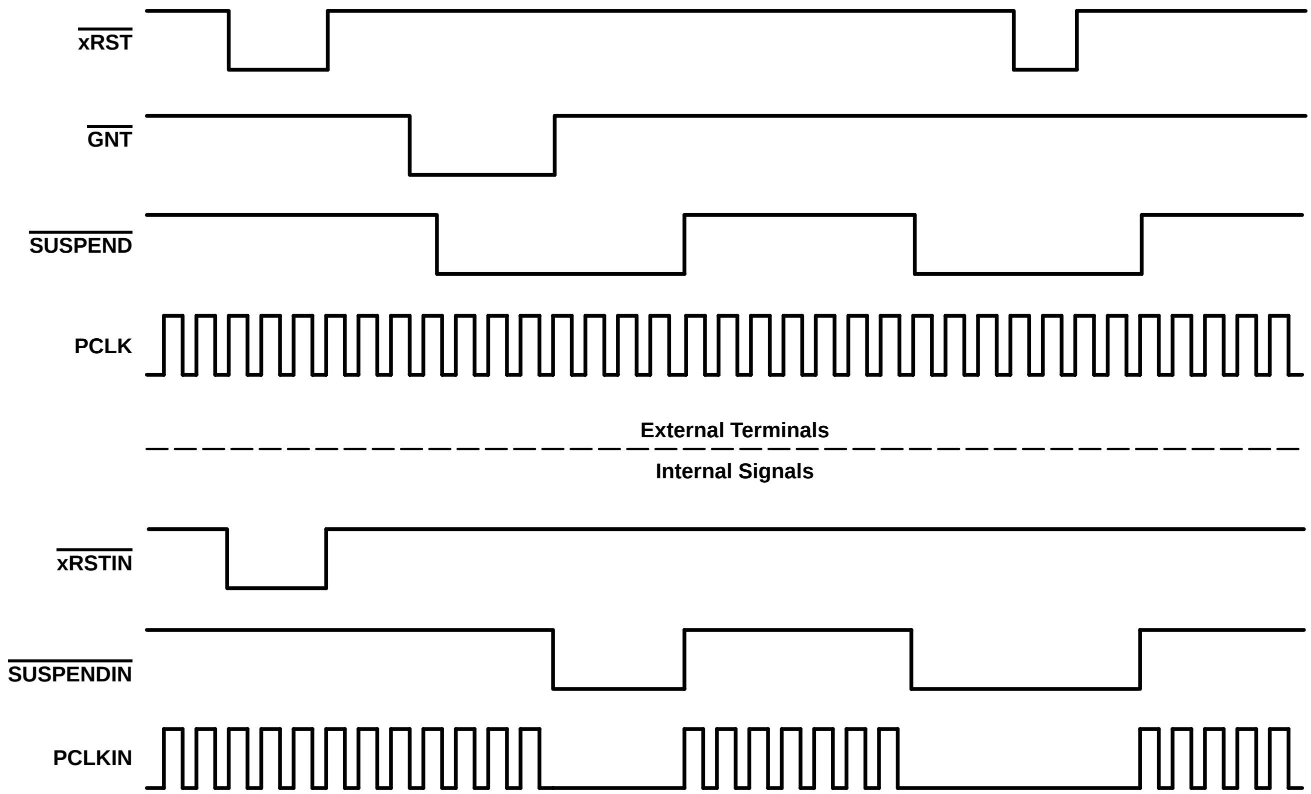


Figure 3–17. Signal Diagram of Suspend Function

3.8.5 Requirements for Suspend Mode

The suspend mode prevents the clearing of all register contents on the assertion of reset (PRST or GRST) that would require the reconfiguration of the PCI4410A device by software. Asserting the 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 currently is in process (GNT is asserted). It is important that the PCI bus not be parked on the PCI4410A device when SUSPEND is asserted, because the outputs are in a high-impedance state.

The GPIOs, MFUNC signals, and RI_OUT signals are all active during SUSPEND, unless they are disabled in the appropriate PCI4410A registers.

3.8.6 Ring Indicate

The 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. RI_OUT on the PCI4410A device can be asserted under any of the following conditions:

- A 16-bit PC Card modem in a powered socket asserts 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 PCI4410A RI_OUT function; however, it does not show the masking of CSC events. See Table 3–8 for a detailed description of CSC interrupt masks and flags.

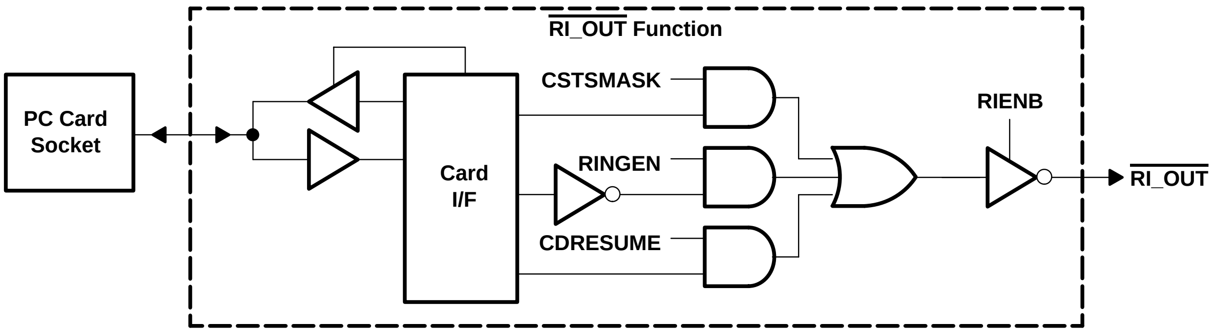


Figure 3–18. RI_OUT Functional Diagram

RI from the 16-bit PC Card interface is masked by bit 7 (RINGEN) in the ExCA interrupt and general control register (ExCA offset 03h, see Section 5.4). This is programmed on a per-socket basis and is applicable only when a 16-bit card is powered in the socket.

The CBWAKE, signaling to 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 (CardBus offset 04h, 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 for the operating system to 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 must 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 (PCI offset 06h, see Section 4.5).

The capabilities pointer provides access to the first item in the linked list of capabilities. For the PCI4410A device, 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, 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 3–11.

Table 3–11. 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 (PCI offset A2h, 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 (PCI 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 $\overline{\text{PME}}$ 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 $\overline{\text{PME}}$ context registers.
- Power source in $D3_{cold}$ if wake-up support is required from this state.

The Texas Instruments PCI4410A device addresses these D3 wake-up issues in the following manner:

- Two resets are provided to handle preservation of $\overline{\text{PME}}$ context bits:
  - Global reset ($\overline{\text{GRST}}$) is used only on the initial boot up of the system after power up. It places the PCI4410A device in its default state and requires BIOS to configure the device before becoming fully functional.
  - PCI reset ($\overline{\text{PRST}}$) now has dual functionality based on whether $\overline{\text{PME}}$ is enabled or not. If $\overline{\text{PME}}$ is enabled, $\overline{\text{PME}}$ context is preserved. If $\overline{\text{PME}}$ is not enabled, $\overline{\text{PRST}}$ acts the same as a normal PCI reset. Please see the master list of $\overline{\text{PME}}$ 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 PCI4410A 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 PCI4410A device offers a generic interface that is compliant with ACPI design rules.

Two doublewords of general-purpose ACPI programming bits reside in PCI4410A 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 (PCI offset A8h, see Section 4.45) and general-purpose event enable (PCI offset AAh, 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 on ACPI, see the *Advanced Configuration and Power Interface (ACPI) Specification.*

3.8.10 Master List of PME Context Bits and Global Reset-Only Bits

If the PME enable bit (PCI offset A4h, bit 8) is asserted, the assertion of PRST will not clear the following PME context bits. If the PME enable bit is not asserted, the PME context bits are cleared with PRST. The 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 PME enable bit. The GRST signal is gated only by the SUSPEND signal. This means that assertion of SUSPEND blocks the GRST signal internally, thus preserving all register contents. The registers cleared by 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 PCI4410A PCI configuration registers that make up the 256-byte PCI configuration header for each PCI4410A 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 PCI4410A device 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)

<table>
<tr><th colspan="4">REGISTER NAME</th><th>OFFSET</th></tr>
<tr><td colspan="2">Device ID</td><td colspan="2">Vendor ID</td><td>00h</td></tr>
<tr><td colspan="2">Status</td><td colspan="2">Command</td><td>04h</td></tr>
<tr><td colspan="3">PCI class code</td><td>Revision ID</td><td>08h</td></tr>
<tr><td>BIST</td><td>Header type</td><td>Latency timer</td><td>Cache line size</td><td>0Ch</td></tr>
<tr><td colspan="4">CardBus socket/ExCA base address</td><td>10h</td></tr>
<tr><td colspan="2">Secondary status</td><td>Reserved</td><td>Capability pointer</td><td>14h</td></tr>
<tr><td>CardBus latency timer</td><td>Subordinate bus number</td><td>CardBus bus number</td><td>PCI bus number</td><td>18h</td></tr>
<tr><td colspan="4">CardBus memory base register 0</td><td>1Ch</td></tr>
<tr><td colspan="4">CardBus memory limit register 0</td><td>20h</td></tr>
<tr><td colspan="4">CardBus memory base register 1</td><td>24h</td></tr>
<tr><td colspan="4">CardBus memory limit register 1</td><td>28h</td></tr>
<tr><td colspan="4">CardBus I/O base register 0</td><td>2Ch</td></tr>
<tr><td colspan="4">CardBus I/O limit register 0</td><td>30h</td></tr>
<tr><td colspan="4">CardBus I/O base register 1</td><td>34h</td></tr>
<tr><td colspan="4">CardBus I/O limit register 1</td><td>38h</td></tr>
<tr><td colspan="2">Bridge control</td><td>Interrupt pin</td><td>Interrupt line</td><td>3Ch</td></tr>
<tr><td colspan="2">Subsystem ID</td><td colspan="2">Subsystem vendor ID</td><td>40h</td></tr>
<tr><td colspan="4">PC Card 16-bit I/F legacy-mode base address</td><td>44h</td></tr>
<tr><td colspan="4">Reserved</td><td>48h–7Ch</td></tr>
<tr><td colspan="4">System control</td><td>80h</td></tr>
<tr><td>Reserved</td><td>General control</td><td>General status</td><td>Reserved</td><td>84h</td></tr>
<tr><td colspan="4">Reserved</td><td>88h–8Bh</td></tr>
<tr><td colspan="4">Multifunction routing</td><td>8Ch</td></tr>
<tr><td>Diagnostic</td><td>Device control</td><td>Card control</td><td>Retry status</td><td>90h</td></tr>
<tr><td colspan="4">Socket DMA register 0</td><td>94h</td></tr>
<tr><td colspan="4">Socket DMA register 1</td><td>98h</td></tr>
<tr><td colspan="4">Reserved</td><td>9Ch</td></tr>
<tr><td colspan="2">Power management capabilities</td><td>Next-item pointer</td><td>Capability ID</td><td>A0h</td></tr>
<tr><td>Power management data</td><td>Power management control/status register bridge support extensions</td><td colspan="2">Power management control/status</td><td>A4h</td></tr>
<tr><td colspan="2">General-purpose event enable</td><td colspan="2">General-purpose event status</td><td>A8h</td></tr>
<tr><td colspan="2">General-purpose output</td><td colspan="2">General-purpose input</td><td>ACh</td></tr>
<tr><td colspan="4">Reserved</td><td>B0h–FCh</td></tr>
</table>

A bit description table, typically included when a register contains bits of more than one type or purpose, indicates bit field names, a detailed field description, and field access tags, which appear in the type column of the bit-description table. Table 4–2 describes the field access tags.

Table 4–2. Bit-Field Access Tag Descriptions

ACCESS TAG	NAME	MEANING
R	Read	Field can be read by software.†
W	Write	Field can be written by software to any value.
S	Set	Field can be set by a write of 1. Writes of 0 have no effect.
C	Clear	Field can be cleared by a write of 1. Writes of 0 have no effect.
U	Update	Field can be autonomously updated by the PCI4410A device.

† A bit can display either of two types of behavior when read. After having been read, it can maintain the value it had previously, or the read process can cause it to be reset to 0.

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 PCI4410A device by TI. The device identification for the PCI4410A device 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 PCI4410A interface to the PCI bus. All bit functions adhere to the definitions in *PCI Local Bus Specification*. 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	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 4–3. Command Register Description

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 PCI4410A device does not generate fast back-to-back transactions; therefore, bit 9 returns 0 when read.
8	SERR_EN	R/W	System error (SERR) enable. Bit 8 controls the enable for the SERR driver on the PCI interface. SERR can be asserted after detecting an address parity error on the PCI bus. Both bits 8 and 6 must be set for the PCI4410A device to report address parity errors. 0 = Disable SERR output driver (default) 1 = Enable SERR output driver
7	STEP_EN	R	Address/data stepping control. The PCI4410A device 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 PCI4410A device's response to parity errors through PERR. Data parity errors are indicated by asserting PERR; address parity errors are indicated by asserting SERR. 0 = PCI4410A device ignores detected parity error (default). 1 = PCI4410A device 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 PCI4410A device does not respond to palette register writes and snoops the data). When bit 5 is 0, the PCI4410A device 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 PCI4410A 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 PCI4410A device 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 PCI4410A device can act as a PCI bus initiator (master). The PCI4410A device can take control of the PCI bus only when this bit is set. 0 = Disables the PCI4410A device's ability to generate PCI bus accesses (default). 1 = Enables the PCI4410A device's ability to generate PCI bus accesses.
1	MEM_EN	R/W	Memory space enable. Bit 1 controls whether or not the PCI4410A device can claim cycles in PCI memory space. 0 = Disables the PCI4410A device's response to memory space accesses (default). 1 = Enables the PCI4410A device's response to memory space accesses.
0	IO_EN	R/W	I/O space control. Bit 0 controls whether or not the PCI4410A device can claim cycles in PCI I/O space. 0 = Disables the PCI4410A device's response to I/O space accesses (default). 1 = Enables the PCI4410A device's response to I/O space accesses.

4.5 Status Register

The status register provides device information to the host system. Bits in this register can 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 4–4 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/Clear
Offset: 06h
Default: 0210h

Table 4–4. Status Register Description

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 PCI4410A device signals a system error to the host.
13	MABORT	R/C	Received master abort. Bit 13 is set when a cycle initiated by the PCI4410A device on the PCI bus is terminated by a master abort.
12	TABT_REC	R/C	Received target abort. Bit 12 is set when a cycle initiated by the PCI4410A device on the PCI bus is terminated by a target abort.
11	TABT_SIG	R/C	Signaled target abort. Bit 11 is set by the PCI4410A device 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 PCI4410A device 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 PCI4410A device. b. The PCI4410A device was the bus master during the data parity error. c. Bit 6 (PERR_EN) in the command register (PCI offset 04h, see Section 4.4) is set.
7	FBB_CAP	R	Fast back-to-back capable. The PCI4410A device cannot accept fast back-to-back transactions; therefore, bit 7 is hardwired to 0.
6	UDF	R	User-definable feature support. The PCI4410A device does not support the user-definable features; therefore, bit 6 is hardwired to 0.
5	66MHZ	R	66-MHz capable. The PCI4410A device 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 PCI4410A device.

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	1	0

Register: **Revision ID**
Type: Read-only
Offset: 08h
Default: 02h

4.7 PCI Class Code Register

The class code register recognizes the PCI4410A device 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: 06 0700h

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 PCI4410A device in units of PCI clock cycles. When the PCI4410A device is a PCI bus initiator and asserts $\overline{\text{FRAME}}$, the latency timer begins counting from zero. If the latency timer expires before the PCI4410A transaction has terminated, the PCI4410A device 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 PCI4410A 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 PCI4410A device 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 4 Kbytes 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 status register (PCI offset 06h); status bits are cleared by writing a 1. 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	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/Clear
Offset: 16h
Default: 0200h

Table 4–5. Secondary Status Register Description

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 CSERR is signaled by a CardBus card. The PCI4410A device does not assert CSERR.
13	CBMABORT	R/C	Received master abort. Bit 13 is set when a cycle initiated by the PCI4410A device on the CardBus bus is terminated by a master abort.
12	REC_CBTA	R/C	Received target abort. Bit 12 is set when a cycle initiated by the PCI4410A device on the CardBus bus is terminated by a target abort.
11	SIG_CBTA	R/C	Signaled target abort. Bit 11 is set by the PCI4410A device when it terminates a transaction on the CardBus bus with a target abort.
10–9	CB_SPEED	R	CDEVSEL timing. These bits encode the timing of CDEVSEL and are hardwired 01b, indicating that the PCI4410A device 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. CPERR was asserted on the CardBus interface. b. The PCI4410A device was the bus master during the data parity error. c. Bit 0 (CPERREN) in the bridge control register (PCI offset 3Eh, see Section 4.25) is set.
7	CBFBB_CAP	R	Fast back-to-back capable. The PCI4410A device cannot accept fast back-to-back transactions; therefore, bit 7 is hardwired to 0.
6	CB_UDF	R	User-definable feature support. The PCI4410A device does not support the user-definable features; therefore, bit 6 is hardwired to 0.
5	CB66MHZ	R	66-MHz capable. The PCI4410A 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 PCI4410A device is connected. The PCI4410A device uses this register in conjunction with the CardBus bus number (PCI offset 19h, see Section 4.16) and subordinate bus number (PCI offset 1Ah, see Section 4.17) 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 PCI4410A device is connected. The PCI4410A device uses this register in conjunction with the PCI bus number (PCI offset 18h, see Section 4.15) and subordinate bus number (PCI offset 1Ah, see Section 4.17) 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 PCI4410A device uses this register in conjunction with the PCI bus number (PCI offset 18h, see Section 4.15) and CardBus bus number (PCI offset 19h, see Section 4.16) 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 PCI4410A CardBus interface in units of CCLK cycles. When the PCI4410A device is a CardBus initiator and asserts $\overline{\text{CFRAME}}$, the CardBus latency timer begins counting. If the latency timer expires before the PCI4410A transaction has terminated, the PCI4410A device 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 PCI4410A device 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 (PCI offset 3Eh, 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 PCI4410A device to claim any memory transactions through CardBus memory windows (that is, these windows are not enabled by default to pass the first 4 Kbytes 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 PCI4410A device 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 (PCI offset 3Eh, 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 PCI4410A device to claim any memory transactions through CardBus memory windows (that is, these windows are not enabled by default to pass the first 4 Kbytes 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 PCI4410A device 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 that 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 PCI4410A device 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 register) 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 PCI4410A device 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 (PCI offset 92h, see Section 4.35). The PCI4410A device 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 PCI4410A bridging functions. See Table 4–6 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	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 4–6. Bridge Control Register Description

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 cannot.
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 are routed to PCI interrupts (default). 1 = Functional interrupts are routed to IRQ interrupts.
6	CRST	R/W	CardBus reset. When bit 6 is set, $\overline{\text{CRST}}$ is asserted on the CardBus interface. $\overline{\text{CRST}}$ also can be asserted by passing a $\overline{\text{PRST}}$ assertion to CardBus. 0 = $\overline{\text{CRST}}$ is deasserted. 1 = $\overline{\text{CRST}}$ is asserted (default).
5	MABTMODE	R/W	Master abort mode. Bit 5 controls how the PCI4410A device responds to a master abort when the PCI4410A device is an initiator on the CardBus interface. 0 = Master aborts are 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 PCI4410A device 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 PCI4410A device passes I/O cycles within the 64-Kbyte ISA range. This bit is not common between sockets. When this bit is set, the PCI4410A device 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 PCI4410A device 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 PCI4410A device 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 (PCI offset 80h, 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 (PCI offset 80h, 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 PCI4410A device 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 4–7 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	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/Clear
Offset: 80h
Default: 0044 9060h

Table 4–7. System Control Register Description

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 PCI4410A functions. 00 = $\overline{INTA}$/$\overline{INTB}$ signal in $\overline{INTA}$/$\overline{INTB}$ slots (default) 01 = $\overline{INTA}$/$\overline{INTB}$ signal in $\overline{INTB}$/$\overline{INTC}$ slots 10 = $\overline{INTA}$/$\overline{INTB}$ signal in $\overline{INTC}$/$\overline{INTD}$ slots 11 = INTA/INTB signal in INTD/INTA slots
29	TIE_INTB_INTA	R/W	Tie $\overline{INTB}$ to $\overline{INTA}$. When bit 29 is set to 1, $\overline{INTB}$ is tied to $\overline{INTA}$ (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 is disabled (default). 1 = Internal oscillator is 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 are 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 is signaled (default). 1 = SMI interrupt is 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 bits 2–0 (VERSION field) in the power management capability register (PCI offset A2h, Section 4.41) 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 is enabled for 16-bit cards (default). 1 = V_{CC} protection is 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 is disabled (default). 1 = Reduced zoomed video is enabled.
19	CDREQEN	R/W	PC/PCI DMA card enable. When bit 19 is set, the PCI4410A device allows 16-bit PC Cards to request PC/PCI DMA using the $\overline{DREQ}$ signaling. $\overline{DREQ}$ is selected through the socket DMA register 0 (PCI offset 94h, see Section 4.37). 0 = Ignore $\overline{DREQ}$ signaling from PC Cards (default) 1 = Signal DMA request on $\overline{DREQ}$
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 4–7. System Control Register Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION
14	MRBURSTUP	R/W	Memory-read burst-enable upstream. When bit 14 is set, the PCI4410A device 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 is not in progress (default). 1 = Interrogation is in progress.
7	AUTOPWRSWEN	R/W	Auto power-switch enable. 0 = Bit 5 (AUTOPWRSWEN) in ExCA power control register (ExCA offset 02h, see Section 5.3) is disabled. (default). 1 = Bit 5 (AUTOPWRSWEN) in ExCA power control register (ExCA offset 02h, 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 is not clocked.
5	SUBSYSRW	R/W	Subsystem ID (PCI offset 42h, see Section 4.27), subsystem vendor ID (PCI offset 40h, see Section 4.26), ExCA identification and revision (ExCA offset 00h, 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 SERR signaling enable 0 = CardBus data parity error is not signaled on PCI SERR. 1 = CardBus data parity error is signaled on PCI 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 is disabled (default). 1 = Centralized DMA is 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 CLKRUN protocols. 0 = Allows normal functioning of both CLKRUN protocols (default) 1 = Does not allow CB clock or PCI clock to be stopped using the CLKRUN protocols
0	RIMUX	R/W	RI_OUT/PME multiplex enable. 0 = RI_OUT and PME are both routed to the RI_OUT/PME terminal. If both are enabled at the same time, RI_OUT has precedence over PME. 1 = Only PME is routed to the RI_OUT/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 4–8 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	R/C	R
Default	0	0	0	0	0	X	0	0

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

Table 4–8. General Status Register Description

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 $\overline{PRST}$ is low. When this bit is set, the serial ROM is detected. This status bit is encoded as: 0 = EEPROM is not detected (default). 1 = EEPROM is 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 is detected (default). 1 = Data error is detected.
0	EEBUSY	R	Serial EEPROM busy status. This bit indicates the status of the PCI4410A 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 4–9 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 4–9. General Control Register Description

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	R/W	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 can 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 can also be loaded through a serial bus EEPROM. See Table 4–10 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	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 4–10. Multifunction Routing Register Description

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 0100 = IRQ4 1000 = IRQ8 1100 = IRQ12 0001 = CLKRUN 0101 = IRQ5 1001 = IRQ9 1101 = IRQ13 0010 = IRQ2 0110 = IRQ6 1010 = IRQ10 1110 = IRQ14 0011 = IRQ3 0111 = IRQ7 1011 = IRQ11 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 0100 = IRQ4 1000 = CAUDPWM 1100 = LED_SKT 0001 = GPO4 0101 = D3_STAT 1001 = IRQ9 1101 = Diagnostic setup: OHCI test 0010 = PCGNT 0110 = ZVSTAT 1010 = IRQ10 1110 = GPE 0011 = IRQ3 0111 = ZVSEL0 1011 = IRQ11 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 VCCD0 and VCCD1 terminals, the MFUNC4 terminal provides the SCL signaling. 0000 = GPI3 0100 = IRQ4 1000 = CAUDPWM 1100 = RI_OUT 0001 = GPO3 0101 = IRQ5 1001 = IRQ9 1101 = LED_SKT 0010 = PCI LOCK 0110 = ZVSTAT 1010 = IRQ10 1110 = GPE 0011 = IRQ3 0111 = ZVSEL0 1011 = IRQ11 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 0100 = IRQ4 1000 = IRQ8 1100 = IRQ12 0001 = IRQSER 0101 = IRQ5 1001 = IRQ9 1101 = IRQ13 0010 = IRQ2 0110 = IRQ6 1010 = IRQ10 1110 = IRQ14 0011 = IRQ3 0111 = IRQ7 1011 = IRQ11 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 0100 = IRQ4 1000 = CAUDPWM 1100 = RI_OUT 0001 = GPO2 0101 = IRQ5 1001 = IRQ9 1101 = D3_STAT 0010 = PCREQ 0110 = ZVSTAT 1010 = IRQ10 1110 = GPE 0011 = IRQ3 0111 = ZVSEL0 1011 = IRQ11 1111 = IRQ7

Table 4–10. Multifunction Routing Register Description (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 VCCD0 and VCCD1 terminals, the MFUNC1 terminal provides the SDA signaling. 0000 = GPI1 0100 = IRQ4 1000 = CAUDPWM 1100 = LED_SKT 0001 = GPO1 0101 = IRQ5 1001 = IRQ9 1101 = IRQ13 0010 = D3_STAT 0110 = ZVSTAT 1010 = IRQ10 1110 = GPE 0011 = IRQ3 0111 = 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 = GPO0 0101 = IRQ5 1001 = IRQ9 1101 = IRQ13 0010 = INTA 0110 = ZVSTAT 1010 = IRQ10 1110 = GPE 0011 = IRQ3 0111 = 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 PCI4410A device 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 command, status, and bridge control registers by the PCI SIG. See Table 4–11 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/Clear
Offset: 90h
Default: C0h

Table 4–11. Retry Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	PCIRETRY	R/W	PCI retry timeout counter enable. Bit 7 is encoded: 0 = PCI retry counter is disabled. 1 = PCI retry counter is enabled (default).
6	CBRETRY	R/W	CardBus retry timeout counter enable. Bit 6 is encoded: 0 = CardBus retry counter is disabled. 1 = CardBus retry counter is 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. RI_OUT is enabled 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	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/Clear
Offset: 91h
Default: 00h

Table 4–12. Card Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	RIENB	R/W	Ring indicate output enable. 0 = Disables any routing of RI_OUT signal (default). 1 = Enables RI_OUT signal for routing to the RI_OUT/PME terminal, when bit 0 (RIMUX) in the system control register (PCI offset 80h, 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, 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, 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 = SPKR to SPKROUT is not enabled (default). 1 = SPKR to SPKROUT is 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 is detected (default). 1 = PC Card functional interrupt is 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 4–13 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 4–13. Device Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	SKTPWR_LOCK	R/W	Socket-power lock bit. When this bit is set to 1, software cannot 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 addition, 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 4–14 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: 21h

Table 4–14. Diagnostic Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	TRUE_VAL	R/W	This bit defaults to 0. This bit causes software to fail to recognize the PCI4410A device 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 are routed to PCI if ExCA 803 (see Section 5.4) bit 4 = 1. 1 = CSC interrupts are 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 (DREQ) signaling. See Table 4–15 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	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 4–15. Socket DMA Register 0 Description

BIT	SIGNAL	TYPE	FUNCTION
31–2	RSVD	R	Reserved. Bits 31–2 return 0s when read.
1–0	DREQPIN	R/W	DMA request (DREQ). Bits 1 and 0 indicate which pin on the 16-bit PC Card interface acts as DREQ during DMA transfers. This field is encoded as: 00 = Socket is not configured for DMA (default). 01 = DREQ uses SPKR. 10 = DREQ uses 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 4–16 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 4–16. Socket DMA Register 1 Description

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 PCI4410A device 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 PCI4410A 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 PCI4410A CardBus bridge functions support D0, D1, D2, and D3 power states. 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 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 4–17. Power Management Capabilities Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	PME_SUPPORT	R/W	$\overline{PME}$ support. This 5-bit field indicates the power states from which the PCI4410A functions can assert $\overline{PME}$. A 0 (zero) for any bit indicates that the function cannot assert the $\overline{PME}$ 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 $\overline{PME}$ 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, the BIOS should write a 0 to this bit.
14–11	PME_SUPPORT	R	Bit 14 contains the value 1, indicating that the $\overline{PME}$ signal can be asserted from $D3_{hot}$ state. Bit 13 contains the value 1, indicating that the $\overline{PME}$ signal can be asserted from D2 state. Bit 12 contains the value 1, indicating that the $\overline{PME}$ signal can be asserted from D1 state. Bit 11 contains the value 1, indicating that the $\overline{PME}$ 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 $\overline{PME}$ 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 PCI4410A device requires an auxiliary power supply, this bit returns 1.
3	PMECLK	R	$\overline{PME}$ clock. Bit 3 returns 0 when read, indicating that no host bus clock is required for the PCI4410A device to generate $\overline{PME}$.
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 bit 23 (PCIPMEN) in the system control register (PCI offset 80h, Section 4.29) for additional information. It is recommended that the PCIPMEN bit be set by BIOS. If PCIPMEN is set, bits 2–0 (VERSION field) will return 010b, indicating support for the *PCI Bus Power Management Interface Specification* (Revision 1.1).

4.42 Power Management Control/Status Register

The power management control/status register determines and changes the current power state of the PCI4410A 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 4–18 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/Clear
Offset: A4h
Default: 0000h

Table 4–18. Power Management Control/Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	PMESTAT	R/C	$\overline{PME}$ status. Bit 15 is set when the CardBus function normally would assert $\overline{PME}$, independent of the state of bit 8 (PME_EN). Bit 15 is cleared by a writeback of 1, and this also clears the $\overline{PME}$ signal if $\overline{PME}$ 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	$\overline{PME}$ enable. When set to 1, bit 8 enables the function to assert $\overline{PME}$. When cleared to 0, the assertion of $\overline{PME}$ is disabled.
7–5	RSVD	R	Reserved. Bits 7–5 return 0s when read.
4	DYN_DATA_PME_EN	R	Dynamic data $\overline{PME}$ enable. Bit 4 returns 0 when read, because the CardBus function does not report dynamic 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 4–19 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 4–19. Power Management Control/Status Register Bridge Support Extensions Description

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 (PCI offset A4h, 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_$\overline{\text{B3}}$	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 meaningful only 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 $\overline{GPE}$ are cleared by writing a 1 to the corresponding bit location. 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 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/Clear
Offset: A8h
Default: 0000h

Table 4–20. General-Purpose Event Status Register Description

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 (PCI offset 91h, 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 V_{PP} 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 $\overline{GPE}$ signal. The $\overline{GPE}$ signal is driven until the corresponding status bit is cleared and the event is serviced. The $\overline{GPE}$ can be signaled only if one of the multifunction terminals, MFUNC6–MFUNC0, is configured for $\overline{GPE}$ signaling. 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 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 4–21. General-Purpose Event Enable Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	ZV_EN	R/W	PC card socket ZV enable. When bit 15 is set, a $\overline{GPE}$ is signaled on a change in status of bit 6 (ZVENABLE) in the card control register (PCI offset 91h, 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 $\overline{GPE}$ 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 $\overline{GPE}$ 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 $\overline{GPE}$ 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 $\overline{GPE}$ 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 $\overline{GPE}$ 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 $\overline{GPE}$ 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 $\overline{GPE}$ 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 4–22 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 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 4–22. General-Purpose Input Register Description

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 4–23 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 4–23. General-Purpose Output Register Description

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	GPO3 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 PCI4410A device are register-compatible with the Intel 82365SL–DF PCMCIA controller. The 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 (PCI offset 44h, 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.

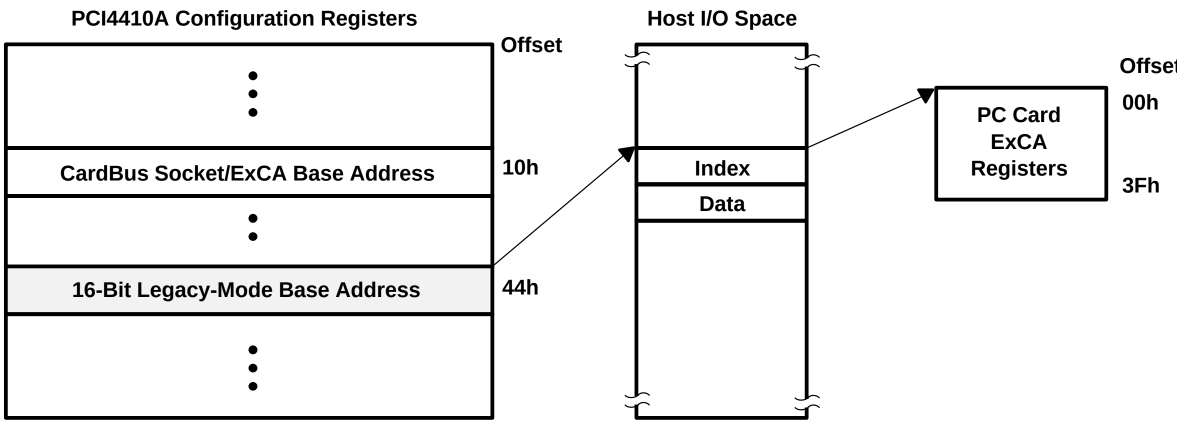


Figure 5–1. ExCA Register Access Through I/O

The TI PCI4410A device 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 (PCI offset 10h, 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.

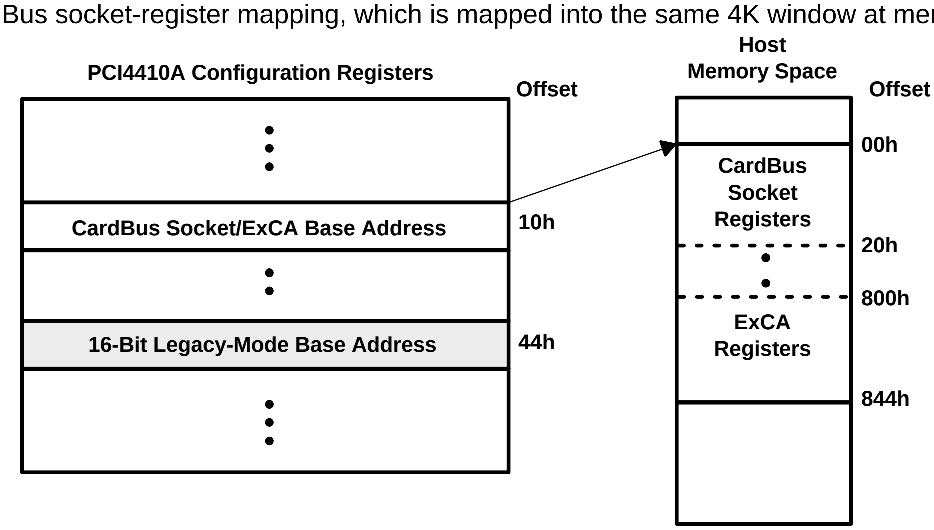


Figure 5–2. ExCA Register Access Through Memory

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 PCI4410A device to ensure that all possible PCI4410A interrupts potentially can 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 (ExCA offset 03h, see Section 5.4) and the ExCA card status-change-interrupt configuration register (ExCA offset 05h, 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 (PCI offset 80h, 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 Description

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 PCI4410A device. The PCI4410A device 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 PCI4410A device. 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 PCI4410A 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 Description

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 (ExCA offset 02h, 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 PCI4410A device whether or not the memory card is write protected. Furthermore, write protection for an entire PCI4410A 16-bit memory window is available by setting the appropriate bit in the ExCA memory window offset-address high-byte register (see Section 5.18). 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 can 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 can 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. The PCI4410A device supports both the 82365SL and 82365SL-DF register models. Bits 3–0 (365REV filed) of the ExCA identification and revision register (ExCA offset 00h, see Section 5.1) control which register model is supported. See Table 5–4 and Table 5–5 for a complete description of the register contents.

5.3.1 Intel 82365SL Support Mode

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

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 PCI4410A device. 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 (PCI offset 80h, 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 (PCI offset 80h, 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 PCI4410A device 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

5.3.2 Intel 82365SL-DF Support Mode

Bit	7	6	5	4	3	2	1	0
Name	ExCA power control							
Type	R/W	R	R	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 5–5. ExCA Power Control Register 82365SL-DF Support Description

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 PCI4410A device. 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 PCI4410A device 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 Description

BIT	SIGNAL	TYPE	FUNCTION
7	RINGEN	R/W	Card ring indicate enable. Bit 7 enables the ring indicate function of BVD1/$\overline{RI}$. 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 (ExCA offset 05h, 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 (ExCA offset 1Eh, 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 Description

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 is detected on either $\overline{CD1}$ or $\overline{CD2}$. 1 = Change is 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 PCI4410A 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 is 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 PCI4410A 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 PCI4410A interrupt was due to a battery-dead condition. This bit is encoded as: 0 = $\overline{STSCHG}$ is deasserted (default). 1 = $\overline{STSCHG}$ is asserted. Ring indicate. When an I/O card is installed in the socket and the PCI4410A device 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 Description

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 (PCI offset 93h, see Section 4.36) is set to 1. In this case bit 4 (CSCROUTE) of the ExCA interrupt and general control register is a "don't care" (ExCA offset 03h, 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 (PCI offset 93h, 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 (ExCA offset 03h, see Section 5.4) to 1. 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 condition (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 PCI4410A device 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 Description

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 is disabled (default). 1 = I/O window 1 is 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 is disabled (default). 1 = I/O window 0 is 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 is disabled (default). 1 = Memory window 4 is 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 is disabled (default). 1 = Memory window 3 is 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 is disabled (default). 1 = Memory window 2 is 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 is disabled (default). 1 = Memory window 1 is 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 is disabled (default). 1 = Memory window 0 is 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 Description

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 PCMCIA controller. 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 PCMCIA controller. 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 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 PCMCIA controller. 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 PCMCIA controller. 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 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

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

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

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

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

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

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

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 PCMCIA controller. 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

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

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

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

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

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

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 always is 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

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

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, and reports the status of VS1 and 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 Description

BIT	SIGNAL	TYPE	FUNCTION
7	VS2STAT	R	VS2 state. Bit 7 reports the current state of VS2 at the PC Card interface and, therefore, has no default value. 0 = VS2 low 1 = VS2 high
6	VS1STAT	R	VS1 state. Bit 6 reports the current state of VS1 at the PC Card interface and, therefore, has no default value. 0 = VS1 low 1 = VS1 high
5	SWCSC	R/W	Software card-detect interrupt. If bit 3 (CDEN) in the ExCA card status-change-interrupt configuration register (ExCA offset 05h, see Section 5.6) is set to 1, 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 (see Section 5.6) is cleared to 0, 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, once a card detect change has been detected on CD1 and CD2 inputs, RI_OUT goes from high to low. RI_OUT remains low until bit 0 (card status change) in the ExCA card status-change register (ExCA offset 04h, see Section 5.5) is cleared. If this bit is a 0, 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 Description

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 PCI4410A 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 (ExCA offset 04h, 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 PCI4410A 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 PCI4410A device is in power-down mode. In power-down mode, the PCI4410A 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 PCI4410A device 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 accessible only 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 *PC Card Standard* requires a CardBus socket controller to provide five 32-bit registers that report and control socket-specific functions. The PCI4410A device provides the CardBus socket/ExCA base-address register (PCI offset 10h, 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 PCI4410A device implements an additional register at offset 20h that provides power-management control for the socket.

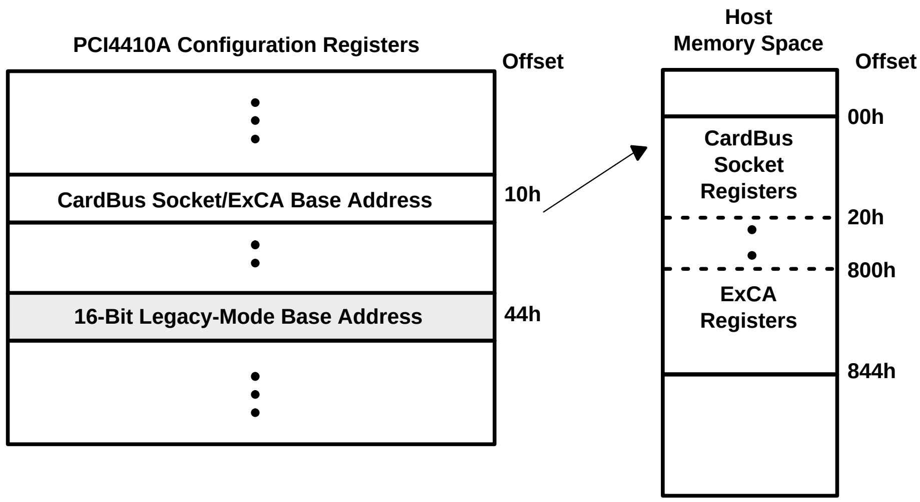


Figure 6–1. Accessing CardBus Socket Registers Through PCI Memory

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 (CardBus offset 08h, 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 (CardBus offset 0Ch, see Section 6.4). All bits in this register are cleared by PCI reset. They can be set again immediately if, when coming out of PC Card reset, the bridge finds the status unchanged (that is, CSTSCHG is reasserted or card detect still is true). Software must clear this register before enabling interrupts. If it is not cleared when interrupts are enabled, 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/Clear
Offset: CardBus socket address + 00h
Default: 0000 0000h

Table 6–2. Socket Event Register Description

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 PCI4410A device detects that bit 3 (PWRCYCLE) in the socket present state register (CardBus offset 08h, 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 PCI4410A device detects that bit 2 (CDETECT2) in the socket present state register (CardBus offset 08h, 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 PCI4410A device detects that bit 1 (CDETECT1) in the socket present state register (CardBus offset 08h, 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 (CardBus offset 08h, 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 (CardBus offset 00h, 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 Description

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 (CardBus offset 08h, 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 (CardBus offset 08h, 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 (CardBus offset 08h, 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 (CardBus offset 0Ch, 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 updated only at each insertion. Also, note that the PCI4410A device 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 Description

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 PCI4410A device 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 PCI4410A device 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 PCI4410A device does support 3.3-V V_{CC}; therefore, this bit always is set unless overridden by the socket force event register (CardBus offset 0Ch, 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 PCI4410A device does support 5-V V_{CC}; therefore, this bit always is set unless overridden by the socket force event register (CardBus offset 0Ch, 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 PCI4410A device. 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 is 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 Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION

6	IREQCINT	R	READY(IREQ)//CINT. Bit 6 indicates the current status of READY(IREQ)//CINT at the PC Card interface.
			0 = READY(IREQ)//CINT low
			1 = READY(IREQ)//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	CCD2. Bit 2 reflects the current status of CCD2 at the PC Card interface. Changes to this signal during card interrogation are not reflected here. 0 = CCD2 low (PC Card may be present) 1 = CCD2 high (PC Card not present)
1	CDETECT1	R	CCD1. Bit 1 reflects the current status of CCD1 at the PC Card interface. Changes to this signal during card interrogation are not reflected here. 0 = CCD1 low (PC Card may be present) 1 = 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 (CardBus offset 00h, see Section 6.1) and the socket present state register (CardBus offset 08h, 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 Description

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 PCI4410A device re-interrogates the PC Card, updates the socket present state register (CardBus offset 08h, see Section 6.3), and enables the socket control register (CardBus offset 10h, 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 (CardBus offset 08h, see Section 6.3). When set, this bit disables the socket control register (CardBus offset 10h, 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 (CardBus offset 08h, see Section 6.3). When set, this bit disables the socket control register (CardBus offset 10h, 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 (CardBus offset 08h, see Section 6.3). When set, this bit disables the socket control register (CardBus offset 10h, 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 (CardBus offset 08h, see Section 6.3). When set, this bit disables the socket control register (CardBus offset 10h, see Section 6.5).
9	FBADVCCREQ	W	Force bad V_{CC} request. Changes to bit 9 (BADVCCREQ) in the socket present state register (CardBus offset 08h, 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 (CardBus offset 08h, 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 (CardBus offset 08h, 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 (CardBus offset 08h, 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 (CardBus offset 08h, 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 (CardBus offset 00h, see Section 6.1), and bit 3 (PWRCYCLE) in the socket present state register is unaffected (CardBus offset 08h, see Section 6.3).

Table 6–5. Socket Force Event Register Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION
2	FCDETECT2	W	Force CCD2. Write transactions to bit 2 cause bit 2 (CD2EVENT) in the socket event register to be written (CardBus offset 00h, see Section 6.1), and bit 2 (CDETECT2) in the socket present state register is unaffected (CardBus offset 08h, see Section 6.3).
1	FCDETECT1	W	Force CCD1. Write transactions to bit 1 cause bit 1 (CD1EVENT) in the socket event register to be written (CardBus offset 00h, see Section 6.1), and bit 1 (CDETECT1) in the socket present state register is unaffected (CardBus offset 08h, 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 (CardBus offset 00h, see Section 6.1), and bit 0 (CARDSTS) in the socket present state register is unaffected (CardBus offset 08h, 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 CLKRUN protocol. The PCI4410A device 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 Description

BIT	SIGNAL	TYPE	FUNCTION
31–8	RSVD	R	Reserved. Bits 31–8 return 0s when read.
7	STOPCLK	R/W	CB CLKRUN protocol instructions. 0 = CB CLKRUN protocol can only attempt to stop/slow the CB clock if the socket is idle and the PCI CLKRUN protocol is preparing to stop/slow the PCI bus clock. 1 = CB 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 Description

BIT	SIGNAL	TYPE	FUNCTION
31–26	RSVD	R	Reserved. Bits 31–26 return 0s when read.
25	SKTACCES	R	Socket access status. This bit indicates 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 as bits 15–4 (DMABASE field) of the socket DMA register 1 (PCI offset 98h, see Section 4.38) points to a 16-byte region in PCI I/O space where the DMA registers reside. The names and locations of these registers are summarized in Table 7–1. These PCI4410A 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 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 PCI4410A device 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

<table>
<tr><th>TYPE</th><th colspan="4">REGISTER NAME</th><th>DMA BASE ADDRESS OFFSET</th></tr>
<tr><td>R</td><td rowspan="2">Reserved</td><td rowspan="2">Page</td><td colspan="2">Current address</td><td rowspan="2">00h</td></tr>
<tr><td>W</td><td colspan="2">Base address</td></tr>
<tr><td>R</td><td rowspan="2">Reserved</td><td rowspan="2">Reserved</td><td colspan="2">Current count</td><td rowspan="2">04h</td></tr>
<tr><td>W</td><td colspan="2">Base count</td></tr>
<tr><td>R</td><td>N/A</td><td rowspan="2">Reserved</td><td>N/A</td><td>Status</td><td rowspan="2">08h</td></tr>
<tr><td>W</td><td>Mode</td><td>Request</td><td>Command</td></tr>
<tr><td>R</td><td>Multichannel</td><td rowspan="2">Reserved</td><td>N/A</td><td rowspan="2">Reserved</td><td rowspan="2">0Ch</td></tr>
<tr><td>W</td><td>Mask</td><td>Master clear</td></tr>
</table>

7.1 DMA Current Address/Base Address Register

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

For the 8-bit DMA 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 DMA page register (see Section 7.2) are presented on AD23–AD16 of the PCI bus during the address phase.

For the 16-bit DMA 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 DMA 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	DMA 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	DMA 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: **DMA current address/base address**
Type: Read/Write
Offset: DMA base address + 00h
Default: 0000h

7.2 DMA Page Register

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

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DMA 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: **DMA page**
Type: Read/Write
Offset: DMA base address + 02h
Default: 00h

7.3 DMA Current Count/Base Count Register

The DMA 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	DMA 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	DMA 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: **DMA current count/base count**
Type: Read/Write
Offset: DMA base address + 04h
Default: 0000h

7.4 DMA Command Register

The DMA command register enables and disables the DMA controller. Bit 2 (DMAEN) defaults to 0, enabling the DMA 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	DMA command							
Type	R	R	R	R	R	R/W	R	R
Default	0	0	0	0	0	0	0	0

Register: **DMA command**
Type: Read-only, Read/Write
Offset: DMA base address + 08h
Default: 00h

Table 7–2. DMA Command Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–3	RSVD	R	Reserved. Bits 7–3 return 0s when read.
2	DMAEN	R/W	DMA controller enable. Bit 2 enables and disables the distributed DMA slave controller in the PCI4410A device and defaults to the enabled state. 0 = DMA controller is enabled (default). 1 = DMA controller is disabled.
1–0	RSVD	R	Reserved. Bits 1 and 0 return 0s when read.

7.5 DMA Status Register

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

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

Register: **DMA status**
Type: Read-only
Offset: DMA base address + 08h
Default: 00h

Table 7–3. DMA Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–4	DREQSTAT	R	Channel request. In the 8237 DMA controller, bits 7–4 indicate the status of $\overline{DREQ}$ of each DMA channel. In the PCI4410A device, these bits indicate the $\overline{DREQ}$ status of the single socket being serviced by this register. All four bits are set to 1 when the PC Card asserts $\overline{DREQ}$ and are reset to 0 when $\overline{DREQ}$ is deasserted. The status of bit 0 (MASKBIT) in the DMA multichannel/mask register (see Section 7.9) has no effect on these bits.
3–0	TC	R	Channel terminal count. The 8327 DMA controller uses bits 3–0 to indicate the TC status of each of its four DMA channels. In the PCI4410A device, 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 DMA Request Register

The DMA request register requests a DMA 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	DMA request							
Type	W	W	W	W	W	W	W	W
Default	0	0	0	0	0	0	0	0

Register: **DMA request**
Type: Write-only
Offset: DMA base address + 09h
Default: 00h

7.7 DMA Mode Register

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

Bit	7	6	5	4	3	2	1	0
Name	DMA 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: **DMA mode**
Type: Read-only, Read/Write
Offset: DMA base address + 0Bh
Default: 00h

Table 7–4. DMA Mode Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–6	DMAMODE	R/W	Mode select. The PCI4410A device 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 PCI4410A device uses bit 5 to select the memory address in the DMA current address/base address register to increment or decrement after each data transfer. This is in accordance with the 8237 DMA controller 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 is disabled (default). 1 = Auto initialization is 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 PCI4410A PC Card interface to memory and a memory read transfer moves data from memory to the PCI4410A 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 DMA Master Clear Register

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

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

Register: **DMA master clear**
Type: Write-only
Offset: DMA base address + 0Dh
Default: 00h

7.9 DMA Multichannel/Mask Register

The PCI4410A device uses only the least significant bit of this register to mask the PC Card DMA channel. The PCI4410A device 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	DMA multichannel/mask							
Type	R	R	R	R	R	R	R	R/W
Default	0	0	0	0	0	0	0	0

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

Table 7–5. DMA Multichannel/Mask Register Description

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 $\overline{DREQ}$ signals from the PC Card. When set to 1, the socket ignores DMA requests from the card. When cleared (or reset to 0), incoming $\overline{DREQ}$ assertions are serviced normally. 0 = DMA service is provided on card $\overline{DREQ}$. 1 = Socket $\overline{DREQ}$ signal is 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 typically is 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 can be read by software.
W	Write	Field can be written by software to any value.
S	Set	Field can be set to 1 by a write of 1. Writes of 0 have no effect.
C	Clear	Field can be reset to 0 by a write of 1. Writes of 0 have no effect.
U	Update	Field can be autonomously updated by the PCI4410A device.

8.1 PCI Configuration Registers

The PCI4410A link function configuration header is compliant with the *PCI Local Bus 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–28h
Subsystem ID		Subsystem vendor ID		2Ch
Reserved				30h
Reserved			Capabilities 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 control and status		48h
Reserved				4C–ECh
PCI miscellaneous configuration				F0h
Link enhancements				F4h
Subsystem ID alias		Subsystem vendor ID alias		F8h
GPIO3	GPIO2	GPIO1	GPIO0	FCh

8.2 Vendor ID Register

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

8.3 Device ID Register

The device ID register contains a value assigned to the PCI4410A device by Texas Instruments. The device identification for the PCI4410A 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 PCI4410A 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 Description

BIT	SIGNAL	TYPE	FUNCTION
15–10	RSVD	R	Reserved. Bits 15–10 return 0s when read.
9	FBB_ENB	R	Fast back-to-back enable. The PCI4410A device does not generate fast back-to-back transactions; therefore, this bit returns 0 when read.
8	SERR_ENB	R/W	$\overline{\text{SERR}}$ enable. When this bit is set to 1, the PCI4410A $\overline{\text{SERR}}$ 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 PCI4410A device does not support address/data stepping; therefore, this bit is hardwired to 0.
6	PERR_ENB	R/W	Parity error enable. When this bit is set to 1, the PCI4410A device is enabled to drive the $\overline{\text{PERR}}$ response to parity errors through the $\overline{\text{PERR}}$ signal.
5	VGA_ENB	R	VGA palette snoop enable. The PCI4410A device 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 PCI4410A device is enabled to generate MWI PCI bus commands. If this bit is cleared, the PCI4410A device generates memory write commands instead.
3	SPECIAL	R	Special cycle enable. The PCI4410A device 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 PCI4410A device is enabled to initiate cycles on the PCI bus.
1	MEMORY_ENB	R/W	Memory response enable. Setting this bit to 1 enables the PCI4410A device 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 PCI4410A link does not implement any I/O-mapped functionality; therefore, 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 Description

BIT	SIGNAL	TYPE	FUNCTION
15	PAR_ERR	RCU	Detected parity error. This bit is set to 1 when either an address parity or data parity error is detected.
14	SYS_ERR	RCU	Signaled system error. This bit is set to 1 when $\overline{\text{SERR}}$ is enabled and the PCI4410A device has 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 PCI4410A device on the PCI bus is terminated by a master abort.
12	TABORT_REC	RCU	Received target abort. This bit is set to 1 when a cycle initiated by the PCI4410A device 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 PCI4410A device when it terminates a transaction on the PCI bus with a target abort.
10–9	PCI_SPEED	R	DEVSEL timing. Bits 10 and 9 encode the timing of $\overline{\text{DEVSEL}}$ and are hardwired 01b, indicating that the PCI4410A device asserts this signal at a medium speed on nonconfiguration 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 PCI4410A device. b. The PCI4410A device was the bus master during the data parity error. c. Bit 6 (PERR_ENB) in the PCI command register (PCI offset 04h, see Section 8.4) is set to 1.
7	FBB_CAP	R	Fast back-to-back capable. The PCI4410A device cannot accept fast back-to-back transactions; therefore, this bit is hardwired to 0.
6	UDF	R	User-definable features (UDF) supported. The PCI4410A device does not support the UDF; therefore, this bit is hardwired to 0.
5	66MHZ	R	66-MHz capable. The PCI4410A device 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, indicating 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. Bits 3–0 return 0s when read.

8.6 Class Code and Revision ID Register

The class code and revision ID register categorizes the PCI4410A device as a serial bus controller (0Ch), controlling an IEEE 1394 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	1	0

Register: **Class code and revision ID**
Type: Read-only
Offset: 08h
Default: 0C00 1002h

Table 8–5. Class Code and Revision ID Register Description

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 IEEE 1394 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 Open Host Controller Interface Specification*.
7–0	CHIPREV	R	Silicon revision. This field returns the silicon revision of the PCI4410A device.

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 PCI4410A device. 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 Description

BIT	SIGNAL	TYPE	FUNCTION
15–8	LATENCY_TIMER	R/W	PCI latency timer. The value in this field specifies the latency timer for the PCI4410A device, in units of PCI clock cycles. When the PCI4410A device is a PCI bus initiator and asserts $\overline{\text{FRAME}}$, the latency timer begins counting from zero. If the latency timer expires before the PCI4410A transaction has terminated, the PCI4410A device terminates the transaction when its $\overline{\text{GNT}}$ is deasserted.
7–0	CACHELINE_SZ	R/W	Cache line size. This value is used by the PCI4410A device during memory write and invalidate, memory-read line, and memory-read multiple transactions.

8.8 Header Type and BIST Register

The header type and built-in self-test (BIST) register indicates that this function is part of a multifunction device, and has a standard PCI header type and no BIST. 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 Description

BIT	SIGNAL	TYPE	FUNCTION
15–8	BIST	R	Built-in self-test. The PCI4410A device does not include a BIST, thus this field returns 00h when read.
7–0	HEADER_TYPE	R	PCI header type. The PCI4410A device includes the standard PCI header, and this is communicated by returning 00h when this field is read.

8.9 Open HCI Base Address Register

The open HCI 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 2 Kbytes 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 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 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 base address**
Type: Read-only, Read/Write
Offset: 10h
Default: 0000 0000h

Table 8–8. Open HCI Registers Base Address Register Description

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, indicating that the OHCI registers require a 2-Kbyte region of memory.
3	OHCI_PF	R	OHCI register prefetch. This bit returns 0 when read, indicating that the OHCI registers are nonprefetchable.
2–1	OHCI_MEMTYPE	R	Open HCI memory type. This field returns 0s when read, indicating that the OHCI base address 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 when read, indicating that 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/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	TI extension 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: **TI extension base address**
Type: Read-only
Offset: 14h
Default: 0000 0000h

Table 8–9. TI Extension Base Address Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–11	TI_EXTREG_PTR	R/W	TI extension register pointer. Specifies the upper 21 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, indicating that the TI extension registers require a 2-Kbyte region of memory.
3	TI_PF	R	TI extension register prefetch. This bit returns 0 when read, indicating that the TI extension registers are nonprefetchable.
2–1	TI_MEMTYPE	R	TI memory type. This field returns 0s when read, indicating 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 when read, indicating that 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 identification register (offset F8h, see Section 8.22). 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 Description

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. The PCI4410A 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 Description

BIT	SIGNAL	TYPE	FUNCTION
15–8	INTR_PIN	R	Interrupt pin. This field returns 01h or 02h when read, indicating that the PCI4410A link function signals interrupts on the INTA or INTB terminal, respectively. If bit 29 (TIE_INTB_INTA) in the system control register (offset 80h, see Section 4.29) is set to 1, the INTR_PIN byte reads 0000 0001b, which indicates the OHCI function is signaling on INTA.
7–0	INTR_LINE	R/W	Interrupt line. This field is programmed by the system and indicates to the software which interrupt line the PCI4410A INTA is connected to.

8.14 MIN_GNT and MAX_LAT Register

The MIN_GNT and MAX_LAT register is used to communicate to the system the desired setting of bits 15–8 in the latency timer and class cache line size register (offset 0Ch, see Section 8.7). If a serial EEPROM is detected, the contents of this register are loaded through the serial EEPROM interface after a PRST. If no serial EEPROM is detected, this register returns 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 Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–8	MAX_LAT	RU	Maximum latency. The contents of this field may be used by host BIOS to assign an arbitration priority level to the PCI4410A device. The default for this field indicates that the PCI4410A device may need to access the PCI bus as often as every 0.25 µs; thus, an extremely high priority level is requested. The contents of this field may also be loaded through the serial EEPROM.
7–0	MIN_GNT	RU	Minimum grant. The contents of this field may be used by host BIOS to assign a latency timer register value to the PCI4410A device. The default for this field indicates that the PCI4410A device may need to sustain burst transfers for nearly 64 µs, thus requesting a large value be programmed in bits 15–8 of the PCI4410A latency timer and class cache line size register (offset 0Ch, see Section 8.7).

8.15 PCI OHCI Control Register

The PCI OHCI control register contains IEEE 1394 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	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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
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 Register

The capability ID and next item pointer register identifies the linked-list capability item and provides 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 Description

BIT	SIGNAL	TYPE	FUNCTION
15–8	NEXT_ITEM	R	Next item pointer. The PCI4410A device 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 PCI4410A device 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, Read-only
Offset: 46h
Default: 6411h

Table 8–14. Power Management Capabilities Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	PME_D3COLD	RU	PCI_PME support from $D3_{cold}$. This bit can be set to 1 or cleared to 0 via bit 15 (PME_D3COLD) in the miscellaneous configuration register (offset F0h, see Section 8.20). The miscellaneous configuration register is loaded from the serial EEPROM. When this bit is set to 1, it indicates that the PCI4410A device is capable of generating a PCI_PME wake event from $D3_{cold}$. This bit state is dependent upon the PCI4410A V_{AUX} implementation and may be configured by using bit 15 (PME_D3COLD) in the miscellaneous configuration register (see Section 8.20).
14–11	PME_SUPPORT	RU	PME support. This four-bit field indicates the power states from which the PCI4410A device may assert PME. This field returns a value of 1100b by default, indicating that PME may be asserted from the $D3_{hot}$ and D2 power states. Bit 13 may be modified by host software using bit 13 (PME_SUPPORT_D2) in the PCI miscellaneous configuration register (offset F0h, see Section 8.20).
10	D2_SUPPORT	RU	D2 support. This bit can be set or cleared via bit 10 (D2_SUPPORT) in the miscellaneous configuration register (offset F0h, see Section 8.20). The miscellaneous configuration register is loaded from the serial EEPROM. When this bit is set, it indicates that D2 support is present. When this bit is cleared, it indicates that D2 support is not present for backward compatibility. For normal operation, this bit is set to 1.
9	D1_SUPPORT	R	D1 support. This bit returns a 0 when read, indicating that the PCI4410A device does not support the D1 power state.
8	DYN_DATA	R	Dynamic data support. This bit returns a 0 when read, indicating that the PCI4410A device does not report dynamic power-consumption data.
7–6	RSVD	R	Reserved. Bits 7 and 6 return 0s when read.
5	DSI	R	Device-specific initialization. This bit returns 0 when read, indicating that the PCI4410A device 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 PCI4410A device supports PME generation in the $D3_{cold}$ device state and requires V_{aux}, this bit returns 1 when read.
3	PME_CLK	R	PME clock. This bit returns 0 when read, indicating that no host bus clock is required for the PCI4410A device to generate PME.
2–0	PM_VERSION	R	Power-management version. This field returns 001b when read, indicating that the PCI4410A device is compatible with the registers described in the *PCI Bus Power Management Interface Specification* (Revision 1.0).

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 Description

BIT	SIGNAL	TYPE	FUNCTION
15	PME_STS	RC	This bit is set to 1 when the PCI4410A device normally would be asserting the $\overline{PME}$ signal, independent of the state of bit 8 (PME_ENB). This bit is cleared by a writeback of 1, which also clears the $\overline{PME}$ signal driven by the PCI4410A device. 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 PCI4410A device does not report dynamic data.
8	PME_ENB	R/W	When bit 8 = 1, $\overline{PME}$ assertion is enabled. When bit 8 = 0, $\overline{PME}$ assertion is disabled. This bit defaults to 0 if the function does not support $\overline{PME}$ generation from $D3_{cold}$. If the function supports $\overline{PME}$ from $D3_{cold}$, then this bit is sticky and must be explicitly cleared by the operating system each time it is initially loaded. Functions that do not support $\overline{PME}$ generation from any D-state (that is, bits 15–11 in the power management capabilities register (offset 46h, see Section 8.17) equal 00000b), may hardwire this bit to be read-only, always returning a 0 when read by system software.
7–5	RSVD	R	Reserved. Bits 7–5 return 0s when read.
4	DYN_DATA	R	Dynamic data. This bit returns 0 when read because the PCI4410A device does not report dynamic data.
3–2	RSVD	R	Reserved. Bits 3 and 2 return 0s when read.
1–0	PWR_STATE	R/W	Power state. This two-bit field is used to set the PCI4410A device power state and is encoded as follows: 00 = Current power state is D0. 01 = Current power state is D1 (not supported by this device). 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 PCI4410A device; 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 Description

BIT	SIGNAL	TYPE	FUNCTION
15–0	RSVD	R	Reserved. Bits 15–0 return 0s when read.

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	R	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 Description

BIT	SIGNAL	TYPE	FUNCTION
31–16	RSVD	R	Reserved. Bits 31–16 return 0s when read.
15	PME_D3COLD	R/W	PME support from $D3_{cold}$. This bit is used to program bit 15 (PME_D3COLD) in the power management capabilities register (offset 46h, see Section 8.17). This bit retains state through PRST and D3–D0 transitions.
14	RSVD	R	Reserved. Bit 14 returns 0 when read.
13	PME_SUPPORT_D2	R/W	PME support. This bit is used to program bit 13 (PME_SUPPORT_D2) in the power management capabilities register (offset 46h, see Section 8.17). If wake up from the D2 power state implemented in the PCI4410A device is not desired, this bit is cleared to indicate to power-management software that wake-up from D2 is not supported. This bit retains state through PRST and D3–D0 transitions.
12–11	RSVD	R	Reserved. Bits 12 and 11 return 0s when read.
10	D2_SUPPORT	R/W	D2 support. This bit is used to program bit 10 (D2_SUPPORT) in the power management capabilities register (offset 46h, see Section 8.17). If the D2 power state implemented in the PCI4410A device is not desired, this bit can be cleared to indicate to power-management software that D2 is not supported. This bit retains state through PRST and D3–D0 transitions.
9–3	RSVD	R	Reserved. Bits 9–3 return 0s when read.
2	DISABLE_SCLKGATE	R/W	When this bit is set to 1, the internal SCLK runs identically with the chip input. This bit is a test feature only and should be cleared to 0 (all applications).
1	DISABLE_PCIGATE	R/W	When this bit is set to 1, the internal PCI clock runs identically with the chip input. This bit is a test feature only and should be cleared to 0 (all applications).
0	KEEP_PCLK	R/W	When this bit is set to 1, the PCI clock always is kept running through the CLKRUN protocol. When this bit is cleared, the PCI clock can be stopped using 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 bit 22 (aPhyEnhanceEnable) 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	R	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 Description

BIT	SIGNAL	TYPE	FUNCTION
31–14	RSVD	R	Reserved. Bits 31–14 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 the PCI4410A device retries the packet, it uses a 2-Kbyte threshold, resulting in a 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 These bits fine-tune the asynchronous transmit threshold. For most applications the 1.7-K threshold is optimal. Changing this value may increase or decrease the 1394 latency depending on the average PCI bus latency. Setting the AT threshold to 1.7K, 1K, or 512 bytes results in data being transmitted at these thresholds, or when an entire packet has been checked into the FIFO. If the packet to be transmitted is larger than the AT threshold, the remaining data must be received before the AT FIFO is emptied; otherwise, an underrun condition will occur, resulting in a packet error at the receiving node. As a result, the link will then commence store-and-forward operation, that is, wait until it has the complete packet in the FIFO before retransmitting it on the second attempt, to ensure delivery. An AT threshold of 2K results in store-and-forward operation, which means that asynchronous data will not be transmitted until an end-of-packet token is received. Restated, setting the AT threshold to 2K results in only complete packets being transmitted.
11–8	RSVD	R	Reserved. Bits 11–8 return 0s when read.
7	enab_unfair	R/W	Enable asynchronous priority requests. OHCI-Lynx™ (TSB12LV22) compatible. Setting this bit to 1 enables the link to respond to requests with priority arbitration. It is recommended that this bit be set to 1.
6	RSVD	R	This reserved field is not assigned in PCI4410A follow-on products, since this bit location loaded by the serial EEPROM from the *enhancements* field corresponds to bit 23 (programPhyEnable) in the host controller control register (offset 50h/54h, see Section 9.16).
5–3	RSVD	R	Reserved. Bits 5–3 return 0 when read.
2	enab_insert_idle	R/W	Enable insert idle. OHCI-Lynx™ (TSB12LV22) compatible. When the PHY device has control of the PHY_CTL0–PHY_CTL1 control lines and PHY_DATA0–PHY_DATA7 data lines and the link requests control, the PHY drives 11b on the PHY_CTL0–PHY_CTL1 lines. The link then can start driving these lines immediately. Setting this bit to 1 inserts an idle state, so the link waits one clock cycle before it starts driving the lines (turnaround time). It is recommended that this bit be set to 1.

Table 8–18. Link Enhancement Control Register Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION
1	enab_accel	R/W	Enable acceleration enhancements. OHCI-Lynx™ (TSB12LV22) compatible. When set to 1, this bit notifies the PHY that the link supports the IEEE 1394a-2000 acceleration enhancements, that is, ack-accelerated, fly-by concatenation, etc. It is recommended that this bit be set to 1.
0	RSVD	R	Reserved. Bit 0 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 the 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 Description

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

BIT	SIGNAL	TYPE	FUNCTION
31–30	RSVD	R	Reserved. Bits 31 and 30 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 is enabled
27–25	RSVD	R	Reserved. Bits 27–25 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. Bits 23 and 22 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 is enabled
19–17	RSVD	R	Reserved. Bits 19–17 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. Bit 14 returns 0 when read.
13	GPIO_INV1	R/W	GPIO1 polarity invert. When bit 15 (DISABLE_LPS) 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 Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION
12	GPIO_ENB1	R/W	GPIO1 enable control. When bit 15 (DISABLE_LPS) is set to 1, this bit controls the output enable for GPIO1. 0 = High-impedance output 1 = Output is enabled (default)
11–9	RSVD	R	Reserved. Bits 11–9 return 0s when read.
8	GPIO_DATA1	R/W	GPIO1 data. When bit 15 (DISABLE_LPS) 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 terminal as bus manager contender or GPIO0. 0 = BMC (default) 1 = GPIO0
6	RSVD	R	Reserved. Bit 6 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 for GPIO0. 0 = Noninverted (default) 1 = Inverted
4	GPIO_ENB0	R/W	GPIO0 enable control. When bit 7 (DISABLE_BMC) is set to 1, this bit controls the output enable for GPIO0. 0 = High-impedance output 1 = Output is enabled (default)
3–1	RSVD	R	Reserved. Bits 3–1 return 0s when read.
0	GPIO_DATA0	R/W	GPIO0 data. When bit 7 (DISABLE_BMC) 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 *1394 Open Host Controller Interface 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 (see Section 8.9). These registers are the primary interface for controlling the PCI4410A IEEE 1394 link function.

This section provides the register interface and bit descriptions. Several set/clear register pairs in this programming model 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-bit written to RegisterSet causes the corresponding bit in the set/clear register to be set to 1; a 0 bit leaves the corresponding bit unaffected. A 1-bit written to RegisterClear causes the corresponding bit in the set/clear register to be cleared; a 0 bit leaves the corresponding bit in the set/clear register unaffected.

Typically, a read from either RegisterSet or RegisterClear returns the contents of the set or clear register, respectively. However, sometimes reading the RegisterClear provides a masked version of the set or 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 – 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
	Host controller control	HCControlSet	50h
		HCControlClr	54h
	Reserved	—	58h – 5Ch

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

DMA CONTEXT	REGISTER NAME	ABBREVIATION	OFFSET
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]	Asynchronous context control	ContextControlSet	180h
		ContextControlClear	184h
	Reserved	—	188h
	Asynchronous context command pointer	CommandPtr	18Ch
	Reserved	—	190h – 19Ch
Asynchronous response transmit [ATRS]	Asynchronous context control	ContextControlSet	1A0h
		ContextControlClear	1A4h
	Reserved	—	1A8h
	Asynchronous context command pointer	CommandPtr	1ACh
	Reserved	—	1B0h – 1BCh
Asynchronous request receive [ARRQ]	Asynchronous context control	ContextControlSet	1C0h
		ContextControlClear	1C4h
	Reserved	—	1C8h
	Asynchronous context command pointer	CommandPtr	1CCh
	Reserved	—	1D0h – 1DCh
Asynchronous response receive [ARRS]	Asynchronous context control	ContextControlSet	1E0h
		ContextControlClear	1E4h
	Reserved	—	1E8h
	Asynchronous context command pointer	CommandPtr	1ECh
	Reserved	—	1F0h – 1FCh
Isochronous transmit context n n = 0, 1, 2, 3, … 7	Isochronous transmit context control	ContextControlSet	200h + 16*n
		ContextControlClear	204h + 16*n
	Reserved	—	208h + 16*n
	Isochronous transmit context command pointer	CommandPtr	20Ch + 16*n
	Reserved	—	210h – 3FCh
Isochronous receive context n n = 0, 1, 2, 3	Isochronous receive context control	ContextControlSet	400h + 32*n
		ContextControlClear	404h + 32*n
	Reserved	—	408h + 32*n
	Isochronous receive context command pointer	CommandPtr	40Ch + 32*n
	Isochronous receive context match	ContextMatch	410h + 32*n

9.1 OHCI Version Register

The OHCI version register indicates the OHCI version support, and whether or not the serial EEPROM 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 Description

BIT	SIGNAL	TYPE	FUNCTION
31–25	RSVD	R	Reserved. Bits 31–25 return 0s when read.
24	GUID_ROM	R	The PCI4410A device sets bit 24 to 1 if the serial EEPROM is detected. If the serial EEPROM is present, the Bus_Info_Block is loaded automatically on hardware reset.
23–16	version	R	Major version of the open HCI. The PCI4410A device is compliant with the *1394 Open Host Controller Interface Specification*; 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 PCI4410A device is compliant with the *1394 Open Host Controller Interface Specification*; thus, this field reads 00h.

9.2 GUID ROM Register

The GUID ROM register is used to access the serial EEPROM, and is applicable only if bit 24 (GUID_ROM) in the OHCI version register (offset 00h, see Section 9.1) is 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
Bit	**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 Description

BIT	SIGNAL	TYPE	FUNCTION
31	addrReset	RSU	Software sets this bit to 1 to reset the GUID ROM address to 0. When the PCI4410A device completes the reset, it clears this bit. The PCI4410A device does not automatically fill bits 23–16 (rdData field) with the 0^{th} 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 PCI4410A device 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

The asynchronous transmit retries register indicates the number of times the PCI4410A device attempts 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 Description

BIT	SIGNAL	TYPE	FUNCTION
31–29	secondLimit	R	The second limit field returns 0s when read, because outbound dual-phase retry is not implemented.
28–16	cycleLimit	R	The cycle limit field returns 0s when read, because outbound dual-phase retry is not implemented.
15–12	RSVD	R	Reserved. Bits 15–12 return 0s 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

The CSR data 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

The CSR compare 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

The CSR control 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 to 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	RU	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 Description

BIT	SIGNAL	TYPE	FUNCTION
31	csrDone	RU	This bit is set to 1 by the PCI4410A device when a compare-swap operation is complete. It is cleared 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

The configuration ROM header 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 Description

BIT	SIGNAL	TYPE	FUNCTION
31–24	info_length	R/W	IEEE 1394 bus-management field. Must be valid when bit 17 (linkEnable) in the host controller control register (offset 50h/54h, see Section 9.16) is set to 1.
23–16	crc_length	R/W	IEEE 1394 bus-management field. Must be valid when bit 17 (linkEnable) in the host controller control register (offset 50h/54h, see Section 9.16) is set to 1.
15–0	rom_crc_value	R/W	IEEE 1394 bus-management field. Must be valid at any time bit 17 (linkEnable) in the host controller control register (offset 50h/54h, see Section 9.16) is set to 1. The reset value is undefined if no serial EEPROM is present. If a serial EEPROM is present, this field is loaded from the serial EEPROM.

9.8 Bus Identification Register

The bus identification 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

The bus options 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 Description

BIT	SIGNAL	TYPE	FUNCTION
31	irmc	R/W	Isochronous resource-manager capable. IEEE 1394 bus-management field. Must be valid when bit 17 (linkEnable) in the host controller control register (offset 50h/54h, see Section 9.16) is set to 1.
30	cmc	R/W	Cycle master capable. IEEE 1394 bus-management field. Must be valid when bit 17 (linkEnable) in the host controller control register (offset 50h/54h, see Section 9.16) is set to 1.
29	isc	R/W	Isochronous support capable. IEEE 1394 bus-management field. Must be valid when bit 17 (linkEnable) in the host controller control register (offset 50h/54h, see Section 9.16) is set to 1.
28	bmc	R/W	Bus manager capable. IEEE 1394 bus-management field. Must be valid when bit 17 (linkEnable) in the host controller control register (offset 50h/54h, see Section 9.16) is set to 1.
27	pmc	R/W	Power-management capable. IEEE 1394 bus-management field. When bit 27 is set, this indicates that the node is power-management capable. Must be valid when bit 17 (linkEnable) in the host controller control register (offset 50h/54h, see Section 9.16) is set to 1.
26–24	RSVD	R	Reserved. Bits 26–24 return 0s when read.
23–16	cyc_clk_acc	R/W	Cycle master clock accuracy in parts per million. IEEE 1394 bus-management field. Must be valid when bit 17 (linkEnable) in the host controller control register (offset 50h/54h, see Section 9.16) is set to 1.
15–12	max_rec	R/W	Maximum request. 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 this field; however, this field must be valid at any time bit 17 (linkEnable) in the host controller control register (offset 50h/54h, see Section 9.16) 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 a 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

The GUID high 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 EEPROM is detected, the contents of this register are loaded through the serial EEPROM interface after a $\overline{\text{PRST}}$. At that point, the contents of this register cannot be changed. If no serial EEPROM is detected, then the contents of this register are loaded by the BIOS after a $\overline{\text{PRST}}$. 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

The GUID low 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 (offset 24h, see Section 9.10).

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

The configuration ROM mapping register contains the start address within system memory that maps 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 Description

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, 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

The posted write address low 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 Description

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

The posted write address high register is used to communicate error information if a write request is posted and an error occurs while writing the posted data packet. 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 Description

BIT	SIGNAL	TYPE	FUNCTION
31–16	sourceID	RU	This field is the 10-bit bus number (bits 31–22) and 6-bit node number (bits 21–16) 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 PCI4410A device 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

The host controller control set/clear register pair provides flags for controlling the PCI4410A 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, Read/Set/Clear, Read/Clear, Read-only
Offset: 50h set register
54h clear register
Default: X00X 0000h

Table 9–11. Host Controller Control Register Description

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 PCI4410A device 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 IEEE 1394a-2000 enhancements in the link and PHY. When this bit is 1, generic software such as the OHCI driver is responsible for configuring IEEE 1394a-2000 enhancements in the PHY and bit 22 (aPhyEnhanceEnable) in the PCI4410A device. When this bit is 0, the generic software may not modify the IEEE 1394a-2000 enhancements in the PCI4410A device or PHY and cannot interpret the setting of bit 22 (aPhyEnhanceEnable). This bit is 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 IEEE 1394a-2000 enhancements. When bit 23 (programPhyEnable) is 0, the software does not change the PHY enhancements or this 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 PCI4410A device is logically and immediately disconnected from the 1394 bus, no packets are received or processed, nor are packets transmitted.
16	SoftReset	RSCU	When this bit is set to 1, all PCI4410A states are reset, all FIFOs 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

The self-ID buffer pointer register points to the 2-Kbyte aligned base address of the buffer in host memory where the self-ID packets are stored during bus initialization. Bits 31–11 are read/write accessible. Bits 10–0 are reserved, and return 0s when read.

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

The self-ID count 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 Description

BIT	SIGNAL	TYPE	FUNCTION
31	selfIDError	RU	This bit is 1 if 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 bits 23–16 (selfIDGeneration field). 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 Isochronous Receive Channel Mask High Register

The isochronous receive channel mask high 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 content of the isochronous receive channel mask high 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	Isochronous 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	Isochronous 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: **Isochronous receive channel mask high**
Type: Read/Set/Clear
Offset: 70h set register
74h clear register
Default: XXXX XXXXh

Table 9–13. Isochronous Receive Channel Mask High Register Description

BIT	SIGNAL	TYPE	FUNCTION
31	isoChannel63	RSC	When bit 31 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 63.
30	isoChannel62	RSC	When bit 30 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 62.
29	isoChannel61	RSC	When bit 29 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 61.
28	isoChannel60	RSC	When bit 28 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 60.
27	isoChannel59	RSC	When bit 27 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 59.
26	isoChannel58	RSC	When bit 26 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 58.
25	isoChannel57	RSC	When bit 25 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 57.
24	isoChannel56	RSC	When bit 24 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 56.
23	isoChannel55	RSC	When bit 23 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 55.
22	isoChannel54	RSC	When bit 22 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 54.
21	isoChannel53	RSC	When bit 21 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 53.
20	isoChannel52	RSC	When bit 20 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 52.
19	isoChannel51	RSC	When bit 19 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 51.
18	isoChannel50	RSC	When bit 18 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 50.
17	isoChannel49	RSC	When bit 17 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 49.
16	isoChannel48	RSC	When bit 16 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 48.
15	isoChannel47	RSC	When bit 15 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 47.
14	isoChannel46	RSC	When bit 14 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 46.
13	isoChannel45	RSC	When bit 13 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 45.
12	isoChannel44	RSC	When bit 12 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 44.
11	isoChannel43	RSC	When bit 11 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 43.
10	isoChannel42	RSC	When bit 10 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 42.
9	isoChannel41	RSC	When bit 9 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 41.
8	isoChannel40	RSC	When bit 8 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 40.
7	isoChannel39	RSC	When bit 7 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 39.
6	isoChannel38	RSC	When bit 6 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 38.

Table 9–13. Isochronous Receive Channel Mask High Register Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION
5	isoChannel37	RSC	When bit 5 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 37.
4	isoChannel36	RSC	When bit 4 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 36.
3	isoChannel35	RSC	When bit 3 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 35.
2	isoChannel34	RSC	When bit 2 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 34.
1	isoChannel33	RSC	When bit 1 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 33.
0	isoChannel32	RSC	When bit 0 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 32.

9.20 Isochronous Receive Channel Mask Low Register

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

Table 9–14. Isochronous Receive Channel Mask Low Register Description

BIT	SIGNAL	TYPE	FUNCTION
31	isoChannel31	RSC	When bit 31 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 31.
30	isoChannel30	RSC	When bit 30 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 30.
⋮	⋮	⋮	Bits 29 through 2 follow the same pattern.
1	isoChannel1	RSC	When bit 1 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 1.
0	isoChannel0	RSC	When bit 0 is set to 1, the PCI4410A device is enabled to receive from isochronous channel number 0.

9.21 Interrupt Event Register

The interrupt event set/clear register reflects the state of the various PCI4410A interrupt sources. The interrupt bits are set 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 clear a bit in this register is to write a 1 to the corresponding bit in the clear register.

This register is fully compliant with the *1394 Open Host Controller Interface Specification*, and the PCI4410A device adds a vendor-specific interrupt function to bit 30. When the interrupt event register is read, the return value is the bit-wise AND function of the interrupt event and interrupt mask registers. 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, Read/Update, Read-only
Offset: 80h set register
84h clear register (returns the content of the interrupt event register bitwise ANDed with the interrupt mask register when read)
Default: XXXX 0XXXh

Table 9–15. Interrupt Event Register Description

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 PCI4410A device has received a PHY register data byte that can be read from bits 23–16 of the PHY control register (offset ECh, see Section 9.30).
25	cycleTooLong	RSCU	If bit 21 (cycleMaster) of the link control register (offset E0h/E4h, see Section 9.28) is set to 1, this indicates that over 125 µs have elapsed between the start of sending a cycle start packet and the end of a subaction gap. The link control register bit 21 (cycleMaster) is cleared by this event.
24	unrecoverableError	RSCU	This event occurs when the PCI4410A device 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 bit 24 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 values for the cycleSeconds and cycleCount fields that are different from the values in bits 31–25 (cycleSeconds field) and bits 24–12 (cycleCount field) in the isochronous cycle timer register (offset F0h, see Section 9.31).
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. Bit 22 may be set to 1 either when a lost cycle occurs or when logic predicts that one will occur.
21	cycle64Seconds	RSCU	Indicates that the 7^{th} bit of the cycle second counter has changed.
20	cycleSynch	RSCU	Indicates that a new isochronous cycle has started. Bit 20 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 the bus reset mode.

Table 9–15. Interrupt Event Register Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION
16	selfIDcomplete	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 bit 17 (busReset) is turned on.
15–10	RSVD	R	Reserved. Bits 15–10 return 0s when read.
9	lockRespErr	RSCU	Indicates that the PCI4410A device 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 PCI4410A device 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 logical OR of all bits in the isochronous receive interrupt event (offset A0h/A4h, see Section 9.25) and isochronous receive interrupt mask (offset A8h/ACh, see Section 9.26) registers. The isochronous receive interrupt event register indicates which contexts have been 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 logical OR of all bits in the isochronous transmit interrupt event (offset 90h/94h, see Section 9.23) and isochronous transmit interrupt mask (offset 98h/9Ch, see Section 9.24) registers. The isochronous transmit interrupt event register indicates which contexts have been 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

The interrupt mask set/clear register is used to enable the various PCI4410A interrupt sources. Reads from either the set register or the clear register always return the contents of the interrupt mask register. 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 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/Set/Clear, Read/Update, Read-only
Offset: 88h set register
8Ch clear register
Default: XXXX 0XXXh

Table 9–16. Interrupt Mask Register Description

BIT	SIGNAL	TYPE	FUNCTION
31	masterIntEnable	RSC	If this bit is set to 1, external interrupts are generated in accordance with the interrupt mask register. If this bit is cleared, no external interrupts are generated regardless of the interrupt mask register settings.
30–0			See Table 9–15.

9.23 Isochronous Transmit Interrupt Event Register

The isochronous transmit interrupt event 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 isochTx (bit 6) interrupt has occurred in the interrupt event register (offset 80h/84h, see Section 9.21), 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 clear a bit in this register 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
94h clear register (returns the contents of the isochronous transmit interrupt event register bitwise ANDed with the isochronous transmit interrupt mask register when read)
Default: 0000 00XXh

Table 9–17. Isochronous Transmit Interrupt Event Register Description

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 interrupt event register bit 6 (isochTx) interrupt.
6	isoXmit6	RSC	Isochronous transmit channel 6 caused the interrupt event register bit 6 (isochTx) interrupt.
5	isoXmit5	RSC	Isochronous transmit channel 5 caused the interrupt event register bit 6 (isochTx) interrupt.
4	isoXmit4	RSC	Isochronous transmit channel 4 caused the interrupt event register bit 6 (isochTx) interrupt.
3	isoXmit3	RSC	Isochronous transmit channel 3 caused the interrupt event register bit 6 (isochTx) interrupt.
2	isoXmit2	RSC	Isochronous transmit channel 2 caused the interrupt event register bit 6 (isochTx) interrupt.
1	isoXmit1	RSC	Isochronous transmit channel 1 caused the interrupt event register bit 6 (isochTx) interrupt.
0	isoXmit0	RSC	Isochronous transmit channel 0 caused the interrupt event register bit 6 (isochTx) interrupt.

9.24 Isochronous Transmit Interrupt Mask Register

The isochronous transmit interrupt mask 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 the contents of the isochronous transmit interrupt mask register. In all cases, the enables for each interrupt event align with the isochronous transmit interrupt 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

The isochronous receive interrupt event 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 isochRx (bit 7) interrupt in the interrupt event register (offset 80h/84h, see Section 9.21) 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 clear a bit in this register 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 the contents of the isochronous receive interrupt event register bitwise ANDed with the isochronous receive interrupt mask register when read)
Default: 0000 000Xh

Table 9–18. Isochronous Receive Interrupt Event Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–4	RSVD	R	Reserved. Bits 31–4 return 0s when read.
3	isoRecv3	RSC	Isochronous receive channel 3 caused the interrupt event register bit 7 (isochRx) interrupt.
2	isoRecv2	RSC	Isochronous receive channel 2 caused the interrupt event register bit 7 (isochRx) interrupt.
1	isoRecv1	RSC	Isochronous receive channel 1 caused the interrupt event register bit 7 (isochRx) interrupt.
0	isoRecv0	RSC	Isochronous receive channel 0 caused the interrupt event register bit 7 (isochRx) interrupt.

9.26 Isochronous Receive Interrupt Mask Register

The isochronous receive interrupt mask 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 the contents of the isochronous receive interrupt mask register. In all cases the enables for each interrupt event align with the isochronous receive interrupt 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)

The fairness control 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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0

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

Table 9–19. Fairness Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–8	RSVD	R	Reserved. Bits 31–8 return 0s when read.
7–0	pri_req	R/W	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 a fairness interval.

9.28 Link Control Register

The link control set/clear register provides the control flags that enable and configure the link core protocol portions of the PCI4410A device. 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 Description

BIT	SIGNAL	TYPE	FUNCTION
31–23	RSVD	R	Reserved. Bits 31–23 return 0s when read.
22	cycleSource	RSC	When bit 22 is set to 1, the cycle timer uses an external source (CYCLEIN) to determine when to roll over the cycle timer. When this bit is cleared, the cycle timer rolls over when the timer reaches 3072 cycles of the 24.576-MHz clock (125 µs).
21	cycleMaster	RSCU	When bit 21 is set to 1 and the PHY has notified the PCI4410A device that it is root, the PCI4410A device generates a cycle start packet every time the cycle timer rolls over, based on the setting of bit 22 (cycleSource). When bit 21 is cleared, the OHCI-Lynx™ accepts received cycle start packets to maintain synchronization with the node that is sending them. Bit 21 is automatically cleared when bit 25 (cycleTooLong) in the interrupt event register (offset 80h/84h, see Section 9.21) is set. Bit 21 cannot be set to 1 until bit 25 (cycleTooLong) is cleared.
20	CycleTimerEnable	RSC	When bit 20 is set to 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 cleared, the cycle timer offset does not count.
19–11	RSVD	R	Reserved. Bits 19–11 return 0s when read.
10	RcvPhyPkt	RSC	When bit 10 is set to 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 bit 9 is set to 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

The node identification register contains the address of the node on which the OHCI-Lynx™ chip resides, and indicates the valid node number status. The 16-bit combination of the busNumber field (bits 15–6) and the NodeNumber field (bits 5–0) 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 FFXXh

Table 9–21. Node Identification Register Description

BIT	SIGNAL	TYPE	FUNCTION
31	iDValid	RU	Bit 31 indicates whether or not the PCI4410A device has a valid node number. It is cleared when a 1394 bus reset is detected and set to 1 when the PCI4410A device receives a new node number from the PHY.
30	root	RU	Bit 30 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	Bit 27 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 field is used to identify the specific 1394 bus the PCI4410A device belongs to when multiple 1394-compatible buses are connected via a bridge.
5–0	NodeNumber	RU	This field 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, software should not set bit 15 (run) of the asynchronous context control register (see Section 9.37) for either of the AT DMA contexts.

9.30 PHY Control Register

The PHY control 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/Write, Read-only
Offset: ECh
Default: XXXX 0XXXh

Table 9–22. PHY Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
31	rdDone	RU	This bit is cleared to 0 by the PCI4410A device 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. Bits 15 and 14 must not be set simultaneously.
14	wrReg	RWU	This bit is set to 1 by software to initiate a write request to a PHY register, and is cleared by hardware when the request has been sent. Bits 14 and 15 must not be set simultaneously.
13–12	RSVD	R	Reserved. Bits 13 and 12 return 0s 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

The isochronous cycle timer register indicates the current cycle number and offset. When the PCI4410A device is cycle master, this register is transmitted with the cycle start message. When the PCI4410A device 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 Description

BIT	SIGNAL	TYPE	FUNCTION
31–25	cycleSeconds	RWU	This field counts seconds [rollovers from bits 24–12 (cycleCount field)] modulo 128.
24–12	cycleCount	RWU	This field counts cycles [rollovers from bits 11–0 (cycleOffset field)] 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, this bit must be cleared at each tick of the external clock.

9.32 Asynchronous Request Filter High Register

The asynchronous request filter high 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, 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 PCI4410A device. 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 Description

BIT	SIGNAL	TYPE	FUNCTION
31	asynReqAllBuses	RSC	If bit 31 is set to 1, all asynchronous requests received by the PCI4410A device from nonlocal bus nodes are accepted.
30	asynReqResource62	RSC	If bit 30 is set to 1 for local bus node number 62, asynchronous requests received by the PCI4410A device from that node are accepted.
29	asynReqResource61	RSC	If bit 29 is set to 1 for local bus node number 61, asynchronous requests received by the PCI4410A device from that node are accepted.
28	asynReqResource60	RSC	If bit 28 is set to 1 for local bus node number 60, asynchronous requests received by the PCI4410A device from that node are accepted.
27	asynReqResource59	RSC	If bit 27 is set to 1 for local bus node number 59, asynchronous requests received by the PCI4410A device from that node are accepted.
26	asynReqResource58	RSC	If bit 26 is set to 1 for local bus node number 58, asynchronous requests received by the PCI4410A device from that node are accepted.
25	asynReqResource57	RSC	If bit 25 is set to 1 for local bus node number 57, asynchronous requests received by the PCI4410A device from that node are accepted.
24	asynReqResource56	RSC	If bit 24 is set to 1 for local bus node number 56, asynchronous requests received by the PCI4410A device from that node are accepted.
23	asynReqResource55	RSC	If bit 23 is set to 1 for local bus node number 55, asynchronous requests received by the PCI4410A device from that node are accepted.
22	asynReqResource54	RSC	If bit 22 is set to 1 for local bus node number 54, asynchronous requests received by the PCI4410A device from that node are accepted.
21	asynReqResource53	RSC	If bit 21 is set to 1 for local bus node number 53, asynchronous requests received by the PCI4410A device from that node are accepted.
20	asynReqResource52	RSC	If bit 20 is set to 1 for local bus node number 52, asynchronous requests received by the PCI4410A device from that node are accepted.

Table 9–24. Asynchronous Request Filter High Register Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION
19	asynReqResource51	RSC	If bit 19 is set to 1 for local bus node number 51, asynchronous requests received by the PCI4410A device from that node are accepted.
18	asynReqResource50	RSC	If bit 18 is set to 1 for local bus node number 50, asynchronous requests received by the PCI4410A device from that node are accepted.
17	asynReqResource49	RSC	If bit 17 is set to 1 for local bus node number 49, asynchronous requests received by the PCI4410A device from that node are accepted.
16	asynReqResource48	RSC	If bit 16 is set to 1 for local bus node number 48, asynchronous requests received by the PCI4410A device from that node are accepted.
15	asynReqResource47	RSC	If bit 15 is set to 1 for local bus node number 47, asynchronous requests received by the PCI4410A device from that node are accepted.
14	asynReqResource46	RSC	If bit 14 is set to 1 for local bus node number 46, asynchronous requests received by the PCI4410A device from that node are accepted.
13	asynReqResource45	RSC	If bit 13 is set to 1 for local bus node number 45, asynchronous requests received by the PCI4410A device from that node are accepted.
12	asynReqResource44	RSC	If bit 12 is set to 1 for local bus node number 44, asynchronous requests received by the PCI4410A device from that node are accepted.
11	asynReqResource43	RSC	If bit 11 is set to 1 for local bus node number 43, asynchronous requests received by the PCI4410A device from that node are accepted.
10	asynReqResource42	RSC	If bit 10 is set to 1 for local bus node number 42, asynchronous requests received by the PCI4410A device from that node are accepted.
9	asynReqResource41	RSC	If bit 9 is set to 1 for local bus node number 41, asynchronous requests received by the PCI4410A device from that node are accepted.
8	asynReqResource40	RSC	If bit 8 is set to 1 for local bus node number 40, asynchronous requests received by the PCI4410A device from that node are accepted.
7	asynReqResource39	RSC	If bit 7 is set to 1 for local bus node number 39, asynchronous requests received by the PCI4410A device from that node are accepted.
6	asynReqResource38	RSC	If bit 6 is set to 1 for local bus node number 38, asynchronous requests received by the PCI4410A device from that node are accepted.
5	asynReqResource37	RSC	If bit 5 is set to 1 for local bus node number 37, asynchronous requests received by the PCI4410A device from that node are accepted.
4	asynReqResource36	RSC	If bit 4 is set to 1 for local bus node number 36, asynchronous requests received by the PCI4410A device from that node are accepted.
3	asynReqResource35	RSC	If bit 3 is set to 1 for local bus node number 35, asynchronous requests received by the PCI4410A device from that node are accepted.
2	asynReqResource34	RSC	If bit 2 is set to 1 for local bus node number 34, asynchronous requests received by the PCI4410A device from that node are accepted.
1	asynReqResource33	RSC	If bit 1 is set to 1 for local bus node number 33, asynchronous requests received by the PCI4410A device from that node are accepted.
0	asynReqResource32	RSC	If bit 0 is set to 1 for local bus node number 32, asynchronous requests received by the PCI4410A device from that node are accepted.

9.33 Asynchronous Request Filter Low Register

The asynchronous request filter low 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 Description

BIT	SIGNAL	TYPE	FUNCTION
31	asynReqResource31	RSC	If bit 31 is set to 1 for local bus node number 31, asynchronous requests received by the PCI4410A device from that node are accepted.
30	asynReqResource30	RSC	If bit 30 is set to 1 for local bus node number 30, asynchronous requests received by the PCI4410A device from that node are accepted.
⋮	⋮	⋮	Bits 29 through 2 follow the same pattern.
1	asynReqResource1	RSC	If bit 1 is set to 1 for local bus node number 1, asynchronous requests received by the PCI4410A device from that node are accepted.
0	asynReqResource0	RSC	If bit 0 is set to 1 for local bus node number 0, asynchronous requests received by the PCI4410A device from that node are accepted.

9.34 Physical Request Filter High Register

The physical request filter high 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, 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 Description

BIT	SIGNAL	TYPE	FUNCTION
31	physReqAllBusses	RSC	If bit 31 is set to 1, all asynchronous requests received by the PCI4410A device from nonlocal bus nodes are accepted.
30	physReqResource62	RSC	If bit 30 is set to 1 for local bus node number 62, physical requests received by the PCI4410A device from that node are handled through the physical request context.
29	physReqResource61	RSC	If bit 29 is set to 1 for local bus node number 61, physical requests received by the PCI4410A device from that node are handled through the physical request context.
28	physReqResource60	RSC	If bit 28 is set to 1 for local bus node number 60, physical requests received by the PCI4410A device from that node are handled through the physical request context.
27	physReqResource59	RSC	If bit 27 is set to 1 for local bus node number 59, physical requests received by the PCI4410A device from that node are handled through the physical request context.
26	physReqResource58	RSC	If bit 26 is set to 1 for local bus node number 58, physical requests received by the PCI4410A device from that node are handled through the physical request context.
25	physReqResource57	RSC	If bit 25 is set to 1 for local bus node number 57, physical requests received by the PCI4410A device from that node are handled through the physical request context.
24	physReqResource56	RSC	If bit 24 is set to 1 for local bus node number 56, physical requests received by the PCI4410A device from that node are handled through the physical request context.
23	physReqResource55	RSC	If bit 23 is set to 1 for local bus node number 55, physical requests received by the PCI4410A device from that node are handled through the physical request context.
22	physReqResource54	RSC	If bit 22 is set to 1 for local bus node number 54, physical requests received by the PCI4410A device from that node are handled through the physical request context.
21	physReqResource53	RSC	If bit 21 is set to 1 for local bus node number 53, physical requests received by the PCI4410A device from that node are handled through the physical request context.
20	physReqResource52	RSC	If bit 20 is set to 1 for local bus node number 52, physical requests received by the PCI4410A device from that node are handled through the physical request context.
19	physReqResource51	RSC	If bit 19 is set to 1 for local bus node number 51, physical requests received by the PCI4410A device from that node are handled through the physical request context.
18	physReqResource50	RSC	If bit 18 is set to 1 for local bus node number 50, physical requests received by the PCI4410A device from that node are handled through the physical request context.

Table 9–26. Physical Request Filter High Register Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION
17	physReqResource49	RSC	If bit 17 is set to 1 for local bus node number 49, physical requests received by the PCI4410A device from that node are handled through the physical request context.
16	physReqResource48	RSC	If bit 16 is set to 1 for local bus node number 48, physical requests received by the PCI4410A device from that node are handled through the physical request context.
15	physReqResource47	RSC	If bit 15 is set to 1 for local bus node number 47, physical requests received by the PCI4410A device from that node are handled through the physical request context.
14	physReqResource46	RSC	If bit 14 is set to 1 for local bus node number 46, physical requests received by the PCI4410A device from that node are handled through the physical request context.
13	physReqResource45	RSC	If bit 13 is set to 1 for local bus node number 45, physical requests received by the PCI4410A device from that node are handled through the physical request context.
12	physReqResource44	RSC	If bit 12 is set to 1 for local bus node number 44, physical requests received by the PCI4410A device from that node are handled through the physical request context.
11	physReqResource43	RSC	If bit 11 is set to 1 for local bus node number 43, physical requests received by the PCI4410A device from that node are handled through the physical request context.
10	physReqResource42	RSC	If bit 10 is set to 1 for local bus node number 42, physical requests received by the PCI4410A device from that node are handled through the physical request context.
9	physReqResource41	RSC	If bit 9 is set to 1 for local bus node number 41, physical requests received by the PCI4410A device from that node are handled through the physical request context.
8	physReqResource40	RSC	If bit 8 is set to 1 for local bus node number 40, physical requests received by the PCI4410A device from that node are handled through the physical request context.
7	physReqResource39	RSC	If bit 7 is set to 1 for local bus node number 39, physical requests received by the PCI4410A device from that node are handled through the physical request context.
6	physReqResource38	RSC	If bit 6 is set to 1 for local bus node number 38, physical requests received by the PCI4410A device from that node are handled through the physical request context.
5	physReqResource37	RSC	If bit 5 is set to 1 for local bus node number 37, physical requests received by the PCI4410A device from that node are handled through the physical request context.
4	physReqResource36	RSC	If bit 4 is set to 1 for local bus node number 36, physical requests received by the PCI4410A device from that node are handled through the physical request context.
3	physReqResource35	RSC	If bit 3 is set to 1 for local bus node number 35, physical requests received by the PCI4410A device from that node are handled through the physical request context.
2	physReqResource34	RSC	If bit 2 is set to 1 for local bus node number 34, physical requests received by the PCI4410A device from that node are handled through the physical request context.
1	physReqResource33	RSC	If bit 1 is set to 1 for local bus node number 33, physical requests received by the PCI4410A device from that node are handled through the physical request context.
0	physReqResource32	RSC	If bit 0 is set to 1 for local bus node number 32, physical requests received by the PCI4410A device from that node are handled through the physical request context.

9.35 Physical Request Filter Low Register

The physical request filter low 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, 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, 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 Description

BIT	SIGNAL	TYPE	FUNCTION
31	physReqResource31	RSC	If bit 31 is set to 1 for local bus node number 31, physical requests received by the PCI4410A device from that node are handled through the physical request context.
30	physReqResource30	RSC	If bit 30 is set to 1 for local bus node number 30, physical requests received by the PCI4410A device from that node are handled through the physical request context.
⋮	⋮	⋮	Bits 29 through 2 follow the same pattern.
1	physReqResource1	RSC	If bit 1 is set to 1 for local bus node number 1, physical requests received by the PCI4410A device from that node are handled through the physical request context.
0	physReqResource0	RSC	If bit 0 is set to 1 for local bus node number 0, physical requests received by the PCI4410A device 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 returns all 0s when read.

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

The asynchronous context control 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 Description

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 PCI4410A device changes this bit only 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 PCI4410A device to continue or resume descriptor processing. The PCI4410A device clears this bit on every descriptor fetch.
11	dead	RU	The PCI4410A device sets this bit to 1 when it encounters a fatal error, and clears the bit when software resets bit 15 (run).
10	active	RU	The PCI4410A device 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/s 001b = 200 Mbits/s 010b = 400 Mbits/s 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

The asynchronous context command pointer register contains a pointer to the address of the first descriptor block that the PCI4410A device accesses when software enables the context by setting bit 15 (run) of the asynchronous context control register (see Section 9.37) 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 Description

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 field (bits 31–4) is not valid.

9.39 Isochronous Transmit Context Control Register

The isochronous transmit context control 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, …, 7). 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 Description

BIT	SIGNAL	TYPE	FUNCTION
31	cycleMatchEnable	RSCU	When bit 31 is set to 1, processing occurs such that the packet described by the context first descriptor block is transmitted in the cycle whose number is specified in the cycleMatch field (bits 30–16). The cycleMatch field (bits 30–16) must match the low-order two bits of cycleSeconds and the 13-bit cycleCount field in the cycle start packet that is sent or received immediately before isochronous transmission begins. Since the isochronous transmit DMA controller may work ahead, the processing of the first descriptor block may begin slightly in advance of the actual cycle in which the first packet is transmitted. The effects of this bit, however, are impacted by the values of other bits in this register and are explained in the *1394 Open Host Controller Interface Specification.* Once the context has become active, hardware clears this bit.
30–16	cycleMatch	RSC	This field contains a 15-bit value, corresponding to the low-order two bits of the isochronous cycle timer register (OHCI offset F0h, see Section 9.31) cycleSeconds field (bits 31–25) and the cycleCount field (bits 24–12). If bit 31 (cycleMatchEnable) is set, then this isochronous transmit DMA context becomes enabled for transmits when the low-order two bits of the isochronous cycle timer register cycleSeconds field (bits 31–25) and the cycleCount field (bits 24–12) value equal this field (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 PCI4410A device changes this bit only 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 PCI4410A device to continue or resume descriptor processing. The PCI4410A device clears this bit on every descriptor fetch.
11	dead	RU	The PCI4410A device sets this bit to 1 when it encounters a fatal error, and clears the bit when software resets bit 15 (run) to 0.
10	active	RU	The PCI4410A device 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 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

The isochronous transmit context command pointer register contains a pointer to the address of the first descriptor block that the PCI4410A device accesses when software enables an isochronous transmit context by setting bit 15 (run) in the isochronous transmit context control register (see Section 9.39) to 1. The n value in the following register addresses indicates the context number (n = 0, 1, 2, 3, …, 7).

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

The isochronous receive context control 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 Description

BIT	SIGNAL	TYPE	FUNCTION
31	bufferFill	RSC	When this bit is set to 1, received packets are placed back-to-back to completely fill each receive buffer. When this bit is cleared, each received packet is placed in a single buffer. If bit 28 (multiChanMode) is set to 1, this bit must also be set to 1. The value of this bit must not be changed while bit 10 (active) or bit 15 (run) is set to 1.
30	isochHeader	RSC	When this bit is set to 1, received isochronous packets include the complete 4-byte isochronous packet header seen by the link layer. The end of the packet is marked with a xferStatus in the first doublet, and a 16-bit timeStamp indicating the time of the most recently received (or sent) cycleStart packet. When this bit is cleared, the packet header is stripped from received isochronous packets. The packet header, if received, immediately precedes the packet payload. The value of this bit must not be changed while bit 10 (active) or bit 15 (run) is set to 1.
29	cycleMatchEnable	RSCU	When this bit is set to 1, the context begins running only when the 13-bit cycleMatch field (bits 24–12) in the isochronous receive context match register (see Section 9.43) matches the 13-bit cycleCount field 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 this bit. The value of this bit must not be changed while bit 10 (active) or bit 15 (run) is set to 1.
28	multiChanMode	RSC	When this bit is set to 1, the corresponding isochronous receive DMA context receives packets for all isochronous channels enabled in the isochronous receive channel mask high (offset 70h/74h, see Section 9.19) and isochronous receive channel mask low (offset 78h/7Ch, see Section 9.20) registers. The isochronous channel number specified in the isochronous receive context match register (see Section 9.43) is ignored. When this bit is cleared, the isochronous receive DMA context receives packets for that single channel. Only one isochronous receive DMA context can use the isochronous receive channel mask registers (see Sections 9.19 and 9.20). If more that one isochronous receive context control register has this bit set to 1, the results are undefined. The value of this bit must not be changed while bit 10 (active) or bit 15 (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 PCI4410A device changes this bit only 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 PCI4410A device to continue or resume descriptor processing. The PCI4410A device clears this bit on every descriptor fetch.
11	dead	RU	The PCI4410A device sets this bit to 1 when it encounters a fatal error, and clears the bit when software resets bit 15 (run).
10	active	RU	The PCI4410A device 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 indicates the speed at which the packet was received. 000b = 100 Mbits/s 001b = 200 Mbits/s 010b = 400 Mbits/s All other values are reserved.
4–0	event code	RU	For bufferFill mode, possible values are: ack_complete, evt_descriptor_read, evt_data_write, and evt_unknown. Packets with data errors (either dataLength mismatches or dataCRC errors) and packets for which a FIFO overrun occurred are backed out. For packet-per-buffer mode, possible values are: ack_complete, ack_data_error, evt_long_packet, evt_overrun, evt_descriptor_read, evt_data_write, and evt_unknown.

9.42 Isochronous Receive Context Command Pointer Register

The isochronous receive context command pointer register contains a pointer to the address of the first descriptor block that the PCI4410A device accesses when software enables an isochronous receive context by setting bit 15 (run) of the isochronous receive context control register (see Section 9.41). 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

The isochronous receive context match 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/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	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 Description

BIT	SIGNAL	TYPE	FUNCTION
31	tag3	R/W	If this bit is set to 1, this context matches on isochronous receive packets with a tag field of 11b.
30	tag2	R/W	If this bit is set to 1, this context matches on isochronous receive packets with a tag field of 10b.
29	tag1	R/W	If this bit is set to 1, this context matches on isochronous receive packets with a tag field of 01b.
28	tag0	R/W	If this bit is set to 1, this context matches on isochronous receive packets with a tag field of 00b.
27	RSVD	R	Reserved. Bit 27 returns 0 when read.
26–12	cycleMatch	R/W	This field contains a 15-bit value corresponding to the low-order two bits of cycleSeconds and the 13-bit cycleCount field in the cycleStart packet. If bit 29 (cycleMatchEnable) of the isochronous receive context control register (see Section 9.41) is set, then this context is enabled for receives when the two low-order bits of the isochronous cycle timer register (OHCI offset F0h, see Section 9.31) cycleSeconds field (bits 31–25) and cycleCount field (bits 24–12) value equal this field (cycleMatch) value.
11–8	sync	R/W	This 4-bit field is compared to the sync field of each isochronous 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, 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 bit 28 (tag0), bit 30 (tag2), and bit 31 (tag3) without any additional restrictions. If this bit is cleared, this context matches on isochronous receive packets as specified in bits 31–28 (tag3–tag0) with no additional restrictions.
5–0	channelNumber	R/W	This 6-bit field indicates the isochronous channel number for which this isochronous receive 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 75, 76, 77, 78, 80, 81, 83, 84, 85, 86, 87, 88, 121, and 122 for the PDV packaged device; and M18, M19, P9, P10, P11, R11, U10, U11, U12, V10, V12, W10, W11, and W12 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 123, 165, 179, and 185 for the PDV packaged device; and A9, E13, F11, and L19 for the GHK packaged device (card detect and voltage sense terminals).

¶ 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		TERMINALS	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	
		$\overline{SERR}$		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 terminals, $V_I = V_{CCP}$. For PC Card terminals, $V_I = V_{CCCB}$. For miscellaneous terminals, $V_I = V_{CCI}$.
‡ For I/O terminals, 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 $\overline{\text{PRST}}$	$t_{rst\text{-}clk}$		100		µs

10.5 PCI Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature

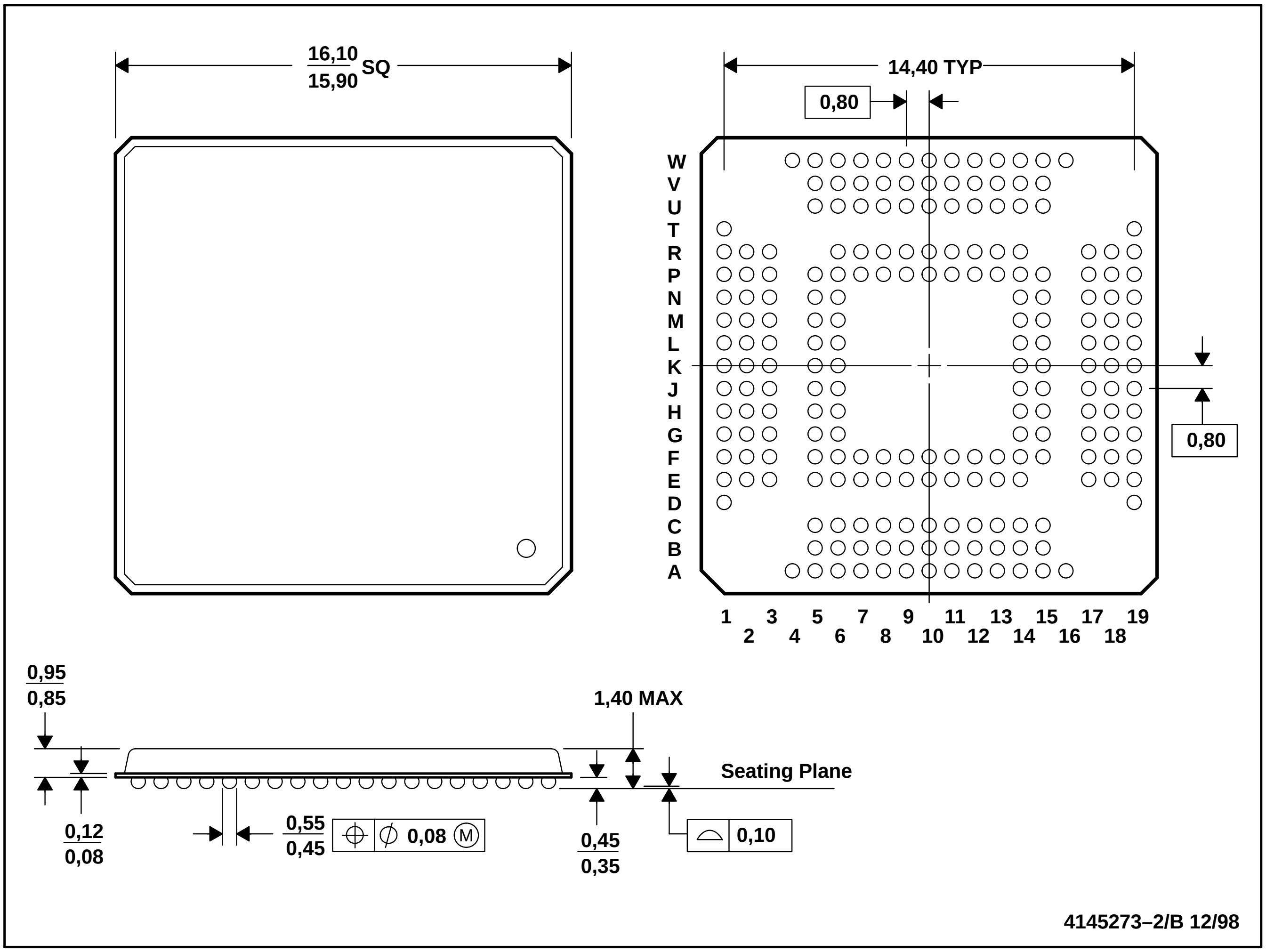
PARAMETER			ALTERNATE SYMBOL	TEST CONDITIONS	MIN	MAX	UNIT
t_{pd}	Propagation delay time, See Note 4	PCLK-to-shared signal valid delay time	t_{val}	C_L = 50 pF, See Note 4		11	ns
		PCLK-to-shared signal invalid delay time	t_{inv}		2		
t_{en}	Enable time, high impedance-to-active delay time from PCLK		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

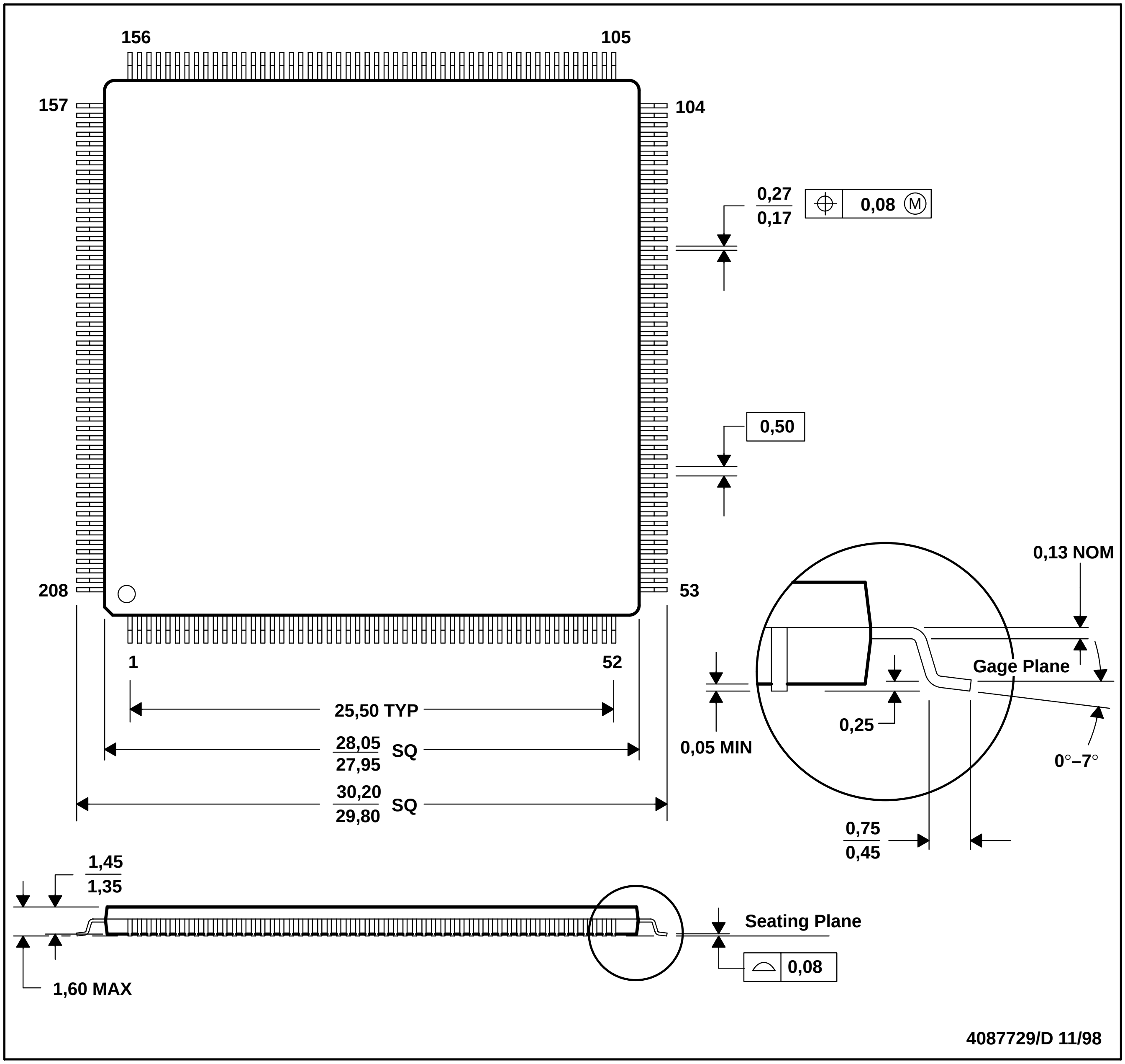
The PCI4410A device is packaged in either a 209-ball GHK MicroStar BGA™ or a 208-pin PDV package. The PCI4410A device is a single-socket CardBus bridge with an integrated OHCI link. The following shows the mechanical dimensions for the GHK and PDV packages.

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. MicroStar BGA™ configuration.

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