TEXAS INSTRUMENTS

PCI4450 GFN/GJG
PC Card and OHCI Controller

Data Sheet

2000

PCIBus Solutions

TEXAS INSTRUMENTS

Printed in U.S.A.
04/00

SCPS046A

PCI4450 GFN/GJG
PC Card and OHCI Controller
Data Manual

Literature Number: SCPS046A
September 1999

Contents

List of Illustrations

List of Tables

1 Introduction

1.1 Description

The Texas Instruments PCI4450 is an integrated dual-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.

The PCI4450 is a three-function PCI device compliant with *PCI Local Bus Specification 2.2*. Functions 0 and 1 provide the independent PC Card socket controllers compliant with the *1997 PC Card Standard*. The PCI4450 provides features that make it the best choice for bridging between the PCI bus and PC Cards, and supports any combination of 16-bit and CardBus PC Cards in the two sockets, powered at 5 V or 3.3 V, as required.

All card signals are internally buffered to allow hot insertion and removal without external buffering. The PCI4450 is register compatible with the Intel 82365SL−DF ExCA controller. The PCI4450 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 PCI4450 can be programmed to accept posted writes to improve bus utilization.

Function 2 of the PCI4450 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 per second. Deep FIFOs are provided to buffer 1394 data and accommodate large host bus latencies. The PCI4450 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 PCI4450 the best-in-class 1394 Open HCI solution.

The PCI4450 provides an internally buffered zoomed video (ZV) path. This reduces the design effort of PC board manufacturers to add a ZV-compatible solution and ensures compliance with the CardBus loading specifications.

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

The PCI4450 is compliant with the latest *PCI Bus Power Management Specification*, and provides several low-power modes which enable the host power system to further reduce power consumption. The *PC Card (CardBus) Controller* and *IEEE 1394 Host Controller Device Class Specifications* required for Microsoft™ OnNow Power Management are supported. Furthermore, an advanced complementary metal-oxide semiconductor (CMOS) process achieves low system power consumption.

1.2 Features

The PCI4450 supports the following features:

- *1997 PC Card Standard* compliant
- *PCI Bus Power Management Interface Specification 1.1* compliant

- ACPI 1.0 compliant
- *PCI Local Bus Specification Revision 2.1/2.2* compliant
- *PC 99* compliant
- Compliant with the *PCI Bus Interface Specification for PCI-to-CardBus Bridges*
- Fully compliant with the *PCI Bus Power Management Specification for PCI to CardBus Bridges*
- Ultra zoomed video
- Zoomed video autodetect
- Advanced filtering on card detect lines provides 90 microseconds of noise immunity.
- Programmable D3 status terminal
- Internal ring oscillator
- 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 PC Card16 cards and 3.3-V CardBus cards
- Supports two PC Card or CardBus slots with hot insertion and removal
- Uses serial interface to TI TPS2206 dual power switch
- Supports 132 Mbyte/sec. burst transfers to maximize data throughput on both the PCI bus and the CardBus bus
- Supports serialized IRQ with PCI interrupts
- Eight programmable multifunction terminals
- Interrupt modes supported: serial ISA/serial PCI, serial ISA/parallel PCI, parallel PCI only
- Serial EEPROM interface for loading subsystem ID and subsystem vendor ID
- Supports zoomed video with internal buffering
- Dedicated terminal for PCI $\overline{\text{CLKRUN}}$
- Four general-purpose event registers
- Multifunction PCI device with separate configuration space for each socket
- Five PCI memory windows and two I/O windows available to each PC Card16 socket
- Two I/O windows and two memory windows available to each CardBus socket
- ExCA-compatible registers are mapped in memory or I/O space
- Supports distributed DMA and PC/PCI DMA
- Intel™ 82365SL-DF register compatible
- Supports 16-bit DMA on both PC Card sockets
- Supports ring indicate, $\overline{\text{SUSPEND}}$, and PCI $\overline{\text{CLKRUN}}$
- Advanced submicron, low-power CMOS technology
- Provides VGA/palette memory and I/O, and subtractive decoding options

- LED activity terminals
- Supports PCI bus lock ($\overline{\text{LOCK}}$)
- Supplied in a 256-terminal BGA or 257-terminal MicroStar BGA
- 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 IEEE 1394-1995 compliant and compatible with proposal 1394a
- Supports serial bus data rates of 100, 200, and 400 Mbits/second
- Provides bus-hold buffers on the PHY-link I/F for low-cost single-capacitor isolation

1.3 Related Documents

- *1997 PC Card Standard*
- ACPI 1.0
- *IEEE 1394 Open Host Controller Interface (OHCI) Specification*
- *P1394 Draft Standard for a High Performance Serial Bus (Supplement)*
- *P1394 Standard for a High Performance Serial Bus* (IEEE 1394-1995)
- *PC 98/99*
- *PCI Bus Interface Specification for PCI-to-CardBus Bridges*
- *PCI Bus Power Management Interface Specification 1.1*
- *PCI Bus Power Management Specification for PCI to CardBus Bridges*
- *PCI Local Bus Specification Revision 2.1/2.2*
- *PCI Bus Power Management Specification for PCI to CardBus Bridges*

1.4 Ordering Information

ORDERING NUMBER	NAME	VOLTAGE	PACKAGE
PCI4450	PC-Card and OHCI controller	3.3V-, 5V-tolerant I/Os	256-terminal PBGA 257-terminal PBGA

2 Terminal Descriptions

The PCI4450 is packaged in either a 256-ball BGA as shown in Figure 2–1 or a 257-ball MicroStar BGA™ package as shown in Figure 2–2.

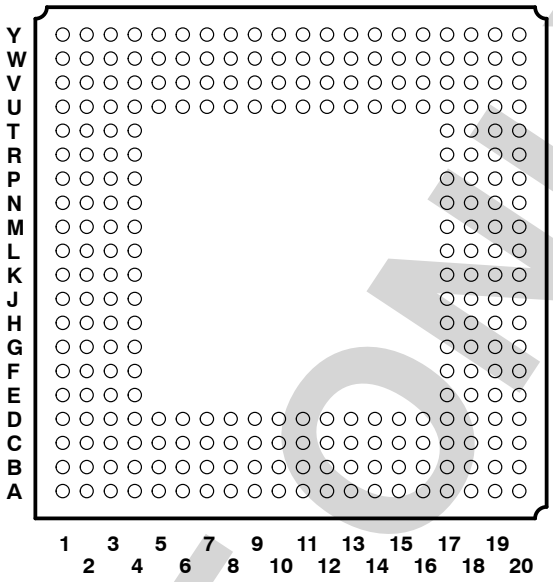


Figure 2–1. PCI4450 GFN Terminal Diagram

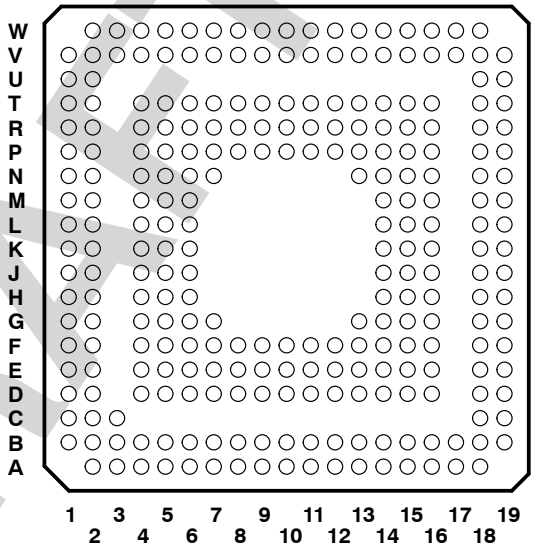


Figure 2–2. PCI4450 GJG Terminal Diagram

MicroStar BGA is a trademark of Texas Instruments.
Other trademarks are the property of their respective owners.

Signal names and their terminal assignments are shown in Table 2–1 through Table 2–6. The GFN package terminals are listed in Table 2–1 through Table 2–3, and the GJG package terminals are listed in Table 2–4 through Table 2–6. Table 2–1 and Table 2–4 list CardBus signal names and 16-bit PC Card signal names arranged in order by terminal number. Table 2–2 and Table 2–5 list terminals arranged in order by CardBus signal names. Table 2–3 and Table 2–6 list terminals arranged in order by 16-bit PC Card signal names.

Table 2–1. GFN Terminals Sorted Alphanumerically for CardBus // 16-Bit Signals and OHCI

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CARDBUS	16-BIT		CARDBUS	16-BIT		CARDBUS	16-BIT
A1	GND	GND	C4	A_CAD12	A_ADDR11	E19	V_{CCB}	V_{CCB}
A2	A_CAD16	A_ADDR17	C5	A_CAD9	A_ADDR10	E20	B_CAD9	B_ADDR10
A3	A_CAD11	A_$\overline{OE}$	C6	A_CAD8	A_DATA15	F1	A_CAD17	A_ADDR24
A4	A_CC/$\overline{BE0}$	A_$\overline{CE1}$	C7	A_CAD6	A_DATA13	F2	A_CC/$\overline{BE2}$	A_ADDR12
A5	A_RSVD	A_DATA14	C8	A_CAD2	A_DATA11	F3	V_{CCA}	V_{CCA}
A6	A_CAD3	A_DATA5	C9	B_RSVD	B_DATA2	F4	V_{CC}	V_{CC}
A7	A_CAD1	A_DATA4	C10	B_CAD27	B_DATA0	F17	V_{CC}	V_{CC}
A8	A_$\overline{CCD1}$	A_$\overline{CD1}$	C11	B_CAUDIO	B_BVD2($\overline{SPKR}$)	F18	B_CAD11	B_$\overline{OE}$
A9	B_CAD29	B_DATA1	C12	B_CAD26	B_ADDR0	F19	B_CC/$\overline{BE0}$	B_$\overline{CE1}$
A10	B_$\overline{CCLKRUN}$	B_WP($\overline{IOIS16}$)	C13	B_CC/$\overline{BE3}$	B_$\overline{REG}$	F20	B_CAD7	B_DATA7
A11	B_CSTSCHG	B_BVD1($\overline{STSCHG}$/$\overline{RI}$)	C14	B_CAD22	B_ADDR4	G1	A_CVS2	A_$\overline{VS2}$
A12	B_$\overline{CINT}$	B_READY($\overline{IREQ}$)	C15	B_CVS2	B_$\overline{VS2}$	G2	A_CAD19	A_ADDR25
A13	B_CAD24	B_ADDR2	C16	B_CAD17	B_ADDR24	G3	A_CAD18	A_ADDR7
A14	B_CAD23	B_ADDR3	C17	B_$\overline{CTRDY}$	B_ADDR22	G4	A_$\overline{CFRAME}$	A_ADDR23
A15	B_CAD21	B_ADDR5	C18	B_$\overline{CBLOCK}$	B_ADDR19	G17	B_CAD10	B_$\overline{CE2}$
A16	B_CAD19	B_ADDR25	C19	B_RSVD	B_ADDR18	G18	B_CAD8	B_DATA15
A17	B_CC/$\overline{BE2}$	B_ADDR12	C20	B_CAD14	B_ADDR9	G19	B_RSVD	B_DATA14
A18	B_$\overline{CFRAME}$	B_ADDR23	D1	A_$\overline{CDEVSEL}$	A_ADDR21	G20	B_CAD5	B_DATA6
A19	B_$\overline{CGNT}$	B_$\overline{WE}$	D2	A_$\overline{CBLOCK}$	A_ADDR19	H1	A_CAD21	A_ADDR5
A20	B_$\overline{CSTOP}$	B_ADDR20	D3	A_$\overline{CPERR}$	A_ADDR14	H2	A_$\overline{CRST}$	A_RESET
B1	A_RSVD	A_ADDR18	D4	GND	GND	H3	A_CAD20	A_ADDR6
B2	A_CAD14	A_ADDR9	D5	A_CAD13	A_$\overline{IORD}$	H4	GND	GND
B3	A_CAD15	A_$\overline{IOWR}$	D6	V_{CC}	V_{CC}	H17	GND	GND
B4	A_CAD10	A_$\overline{CE2}$	D7	A_CAD7	A_DATA7	H18	B_CAD6	B_DATA13
B5	V_{CCA}	V_{CCA}	D8	GND	GND	H19	B_CAD3	B_DATA5
B6	A_CAD5	A_DATA6	D9	B_CAD31	B_DATA10	H20	B_CAD4	B_DATA12
B7	A_CAD4	A_DATA12	D10	B_CAD28	B_DATA8	J1	A_CC/$\overline{BE3}$	A_$\overline{REG}$
B8	A_CAD0	A_DATA3	D11	V_{CC}	V_{CC}	J2	A_CAD23	A_ADDR3
B9	B_CAD30	B_DATA9	D12	B_CAD25	B_ADDR1	J3	A_$\overline{CREQ}$	A_$\overline{INPACK}$
B10	B_$\overline{CCD2}$	B_$\overline{CD2}$	D13	GND	GND	J4	A_CAD22	A_ADDR4
B11	B_$\overline{CSERR}$	B_$\overline{WAIT}$	D14	B_CAD20	B_ADDR6	J17	B_CAD1	B_DATA4
B12	B_CVS1	B_$\overline{VS1}$	D15	V_{CC}	V_{CC}	J18	B_CAD2	B_DATA11
B13	V_{CCB}	V_{CCB}	D16	B_$\overline{CIRDY}$	B_ADDR15	J19	B_CAD0	B_DATA3
B14	B_$\overline{CREQ}$	B_$\overline{INPACK}$	D17	GND	GND	J20	B_$\overline{CCD1}$	B_$\overline{CD1}$
B15	B_$\overline{CRST}$	B_RESET	D18	B_CC/$\overline{BE1}$	B_ADDR8	K1	A_CAD26	A_ADDR0
B16	B_CAD18	B_ADDR7	D19	B_CAD15	B_$\overline{IOWR}$	K2	A_CAD24	A_ADDR2
B17	B_CCLK	B_ADDR16	D20	B_CAD13	B_$\overline{IORD}$	K3	A_CAD25	A_ADDR1
B18	B_$\overline{CDEVSEL}$	B_ADDR21	E1	A_$\overline{CIRDY}$	A_ADDR15	K4	V_{CC}	V_{CC}
B19	B_$\overline{CPERR}$	B_ADDR14	E2	A_$\overline{CTRDY}$	A_ADDR22	K17	PCLK	PCLK
B20	B_CPAR	B_ADDR13	E3	A_CCLK	A_ADDR16	K18	$\overline{CLKRUN}$	$\overline{CLKRUN}$
C1	A_$\overline{CGNT}$	A_$\overline{WE}$	E4	A_$\overline{CSTOP}$	A_ADDR20	K19	$\overline{PRST}$	$\overline{PRST}$
C2	A_CPAR	A_ADDR13	E17	B_CAD16	B_ADDR17	K20	$\overline{GNT}$	$\overline{GNT}$
C3	A_CC/$\overline{BE1}$	A_ADDR8	E18	B_CAD12	B_ADDR11	L1	A_CVS1	A_$\overline{VS1}$

Table 2–1. GFN Terminals Sorted Alphanumerically for CardBus // 16-Bit Signals and OHCI (continued)

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CARDBUS	16-BIT		CARDBUS	16-BIT		CARDBUS	16-BIT
L2	A_CINT	A_READY(IREQ)	T17	IRDY	IRDY	V20	FRAME	FRAME
L3	A_CSERR	A_WAIT	T18	AD16	AD16	W1	ZV_UV4	ZV_UV4
L4	V_{CCA}	V_{CCA}	T19	AD17	AD17	W2	ZV_UV6	ZV_UV6
L17	V_{CC}	V_{CC}	T20	AD18	AD18	W3	ZV_SCLK	ZV_SCLK
L18	AD31	AD31	U1	ZV_Y4	ZV_Y4	W4	ZV_MCLK	ZV_MCLK
L19	AD30	AD30	U2	ZV_Y6	ZV_Y6	W5	LPS	LPS
L20	REQ	REQ	U3	ZV_UV2	ZV_UV2	W6	PHY_CTL1	PHY_CTL1
M1	A_CAUDIO	A_BVD2(SPKR)	U4	GND	GND	W7	PHY_DATA1	PHY_DATA1
M2	A_CSTSCHG	A_BVD1(STSCHG/RI)	U5	ZV_SDATA	ZV_SDATA	W8	PHY_DATA4	PHY_DATA4
M3	A_CCLKRUN	A_WP(IOIS16)	U6	V_{CC}	V_{CC}	W9	MFUNC4	MFUNC4
M4	A_CCD2	A_CD2	U7	PHY_CTL0	PHY_CTL0	W10	SCL	SCL
M17	AD26	AD26	U8	GND	GND	W11	MFUNC0	MFUNC0
M18	AD27	AD27	U9	PHY_DATA6	PHY_DATA6	W12	LATCH	LATCH
M19	AD28	AD28	U10	V_{CC}	V_{CC}	W13	IRQSER	IRQSER
M20	AD29	AD29	U11	SUSPEND	SUSPEND	W14	AD2	AD2
N1	A_CAD27	A_DATA0	U12	CLOCK	CLOCK	W15	AD4	AD4
N2	A_CAD28	A_DATA8	U13	GND	GND	W16	C/BE0	C/BE0
N3	A_CAD29	A_DATA1	U14	AD6	AD6	W17	AD10	AD10
N4	GND	GND	U15	V_{CC}	V_{CC}	W18	AD14	AD14
N17	GND	GND	U16	AD12	AD12	W19	PAR	PAR
N18	C/BE3	C/BE3	U17	GND	GND	W20	PERR	PERR
N19	AD24	AD24	U18	TRDY	TRDY	Y1	ZV_UV5	ZV_UV5
N20	AD25	AD25	U19	DEVSEL	DEVSEL	Y2	ZV_UV7	ZV_UV7
P1	A_CAD30	A_DATA9	U20	C/BE2	C/BE2	Y3	ZV_PCLK	ZV_PCLK
P2	A_RSVD	A_DATA2	V1	ZV_Y7	ZV_Y7	Y4	MFUNC6	MFUNC6
P3	ZV_HREF	ZV_HREF	V2	ZV_UV1	ZV_UV1	Y5	PHY_LREQ	PHY_LREQ
P4	ZV_Y1	ZV_Y1	V3	ZV_UV3	ZV_UV3	Y6	LINKON	LINKON
P17	AD20	AD20	V4	ZV_LRCLK	ZV_LRCLK	Y7	PHY_DATA2	PHY_DATA2
P18	AD23	AD23	V5	MFUNC5	MFUNC5	Y8	PHY_DATA5	PHY_DATA5
P19	V_{CCP}	V_{CCP}	V6	PHY_CLK	PHY_CLK	Y9	SDA	SDA
P20	IDSEL/MFUNC7	IDSEL/MFUNC7	V7	PHY_DATA0	PHY_DATA0	Y10	MFUNC2	MFUNC2
R1	A_CAD31	A_DATA10	V8	PHY_DATA3	PHY_DATA3	Y11	MFUNC1	MFUNC1
R2	ZV_VSYNC	ZV_VSYNC	V9	PHY_DATA7	PHY_DATA7	Y12	G_RST	G_RST
R3	ZV_Y2	ZV_Y2	V10	MFUNC3	MFUNC3	Y13	RI_OUT	RI_OUT
R4	V_{CC}	V_{CC}	V11	SPKROUT	SPKROUT	Y14	AD1	AD1
R17	V_{CC}	V_{CC}	V12	DATA	DATA	Y15	AD3	AD3
R18	AD19	AD19	V13	AD0	AD0	Y16	AD5	AD5
R19	AD21	AD21	V14	V_{CCP}	V_{CCP}	Y17	AD8	AD8
R20	AD22	AD22	V15	AD7	AD7	Y18	AD11	AD11
T1	ZV_Y0	ZV_Y0	V16	AD9	AD9	Y19	AD15	AD15
T2	ZV_Y3	ZV_Y3	V17	AD13	AD13	Y20	SERR	SERR
T3	ZV_Y5	ZV_Y5	V18	C/BE1	C/BE1			
T4	ZV_UV0	ZV_UV0	V19	STOP	STOP			

Table 2–2. CardBus PC Card Signal Names Sorted Alphanumerically to GFN Terminal Number

SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.
A_CAD0	B8	A_CFRAME	G4	AD26	M17	B_CC/BE3	C13
A_CAD1	A7	A_CGNT	C1	AD27	M18	B_CCD1	J20
A_CAD2	C8	A_CINT	L2	AD28	M19	B_CCD2	B10
A_CAD3	A6	A_CIRDY	E1	AD29	M20	B_CCLK	B17
A_CAD4	B7	A_CPAR	C2	AD30	L19	B_CCLKRUN	A10
A_CAD5	B6	A_CPERR	D3	AD31	L18	B_CDEVSEL	B18
A_CAD6	C7	A_CREQ	J3	B_CAD0	J19	B_CFRAME	A18
A_CAD7	D7	A_CRST	H2	B_CAD1	J17	B_CGNT	A19
A_CAD8	C6	A_CSERR	L3	B_CAD2	J18	B_CINT	A12
A_CAD9	C5	A_CSTOP	E4	B_CAD3	H19	B_CIRDY	D16
A_CAD10	B4	A_CSTSCHG	M2	B_CAD4	H20	B_CPAR	B20
A_CAD11	A3	A_CTRDY	E2	B_CAD5	G20	B_CPERR	B19
A_CAD12	C4	A_CVS1	L1	B_CAD6	H18	B_CREQ	B14
A_CAD13	D5	A_CVS2	G1	B_CAD7	F20	B_CRST	B15
A_CAD14	B2	A_RSVD	A5	B_CAD8	G18	B_CSERR	B11
A_CAD15	B3	A_RSVD	B1	B_CAD9	E20	B_CSTOP	A20
A_CAD16	A2	A_RSVD	P2	B_CAD10	G17	B_CSTSCHG	A11
A_CAD17	F1	AD0	V13	B_CAD11	F18	B_CTRDY	C17
A_CAD18	G3	AD1	Y14	B_CAD12	E18	B_CVS1	B12
A_CAD19	G2	AD2	W14	B_CAD13	D20	B_CVS2	C15
A_CAD20	H3	AD3	Y15	B_CAD14	C20	B_RSVD	C9
A_CAD21	H1	AD4	W15	B_CAD15	D19	B_RSVD	C19
A_CAD22	J4	AD5	Y16	B_CAD16	E17	B_RSVD	G19
A_CAD23	J2	AD6	U14	B_CAD17	C16	C/BE0	W16
A_CAD24	K2	AD7	V15	B_CAD18	B16	C/BE1	V18
A_CAD25	K3	AD8	Y17	B_CAD19	A16	C/BE2	U20
A_CAD26	K1	AD9	V16	B_CAD20	D14	C/BE3	N18
A_CAD27	N1	AD10	W17	B_CAD21	A15	CLKRUN	K18
A_CAD28	N2	AD11	Y18	B_CAD22	C14	CLOCK	U12
A_CAD29	N3	AD12	U16	B_CAD23	A14	DATA	V12
A_CAD30	P1	AD13	V17	B_CAD24	A13	DEVSEL	U19
A_CAD31	R1	AD14	W18	B_CAD25	D12	FRAME	V20
A_CAUDIO	M1	AD15	Y19	B_CAD26	C12	GND	A1
A_CBLOCK	D2	AD16	T18	B_CAD27	C10	GND	D4
A_CC/BE0	A4	AD17	T19	B_CAD28	D10	GND	D8
A_CC/BE1	C3	AD18	T20	B_CAD29	A9	GND	D13
A_CC/BE2	F2	AD19	R18	B_CAD30	B9	GND	D17
A_CC/BE3	J1	AD20	P17	B_CAD31	D9	GND	H4
A_CCD1	A8	AD21	R19	B_CAUDIO	C11	GND	H17
A_CCD2	M4	AD22	R20	B_CBLOCK	C18	GND	N4
A_CCLK	E3	AD23	P18	B_CC/BE0	F19	GND	N17
A_CCLKRUN	M3	AD24	N19	B_CC/BE1	D18	GND	U4
A_CDEVSEL	D1	AD25	N20	B_CC/BE2	A17	GND	U8

Table 2–2. CardBus PC Card Signal Names Sorted Alphanumerically to GFN Terminal Number (continued)

SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.
GND	U13	PHY_CTL0	U7	V_{CC}	D6	ZV_MCLK	W4
GND	U17	PHY_CTL1	W6	V_{CC}	D11	ZV_PCLK	Y3
$\overline{GNT}$	K20	PHY_DATA0	V7	V_{CC}	D15	ZV_SCLK	W3
$\overline{GRST}$	Y12	PHY_DATA1	W7	V_{CC}	F4	ZV_SDATA	U5
IDSEL/MFUNC7	P20	PHY_DATA2	Y7	V_{CC}	F17	ZV_UV0	T4
$\overline{IRDY}$	T17	PHY_DATA3	V8	V_{CC}	K4	ZV_UV1	V2
IRQSER	W13	PHY_DATA4	W8	V_{CC}	L17	ZV_UV2	U3
LATCH	W12	PHY_DATA5	Y8	V_{CC}	R4	ZV_UV3	V3
LINKON	Y6	PHY_DATA6	U9	V_{CC}	R17	ZV_UV4	W1
LPS	W5	PHY_DATA7	V9	V_{CC}	U6	ZV_UV5	Y1
MFUNC0	W11	PHY_LREQ	Y5	V_{CC}	U10	ZV_UV6	W2
MFUNC1	Y11	$\overline{PRST}$	K19	V_{CC}	U15	ZV_UV7	Y2
MFUNC2	Y10	$\overline{REQ}$	L20	V_{CCA}	B5	ZV_VSYNC	R2
MFUNC3	V10	$\overline{RI\_OUT}$	Y13	V_{CCA}	F3	ZV_Y0	T1
MFUNC4	W9	SCL	W10	V_{CCA}	L4	ZV_Y1	P4
MFUNC5	V5	SDA	Y9	V_{CCB}	B13	ZV_Y2	R3
MFUNC6	Y4	$\overline{SERR}$	Y20	V_{CCB}	E19	ZV_Y3	T2
PAR	W19	$\overline{SPKROUT}$	V11	V_{CCP}	P19	ZV_Y4	U1
PCLK	K17	$\overline{STOP}$	V19	V_{CCP}	V14	ZV_Y5	T3
$\overline{PERR}$	W20	$\overline{SUSPEND}$	U11	ZV_HREF	P3	ZV_Y6	U2
PHY_CLK	V6	$\overline{TRDY}$	U18	ZV_LRCLK	V4	ZV_Y7	V1

Table 2–3. 16-Bit PC Card Signal Names Sorted Alphanumerically to GFN Terminal Number

SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.
A_ADDR0	K1	A_DATA11	C8	AD26	M17	B_DATA5	H19
A_ADDR1	K3	A_DATA12	B7	AD27	M18	B_DATA6	G20
A_ADDR2	K2	A_DATA13	C7	AD28	M19	B_DATA7	F20
A_ADDR3	J2	A_DATA14	A5	AD29	M20	B_DATA8	D10
A_ADDR4	J4	A_DATA15	C6	AD30	L19	B_DATA9	B9
A_ADDR5	H1	A_$\overline{\text{INPACK}}$	J3	AD31	L18	B_DATA10	D9
A_ADDR6	H3	A_$\overline{\text{IORD}}$	D5	B_ADDR0	C12	B_DATA11	J18
A_ADDR7	G3	A_$\overline{\text{IOWR}}$	B3	B_ADDR1	D12	B_DATA12	H20
A_ADDR8	C3	A_$\overline{\text{OE}}$	A3	B_ADDR2	A13	B_DATA13	H18
A_ADDR9	B2	A_READY($\overline{\text{IREQ}}$)	L2	B_ADDR3	A14	B_DATA14	G19
A_ADDR10	C5	A_$\overline{\text{REG}}$	J1	B_ADDR4	C14	B_DATA15	G18
A_ADDR11	C4	A_RESET	H2	B_ADDR5	A15	B_$\overline{\text{INPACK}}$	B14
A_ADDR12	F2	A_$\overline{\text{VS1}}$	L1	B_ADDR6	D14	B_$\overline{\text{IORD}}$	D20
A_ADDR13	C2	A_$\overline{\text{VS2}}$	G1	B_ADDR7	B16	B_$\overline{\text{IOWR}}$	D19
A_ADDR14	D3	A_$\overline{\text{WAIT}}$	L3	B_ADDR8	D18	B_$\overline{\text{OE}}$	F18
A_ADDR15	E1	A_$\overline{\text{WE}}$	C1	B_ADDR9	C20	B_READY($\overline{\text{IREQ}}$)	A12
A_ADDR16	E3	A_WP($\overline{\text{IOIS16}}$)	M3	B_ADDR10	E20	B_$\overline{\text{REG}}$	C13
A_ADDR17	A2	AD0	V13	B_ADDR11	E18	B_RESET	B15
A_ADDR18	B1	AD1	Y14	B_ADDR12	A17	B_$\overline{\text{VS1}}$	B12
A_ADDR19	D2	AD2	W14	B_ADDR13	B20	B_$\overline{\text{VS2}}$	C15
A_ADDR20	E4	AD3	Y15	B_ADDR14	B19	B_$\overline{\text{WAIT}}$	B11
A_ADDR21	D1	AD4	W15	B_ADDR15	D16	B_$\overline{\text{WE}}$	A19
A_ADDR22	E2	AD5	Y16	B_ADDR16	B17	B_WP($\overline{\text{IOIS16}}$)	A10
A_ADDR23	G4	AD6	U14	B_ADDR17	E17	C/$\overline{\text{BE0}}$	W16
A_ADDR24	F1	AD7	V15	B_ADDR18	C19	C/$\overline{\text{BE1}}$	V18
A_ADDR25	G2	AD8	Y17	B_ADDR19	C18	C/$\overline{\text{BE2}}$	U20
A_BVD1($\overline{\text{STSCHG}}$/$\overline{\text{RI}}$)	M2	AD9	V16	B_ADDR20	A20	C/$\overline{\text{BE3}}$	N18
A_BVD2($\overline{\text{SPKR}}$)	M1	AD10	W17	B_ADDR21	B18	$\overline{\text{CLKRUN}}$	K18
A_$\overline{\text{CD1}}$	A8	AD11	Y18	B_ADDR22	C17	CLOCK	U12
A_$\overline{\text{CD2}}$	M4	AD12	U16	B_ADDR23	A18	DATA	V12
A_$\overline{\text{CE1}}$	A4	AD13	V17	B_ADDR24	C16	$\overline{\text{DEVSEL}}$	U19
A_$\overline{\text{CE2}}$	B4	AD14	W18	B_ADDR25	A16	$\overline{\text{FRAME}}$	V20
A_DATA0	N1	AD15	Y19	B_BVD1($\overline{\text{STSCHG}}$/$\overline{\text{RI}}$)	A11	GND	A1
A_DATA1	N3	AD16	T18	B_BVD2($\overline{\text{SPKR}}$)	C11	GND	D4
A_DATA2	P2	AD17	T19	B_$\overline{\text{CD1}}$	J20	GND	D8
A_DATA3	B8	AD18	T20	B_$\overline{\text{CD2}}$	B10	GND	D13
A_DATA4	A7	AD19	R18	B_$\overline{\text{CE1}}$	F19	GND	D17
A_DATA5	A6	AD20	P17	B_$\overline{\text{CE2}}$	G17	GND	H4
A_DATA6	B6	AD21	R19	B_DATA0	C10	GND	H17
A_DATA7	D7	AD22	R20	B_DATA1	A9	GND	N4
A_DATA8	N2	AD23	P18	B_DATA2	C9	GND	N17
A_DATA9	P1	AD24	N19	B_DATA3	J19	GND	U4
A_DATA10	R1	AD25	N20	B_DATA4	J17	GND	U8

Table 2–3. 16-Bit PC Card Signal Names Sorted Alphanumerically to GFN Terminal Number (continued)

SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.
GND	U13	PHY_CTL0	U7	V_{CC}	D6	ZV_MCLK	W4
GND	U17	PHY_CTL1	W6	V_{CC}	D11	ZV_PCLK	Y3
$\overline{\text{GNT}}$	K20	PHY_DATA0	V7	V_{CC}	D15	ZV_SCLK	W3
$\overline{\text{GRST}}$	Y12	PHY_DATA1	W7	V_{CC}	F4	ZV_SDATA	U5
IDSEL/MFUNC7	P20	PHY_DATA2	Y7	V_{CC}	F17	ZV_UV0	T4
$\overline{\text{IRDY}}$	T17	PHY_DATA3	V8	V_{CC}	K4	ZV_UV1	V2
IRQSER	W13	PHY_DATA4	W8	V_{CC}	L17	ZV_UV2	U3
LATCH	W12	PHY_DATA5	Y8	V_{CC}	R4	ZV_UV3	V3
LINKON	Y6	PHY_DATA6	U9	V_{CC}	R17	ZV_UV4	W1
LPS	W5	PHY_DATA7	V9	V_{CC}	U6	ZV_UV5	Y1
MFUNC0	W11	PHY_LREQ	Y5	V_{CC}	U10	ZV_UV6	W2
MFUNC1	Y11	$\overline{\text{PRST}}$	K19	V_{CC}	U15	ZV_UV7	Y2
MFUNC2	Y10	$\overline{\text{REQ}}$	L20	V_{CCA}	B5	ZV_VSYNC	R2
MFUNC3	V10	$\overline{\text{RI\_OUT}}$	Y13	V_{CCA}	F3	ZV_Y0	T1
MFUNC4	W9	SCL	W10	V_{CCA}	L4	ZV_Y1	P4
MFUNC5	V5	SDA	Y9	V_{CCB}	B13	ZV_Y2	R3
MFUNC6	Y4	$\overline{\text{SERR}}$	Y20	V_{CCB}	E19	ZV_Y3	T2
PAR	W19	$\overline{\text{SPKROUT}}$	V11	V_{CCP}	P19	ZV_Y4	U1
PCLK	K17	$\overline{\text{STOP}}$	V19	V_{CCP}	V14	ZV_Y5	T3
$\overline{\text{PERR}}$	W20	$\overline{\text{SUSPEND}}$	U11	ZV_HREF	P3	ZV_Y6	U2
PHY_CLK	V6	$\overline{\text{TRDY}}$	U18	ZV_LRCLK	V4	ZV_Y7	V1

Table 2–4. GJG Terminals Sorted Alphanumerically for CardBus // 16-Bit Signals and OHCI

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CARDBUS	16-BIT		CARDBUS	16-BIT		CARDBUS	16-BIT
A2	A_CC/$\overline{BE1}$	A_ADDR8	D5	A_CAD13	A_$\overline{IORD}$	F14	B_CAD15	B_$\overline{IOWR}$
A3	GND	GND	D6	A_CC/$\overline{BE0}$	A_$\overline{CE1}$	F15	B_CAD12	B_ADDR11
A4	A_CAD12	A_ADDR11	D7	A_CAD5	A_DATA6	F16	B_CAD13	B_$\overline{IORD}$
A5	A_CAD10	A_$\overline{CE2}$	D8	GND	GND	F18	V_{CCB}	V_{CCB}
A6	A_CAD8	A_DATA15	D9	B_RSVD	B_DATA2	F19	B_CAD11	B_$\overline{OE}$
A7	A_CAD3	A_DATA5	D10	B_$\overline{CCD2}$	B_$\overline{CD2}$	G1	GND	GND
A8	A_CAD0	A_DATA3	D11	B_CAD26	B_ADDR0	G2	A_CAD18	A_ADDR7
A9	B_CAD29	B_DATA1	D12	B_CAD24	B_ADDR2	G4	A_CAD19	A_ADDR25
A10	B_CSTSCHG	B_BVD1($\overline{STSCHG}$/$\overline{RI}$)	D13	B_CAD23	B_ADDR3	G5	A_CAD17	A_ADDR24
A11	V_{CC}	V_{CC}	D14	V_{CC}	V_{CC}	G6	A_CC/$\overline{BE2}$	A_ADDR12
A12	B_CC/$\overline{BE3}$	B_$\overline{REG}$	D15	B_$\overline{CFRAME}$	B_ADDR23	G7	A_CAD4	A_DATA12
A13	B_$\overline{CREQ}$	B_$\overline{INPACK}$	D16	B_$\overline{CBLOCK}$	B_ADDR19	G13	B_CAD7	B_DATA7
A14	B_CVS2	B_$\overline{VS2}$	D18	B_RSVD	B_ADDR18	G14	B_CAD10	B_$\overline{CE2}$
A15	B_CAD17	B_ADDR24	D19	B_CC/$\overline{BE1}$	B_ADDR8	G15	B_CAD9	B_ADDR10
A16	GND	GND	E1	V_{CC}	V_{CC}	G16	B_CC/$\overline{BE0}$	B_$\overline{CE1}$
A17	B_CCLK	B_ADDR16	E2	A_CCLK	A_ADDR16	G18	B_CAD8	B_DATA15
A18	B_$\overline{CDEVSEL}$	B_ADDR21	E4	A_$\overline{CGNT}$	A_$\overline{WE}$	G19	GND	GND
B1	A_CPAR	A_ADDR13	E5	A_$\overline{CDEVSEL}$	A_ADDR21	H1	A_CAD20	A_ADDR6
B2	A_RSVD	A_ADDR18	E6	V_{CC}	V_{CC}	H2	A_$\overline{CRST}$	A_RESET
B3	A_CAD16	A_ADDR17	E7	A_RSVD	A_DATA14	H4	A_CAD21	A_ADDR5
B4	A_CAD15	A_$\overline{IOWR}$	E8	A_CAD1	A_DATA4	H5	A_CAD22	A_ADDR4
B5	A_CAD11	A_$\overline{OE}$	E9	B_CAD31	B_DATA10	H6	A_CVS2	A_$\overline{VS2}$
B6	V_{CCA}	V_{CCA}	E10	B_CAD27	B_DATA0	H14	B_CAD4	B_DATA12
B7	A_CAD6	A_DATA13	E11	B_$\overline{CINT}$	B_READY($\overline{IREQ}$)	H15	B_RSVD	B_DATA14
B8	A_CAD2	A_DATA11	E12	B_CAD25	B_ADDR1	H16	B_CAD5	B_DATA6
B9	B_CAD30	B_DATA9	E13	B_CAD21	B_ADDR5	H18	B_CAD6	B_DATA13
B10	B_$\overline{CCLKRUN}$	B_WP($\overline{IOIS16}$)	E14	B_CAD19	B_ADDR25	H19	B_CAD3	B_DATA5
B11	B_CVS1	B_$\overline{VS1}$	E15	B_CC/$\overline{BE2}$	B_ADDR12	J1	A_CAD23	A_ADDR3
B12	V_{CCB}	V_{CCB}	E16	B_CAD16	B_ADDR17	J2	A_CC/$\overline{BE3}$	A_$\overline{REG}$
B13	B_CAD22	B_ADDR4	E18	B_CAD14	B_ADDR9	J4	A_$\overline{CREQ}$	A_$\overline{INPACK}$
B14	B_CAD20	B_ADDR6	E19	V_{CC}	V_{CC}	J5	A_CAD24	A_ADDR2
B15	B_CAD18	B_ADDR7	F1	V_{CCA}	V_{CCA}	J6	A_CAD25	A_ADDR1
B16	B_$\overline{CIRDY}$	B_ADDR15	F2	A_$\overline{CFRAME}$	A_ADDR23	J14	V_{CC}	V_{CC}
B17	B_$\overline{CTRDY}$	B_ADDR22	F4	A_$\overline{CIRDY}$	A_ADDR15	J15	B_CAD1	B_DATA4
B18	B_$\overline{CGNT}$	B_$\overline{WE}$	F5	A_$\overline{CTRDY}$	A_ADDR22	J16	B_CAD2	B_DATA11
B19	B_$\overline{CSTOP}$	B_ADDR20	F6	A_CAD9	A_ADDR10	J18	B_CAD0	B_DATA3
C1	GND	GND	F7	A_CAD7	A_DATA7	J19	B_$\overline{CCD1}$	B_$\overline{CD1}$
C2	A_$\overline{CBLOCK}$	A_ADDR19	F8	A_$\overline{CCD1}$	A_$\overline{CD1}$	K1	A_CVS1	A_$\overline{VS1}$
C18	B_$\overline{CPERR}$	B_ADDR14	F9	B_CAD28	B_DATA8	K2	A_$\overline{CINT}$	A_READY($\overline{IREQ}$)
C19	B_CPAR	B_ADDR13	F10	B_CAUDIO	B_BVD2($\overline{SPKR}$)	K4	A_$\overline{CSERR}$	A_$\overline{WAIT}$
D1	A_$\overline{CPERR}$	A_ADDR14	F11	B_$\overline{CSERR}$	B_$\overline{WAIT}$	K5	V_{CCA}	V_{CCA}
D2	A_$\overline{CSTOP}$	A_ADDR20	F12	GND	GND	K6	A_CAD26	A_ADDR0
D4	A_CAD14	A_ADDR9	F13	B_$\overline{CRST}$	B_RESET	K14	$\overline{GNT}$	$\overline{GNT}$

Table 2–4. GJG Terminals Sorted Alphanumerically for CardBus // 16-Bit Signals and OHCI (continued)

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CARDBUS	16-BIT		CARDBUS	16-BIT		CARDBUS	16-BIT
K15	PCLK	PCLK	P9	MFUNC2	MFUNC2	T18	FRAME	FRAME
K18	CLKRUN	CLKRUN	P10	MFUNC1	MFUNC1	T19	IRDY	IRDY
K19	PRST	PRST	P11	GRST	GRST	U1	ZV_UV3	ZV_UV3
L1	A_CSTSCHG	A_BVD1(STSCHG/RI)	P12	IRQSER	IRQSER	U2	ZV_UV6	ZV_UV6
L2	A_CCLKRUN	A_WP(IOIS16)	P13	AD6	AD6	U18	TRDY	TRDY
L4	A_CCD2	A_CD2	P14	AD9	AD9	U19	DEVSEL	DEVSEL
L5	A_CAD27	A_DATA0	P15	V_{CC}	V_{CC}	V1	ZV_UV5	ZV_UV5
L6	A_CAUDIO	A_BVD2(SPKR)	P16	AD19	AD19	V2	ZV_SCLK	ZV_SCLK
L14	REQ	REQ	P18	AD21	AD21	V3	ZV_LRCLK	ZV_LRCLK
L15	AD31	AD31	P19	AD20	AD20	V4	ZV_PCLK	ZV_PCLK
L16	AD28	AD28	R1	ZV_Y7	ZV_Y7	V5	LPS	LPS
L18	AD30	AD30	R2	ZV_UV0	ZV_UV0	V6	PHY_CTL1	PHY_CTL1
L19	AD29	AD29	R4	ZV_UV2	ZV_UV2	V7	PHY_DATA1	PHY_DATA1
M1	A_CAD29	A_DATA1	R5	MFUNC6	MFUNC6	V8	PHY_DATA5	PHY_DATA5
M2	GND	GND	R6	PHY_LREQ	PHY_LREQ	V9	SCL	SCL
M4	A_CAD30	A_DATA9	R7	PHY_DATA0	PHY_DATA0	V10	V_{CC}	V_{CC}
M5	A_RSVD	A_DATA2	R8	PHY_DATA7	PHY_DATA7	V11	DATA	DATA
M6	A_CAD28	A_DATA8	R9	MFUNC3	MFUNC3	V12	AD0	AD0
M14	C/BE3	C/BE3	R10	SUSPEND	SUSPEND	V13	V_{CC}	V_{CC}
M15	AD27	AD27	R11	RI_OUT	RI_OUT	V14	GND	GND
M16	AD26	AD26	R12	AD2	AD2	V15	AD11	AD11
M18	AD25	AD25	R13	AD5	AD5	V16	AD14	AD14
M19	AD24	AD24	R14	AD8	AD8	V17	PAR	PAR
N1	ZV_HREF	ZV_HREF	R15	AD16	AD16	V18	PERR	PERR
N2	ZV_VSYNC	ZV_VSYNC	R16	C/BE2	C/BE2	V19	STOP	STOP
N4	ZV_Y0	ZV_Y0	R18	AD18	AD18	W2	ZV_UV7	ZV_UV7
N5	ZV_Y1	ZV_Y1	R19	AD17	AD17	W3	ZV_MCLK	ZV_MCLK
N6	ZV_Y2	ZV_Y2	T1	ZV_UV1	ZV_UV1	W4	ZV_SDATA	ZV_SDATA
N7	A_CAD31	A_DATA10	T2	ZV_UV4	ZV_UV4	W5	MFUNC5	MFUNC5
N13	AD3	AD3	T4	GND	GND	W6	PHY_CTL0	PHY_CTL0
N14	AD22	AD22	T5	V_{CC}	V_{CC}	W7	PHY_DATA2	PHY_DATA2
N15	AD23	AD23	T6	PHY_CLK	PHY_CLK	W8	PHY_DATA4	PHY_DATA4
N16	GND	GND	T7	GND	GND	W9	SDA	SDA
N18	V_{CCP}	V_{CCP}	T8	PHY_DATA6	PHY_DATA6	W10	MFUNC0	MFUNC0
N19	IDSEL/MFUNC7	IDSEL/MFUNC7	T9	MFUNC4	MFUNC4	W11	LATCH	LATCH
P1	V_{CC}	V_{CC}	T10	SPKROUT	SPKROUT	W12	GND	GND
P2	ZV_Y3	ZV_Y3	T11	CLOCK	CLOCK	W13	V_{CCP}	V_{CCP}
P4	ZV_Y4	ZV_Y4	T12	AD1	AD1	W14	AD7	AD7
P5	ZV_Y5	ZV_Y5	T13	AD4	AD4	W15	AD10	AD10
P6	ZV_Y6	ZV_Y6	T14	C/BE0	C/BE0	W16	AD13	AD13
P7	LINKON	LINKON	T15	AD12	AD12	W17	AD15	AD15
P8	PHY_DATA3	PHY_DATA3	T16	C/BE1	C/BE1	W18	SERR	SERR

Table 2–5. CardBus PC Card Signal Names Sorted Alphanumerically to GJG Terminal Number

SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.
A_CAD0	A8	A_CFRAME	F2	AD26	M16	B_CC/BE3	A12
A_CAD1	E8	A_CGNT	E4	AD27	M15	B_CCD1	J19
A_CAD2	B8	A_CINT	K2	AD28	L16	B_CCD2	D10
A_CAD3	A7	A_CIRDY	F4	AD29	L19	B_CCLK	A17
A_CAD4	G7	A_CPAR	B1	AD30	L18	B_CCLKRUN	B10
A_CAD5	D7	A_CPERR	D1	AD31	L15	B_CDEVSEL	A18
A_CAD6	B7	A_CREQ	J4	B_CAD0	J18	B_CFRAME	D15
A_CAD7	F7	A_CRST	H2	B_CAD1	J15	B_CGNT	B18
A_CAD8	A6	A_CSERR	K4	B_CAD2	J16	B_CINT	E11
A_CAD9	F6	A_CSTOP	D2	B_CAD3	H19	B_CIRDY	B16
A_CAD10	A5	A_CSTSCHG	L1	B_CAD4	H14	B_CPAR	C19
A_CAD11	B5	A_CTRDY	F5	B_CAD5	H16	B_CPERR	C18
A_CAD12	A4	A_CVS1	K1	B_CAD6	H18	B_CREQ	A13
A_CAD13	D5	A_CVS2	H6	B_CAD7	G13	B_CRST	F13
A_CAD14	D4	A_RSVD	B2	B_CAD8	G18	B_CSERR	F11
A_CAD15	B4	A_RSVD	E7	B_CAD9	G15	B_CSTOP	B19
A_CAD16	B3	A_RSVD	M5	B_CAD10	G14	B_CSTSCHG	A10
A_CAD17	G5	AD0	V12	B_CAD11	F19	B_CTRDY	B17
A_CAD18	G2	AD1	T12	B_CAD12	F15	B_CVS1	B11
A_CAD19	G4	AD2	R12	B_CAD13	F16	B_CVS2	A14
A_CAD20	H1	AD3	N13	B_CAD14	E18	B_RSVD	D9
A_CAD21	H4	AD4	T13	B_CAD15	F14	B_RSVD	D18
A_CAD22	H5	AD5	R13	B_CAD16	E16	B_RSVD	H15
A_CAD23	J1	AD6	P13	B_CAD17	A15	C/BE0	T14
A_CAD24	J5	AD7	W14	B_CAD18	B15	C/BE1	T16
A_CAD25	J6	AD8	R14	B_CAD19	E14	C/BE2	R16
A_CAD26	K6	AD9	P14	B_CAD20	B14	C/BE3	M14
A_CAD27	L5	AD10	W15	B_CAD21	E13	CLKRUN	K18
A_CAD28	M6	AD11	V15	B_CAD22	B13	CLOCK	T11
A_CAD29	M1	AD12	T15	B_CAD23	D13	DATA	V11
A_CAD30	M4	AD13	W16	B_CAD24	D12	DEVSEL	U19
A_CAD31	N7	AD14	V16	B_CAD25	E12	FRAME	T18
A_CAUDIO	L6	AD15	W17	B_CAD26	D11	GND	A3
A_CBLOCK	C2	AD16	R15	B_CAD27	E10	GND	A16
A_CC/BE0	D6	AD17	R19	B_CAD28	F9	GND	C1
A_CC/BE1	A2	AD18	R18	B_CAD29	A9	GND	D8
A_CC/BE2	G6	AD19	P16	B_CAD30	B9	GND	F12
A_CC/BE3	J2	AD20	P19	B_CAD31	E9	GND	G1
A_CCD1	F8	AD21	P18	B_CAUDIO	F10	GND	G19
A_CCD2	L4	AD22	N14	B_CBLOCK	D16	GND	M2
A_CCLK	E2	AD23	N15	B_CC/BE0	G16	GND	N16
A_CCLKRUN	L2	AD24	M19	B_CC/BE1	D19	GND	T4
A_CDEVSEL	E5	AD25	M18	B_CC/BE2	E15	GND	T7

Table 2–5. CardBus PC Card Signal Names Sorted Alphanumerically to GJG Terminal Number (continued)

SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.
GND	V14	PHY_CTL0	W6	V_{CC}	A11	ZV_PCLK	V4
GND	W12	PHY_CTL1	V6	V_{CC}	D14	ZV_SCLK	V2
$\overline{GNT}$	K14	PHY_DATA0	R7	V_{CC}	E1	ZV_SDATA	W4
$\overline{GRST}$	P11	PHY_DATA1	V7	V_{CC}	E6	ZV_UV0	R2
IDSEL/MFUNC7	N19	PHY_DATA2	W7	V_{CC}	E19	ZV_UV1	T1
$\overline{IRDY}$	T19	PHY_DATA3	P8	V_{CC}	J14	ZV_UV2	R4
IRQSER	P12	PHY_DATA4	W8	V_{CC}	P1	ZV_UV3	U1
LATCH	W11	PHY_DATA5	V8	V_{CC}	P15	ZV_UV4	T2
LINKON	P7	PHY_DATA6	T8	V_{CC}	T5	ZV_UV5	V1
LPS	V5	PHY_DATA7	R8	V_{CC}	V10	ZV_UV6	U2
MFUNC0	W10	PHY_LREQ	R6	V_{CC}	V13	ZV_UV7	W2
MFUNC1	P10	$\overline{PRST}$	K19	V_{CCA}	B6	ZV_VSYNC	N2
MFUNC2	P9	$\overline{REQ}$	L14	V_{CCA}	F1	ZV_Y0	N4
MFUNC3	R9	$\overline{RI\_OUT}$	R11	V_{CCA}	K5	ZV_Y1	N5
MFUNC4	T9	SCL	V9	V_{CCB}	B12	ZV_Y2	N6
MFUNC5	W5	SDA	W9	V_{CCB}	F18	ZV_Y3	P2
MFUNC6	R5	$\overline{SERR}$	W18	V_{CCP}	N18	ZV_Y4	P4
PAR	V17	$\overline{SPKROUT}$	T10	V_{CCP}	W13	ZV_Y5	P5
PCLK	K15	$\overline{STOP}$	V19	ZV_HREF	N1	ZV_Y6	P6
$\overline{PERR}$	V18	$\overline{SUSPEND}$	R10	ZV_LRCLK	V3	ZV_Y7	R1
PHY_CLK	T6	$\overline{TRDY}$	U18	ZV_MCLK	W3		

Table 2–6. 16-Bit PC Card Signal Names Sorted Alphanumerically to GJG Terminal Number

SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM NO.	SIGNAL NAME	TERM. NO.
A_ADDR0	K6	A_DATA11	B8	AD26	M16	B_DATA5	H19
A_ADDR1	J6	A_DATA12	G7	AD27	M15	B_DATA6	H16
A_ADDR2	J5	A_DATA13	B7	AD28	L16	B_DATA7	G13
A_ADDR3	J1	A_DATA14	E7	AD29	L19	B_DATA8	F9
A_ADDR4	H5	A_DATA15	A6	AD30	L18	B_DATA9	B9
A_ADDR5	H4	A_$\overline{\text{INPACK}}$	J4	AD31	L15	B_DATA10	E9
A_ADDR6	H1	A_$\overline{\text{IORD}}$	D5	B_ADDR0	D11	B_DATA11	J16
A_ADDR7	G2	A_$\overline{\text{IOWR}}$	B4	B_ADDR1	E12	B_DATA12	H14
A_ADDR8	A2	A_$\overline{\text{OE}}$	B5	B_ADDR2	D12	B_DATA13	H18
A_ADDR9	D4	A_READY($\overline{\text{IREQ}}$)	K2	B_ADDR3	D13	B_DATA14	H15
A_ADDR10	F6	A_$\overline{\text{REG}}$	J2	B_ADDR4	B13	B_DATA15	G18
A_ADDR11	A4	A_RESET	H2	B_ADDR5	E13	B_$\overline{\text{INPACK}}$	A13
A_ADDR12	G6	A_$\overline{\text{VS1}}$	K1	B_ADDR6	B14	B_$\overline{\text{IORD}}$	F16
A_ADDR13	B1	A_$\overline{\text{VS2}}$	H6	B_ADDR7	B15	B_$\overline{\text{IOWR}}$	F14
A_ADDR14	D1	A_$\overline{\text{WAIT}}$	K4	B_ADDR8	D19	B_$\overline{\text{OE}}$	F19
A_ADDR15	F4	A_$\overline{\text{WE}}$	E4	B_ADDR9	E18	B_READY($\overline{\text{IREQ}}$)	E11
A_ADDR16	E2	A_WP($\overline{\text{IOIS16}}$)	L2	B_ADDR10	G15	B_$\overline{\text{REG}}$	A12
A_ADDR17	B3	AD0	V12	B_ADDR11	F15	B_RESET	F13
A_ADDR18	B2	AD1	T12	B_ADDR12	E15	B_$\overline{\text{VS1}}$	B11
A_ADDR19	C2	AD2	R12	B_ADDR13	C19	B_$\overline{\text{VS2}}$	A14
A_ADDR20	D2	AD3	N13	B_ADDR14	C18	B_$\overline{\text{WAIT}}$	F11
A_ADDR21	E5	AD4	T13	B_ADDR15	B16	B_$\overline{\text{WE}}$	B18
A_ADDR22	F5	AD5	R13	B_ADDR16	A17	B_WP($\overline{\text{IOIS16}}$)	B10
A_ADDR23	F2	AD6	P13	B_ADDR17	E16	C/$\overline{\text{BE0}}$	T14
A_ADDR24	G5	AD7	W14	B_ADDR18	D18	C/$\overline{\text{BE1}}$	T16
A_ADDR25	G4	AD8	R14	B_ADDR19	D16	C/$\overline{\text{BE2}}$	R16
A_BVD1($\overline{\text{STSCHG}}$/$\overline{\text{RI}}$)	L1	AD9	P14	B_ADDR20	B19	C/$\overline{\text{BE3}}$	M14
A_BVD2($\overline{\text{SPKR}}$)	L6	AD10	W15	B_ADDR21	A18	$\overline{\text{CLKRUN}}$	K18
A_$\overline{\text{CD1}}$	F8	AD11	V15	B_ADDR22	B17	CLOCK	T11
A_$\overline{\text{CD2}}$	L4	AD12	T15	B_ADDR23	D15	DATA	V11
A_$\overline{\text{CE1}}$	D6	AD13	W16	B_ADDR24	A15	$\overline{\text{DEVSEL}}$	U19
A_$\overline{\text{CE2}}$	A5	AD14	V16	B_ADDR25	E14	$\overline{\text{FRAME}}$	T18
A_DATA0	L5	AD15	W17	B_BVD1($\overline{\text{STSCHG}}$/$\overline{\text{RI}}$)	A10	GND	A3
A_DATA1	M1	AD16	R15	B_BVD2($\overline{\text{SPKR}}$)	F10	GND	A16
A_DATA2	M5	AD17	R19	B_$\overline{\text{CD1}}$	J19	GND	C1
A_DATA3	A8	AD18	R18	B_$\overline{\text{CD2}}$	D10	GND	D8
A_DATA4	E8	AD19	P16	B_$\overline{\text{CE1}}$	G16	GND	F12
A_DATA5	A7	AD20	P19	B_$\overline{\text{CE2}}$	G14	GND	G1
A_DATA6	D7	AD21	P18	B_DATA0	E10	GND	G19
A_DATA7	F7	AD22	N14	B_DATA1	A9	GND	M2
A_DATA8	M6	AD23	N15	B_DATA2	D9	GND	N16
A_DATA9	M4	AD24	M19	B_DATA3	J18	GND	T4
A_DATA10	N7	AD25	M18	B_DATA4	J15	GND	T7

Table 2–6. 16-Bit PC Card Signal Names Sorted Alphanumerically to GJG Terminal Number (continued)

SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.
GND	V14	PHY_CTL0	W6	V_{CC}	A11	ZV_PCLK	V4
GND	W12	PHY_CTL1	V6	V_{CC}	D14	ZV_SCLK	V2
$\overline{GNT}$	K14	PHY_DATA0	R7	V_{CC}	E1	ZV_SDATA	W4
$\overline{GRST}$	P11	PHY_DATA1	V7	V_{CC}	E6	ZV_UV0	R2
IDSEL/MFUNC7	N19	PHY_DATA2	W7	V_{CC}	E19	ZV_UV1	T1
$\overline{IRDY}$	T19	PHY_DATA3	P8	V_{CC}	J14	ZV_UV2	R4
IRQSER	P12	PHY_DATA4	W8	V_{CC}	P1	ZV_UV3	U1
LATCH	W11	PHY_DATA5	V8	V_{CC}	P15	ZV_UV4	T2
LINKON	P7	PHY_DATA6	T8	V_{CC}	T5	ZV_UV5	V1
LPS	V5	PHY_DATA7	R8	V_{CC}	V10	ZV_UV6	U2
MFUNC0	W10	PHY_LREQ	R6	V_{CC}	V13	ZV_UV7	W2
MFUNC1	P10	$\overline{PRST}$	K19	V_{CCA}	B6	ZV_VSYNC	N2
MFUNC2	P9	$\overline{REQ}$	L14	V_{CCA}	F1	ZV_Y0	N4
MFUNC3	R9	$\overline{RI\_OUT}$	R11	V_{CCA}	K5	ZV_Y1	N5
MFUNC4	T9	SCL	V9	V_{CCB}	B12	ZV_Y2	N6
MFUNC5	W5	SDA	W9	V_{CCB}	F18	ZV_Y3	P2
MFUNC6	R5	$\overline{SERR}$	W18	V_{CCP}	N18	ZV_Y4	P4
PAR	V17	$\overline{SPKROUT}$	T10	V_{CCP}	W13	ZV_Y5	P5
PCLK	K15	$\overline{STOP}$	V19	ZV_HREF	N1	ZV_Y6	P6
$\overline{PERR}$	V18	$\overline{SUSPEND}$	R10	ZV_LRCLK	V3	ZV_Y7	R1
PHY_CLK	T6	$\overline{TRDY}$	U18	ZV_MCLK	W3		

The terminals in Table 2–7 through Table 2–21 are grouped in tables by functionality such as PCI system function and power supply function, for quick reference. The terminal numbers are listed for convenient reference, and a full description of the signal function is given.

Table 2–7. Power Supply Terminals

TERMINAL			FUNCTION
NAME	GFN NO.	GJG NO.	
GND	A1, D4, D8, D13, D17, H4, H17, N4, N17, U4, U8, U13, U17	A3, A16, C1, D8, F12, G1, G19, M2, N16, T4, T7, V14, W12	Device ground terminals
V_{CC}	D6, D11, D15, F4, F17, K4, L17, R4, R17, U6, U10, U15	A11, D14, E1, E6, E19, J14, P1, P15, T5, V10, V13	Power supply terminal for core logic (3.3 Vdc)
V_{CCA}	B5, F3, L4	B6, F1, K5	Clamp voltage for PC Card A interface. Indicates Card A signaling environment.
V_{CCB}	B13, E19	B12, F18	Clamp voltage for PC Card B interface. Indicates Card B signaling environment.
V_{CCP}	P19, V14	N18, W13	Clamp voltage for PCI signaling (3.3 Vdc or 5 Vdc)

Table 2–8. PC Card Power Switch Terminals

TERMINAL			I/O	FUNCTION
NAME	GFN NO.	GJG NO.		
CLOCK	U12	T11	I/O	3-line power switch clock. Information on the DATA line is sampled at the rising edge of CLOCK. This terminal defaults as an input which means an external clock source must be used. If the internal ring oscillator is used, then an external CLOCK source is not required. The internal oscillator may be enabled by setting bit 27 (P2CCLK) of the system control register (PCI offset 80h, see Section 4.28) to a 1b. A 43-kΩ pulldown resistor should be tied to this terminal.
DATA	V12	V11	O	3-line power switch data. DATA is used to communicate socket power-control information to the power switch serially.
LATCH	W12	W11	O	3-line power switch latch. LATCH is asserted by the PCI4450 to indicate to the PC Card power switch that the data on the DATA line is valid.

Table 2–9. PCI System Terminals

TERMINAL			I/O	FUNCTION
NAME	GFN NO.	GJG NO.		
$\overline{\text{CLKRUN}}$	K18	K18	I/O	PCI clock run. $\overline{\text{CLKRUN}}$ is used by the central resource to request permission to stop the PCI clock or to slow it down, and the PCI4450 responds accordingly. If $\overline{\text{CLKRUN}}$ is not implemented, then this terminal should be tied low. $\overline{\text{CLKRUN}}$ is enabled by default by bit 1 (KEEPCLK) in the system control register (PCI offset 80h, see Section 4.28).
PCLK	K17	K15	I	PCI bus clock. PCLK provides timing for all transactions on the PCI bus. All PCI signals are sampled at the rising edge of PCLK.
$\overline{\text{PRST}}$	K19	K19	I	PCI bus reset. When the PCI bus reset is asserted, $\overline{\text{PRST}}$ causes the PCI4450 to place all output buffers in a high-impedance state and reset all internal registers. When $\overline{\text{PRST}}$ is asserted, the device is completely nonfunctional. After $\overline{\text{PRST}}$ is deasserted, the PCI4450 is in its default state. When the $\overline{\text{SUSPEND}}$ mode is enabled, the device is protected from the $\overline{\text{PRST}}$ and the internal registers are preserved. All outputs are placed in a high-impedance state, but the contents of the registers are preserved.
$\overline{\text{GRST}}$	Y12	P11	I	Global reset. When the global reset is asserted, the $\overline{\text{GRST}}$ signal causes the PCI4450 to place all output buffers in a high-impedance state and reset all internal registers. When $\overline{\text{GRST}}$ is asserted, the device is completely in its default state. For systems that require wake-up from D3, $\overline{\text{GRST}}$ will normally be asserted only during initial boot. $\overline{\text{PRST}}$ should be asserted following initial boot so that PME context is retained when transitioning from D3 to D0. For systems that do not require wake-up from D3, $\overline{\text{GRST}}$ should be tied to $\overline{\text{PRST}}$.

Table 2–10. PCI Address and Data Terminals

TERMINAL			I/O	FUNCTION
NAME	GFN NO.	GJG NO.		
AD31	L18	L15	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	L19	L18		
AD29	M20	L19		
AD28	M19	L16		
AD27	M18	M15		
AD26	M17	M16		
AD25	N20	M18		
AD24	N19	M19		
AD23	P18	N15		
AD22	R20	N14		
AD21	R19	P18		
AD20	P17	P19		
AD19	R18	P16		
AD18	T20	R18		
AD17	T19	R19		
AD16	T18	R15		
AD15	Y19	W17		
AD14	W18	V16		
AD13	V17	W16		
AD12	U16	T15		
AD11	Y18	V15		
AD10	W17	W15		
AD9	V16	P14		
AD8	Y17	R14		
AD7	V15	W14		
AD6	U14	P13		
AD5	Y16	R13		
AD4	W15	T13		
AD3	Y15	N13		
AD2	W14	R12		
AD1	Y14	T12		
AD0	V13	V12		
C/$\overline{BE3}$	N18	M14	I/O	PCI bus commands and byte enables. These signals are multiplexed on the same PCI terminals. During the address phase of a primary bus PCI cycle, C/$\overline{BE3}$–C/$\overline{BE0}$ define the bus command. During the data phase, this 4-bit bus is used as byte enables. The byte enables determine which byte paths of the full 32-bit data bus carry meaningful data. C/$\overline{BE0}$ applies to byte 0 (AD7–AD0), C/$\overline{BE1}$ applies to byte 1 (AD15–AD8), C/$\overline{BE2}$ applies to byte 2 (AD23–AD16), and C/$\overline{BE3}$ applies to byte 3 (AD31–AD24).
C/$\overline{BE2}$	U20	R16		
C/$\overline{BE1}$	V18	T16		
C/$\overline{BE0}$	W16	T14		
PAR	W19	V17	I/O	PCI bus parity. In all PCI bus read and write cycles, the PCI4450 calculates even parity across the AD31–AD0 and C/$\overline{BE3}$–C/$\overline{BE0}$ buses. As an initiator during PCI cycles, the PCI4450 outputs this parity indicator with a one-PCLK delay. As a target during PCI cycles, the calculated parity is compared to the initiator parity indicator. A compare error results in the assertion of a parity error ($\overline{PERR}$).

Table 2–11. PCI Interface Control Terminals

TERMINAL			I/O	FUNCTION
NAME	GFN NO.	GJG NO.		
$\overline{\text{DEVSEL}}$	U19	U19	I/O	PCI device select. The PCI4450 asserts $\overline{\text{DEVSEL}}$ to claim a PCI cycle as the target device. As a PCI initiator on the bus, the PCI4450 monitors $\overline{\text{DEVSEL}}$ until a target responds. If no target responds before timeout occurs, then the PCI4450 terminates the cycle with an initiator abort.
$\overline{\text{FRAME}}$	V20	T18	I/O	PCI cycle frame. $\overline{\text{FRAME}}$ is driven by the initiator of a bus cycle. $\overline{\text{FRAME}}$ is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When $\overline{\text{FRAME}}$ is deasserted, the PCI bus transaction is in the final data phase.
$\overline{\text{GNT}}$	K20	K14	I	PCI bus grant. $\overline{\text{GNT}}$ is driven by the PCI bus arbiter to grant the PCI4450 access to the PCI bus after the current data transaction has completed. $\overline{\text{GNT}}$ may or may not follow a PCI bus request, depending on the PCI bus parking algorithm.
$\overline{\text{LOCK}}$ (MFUNC7)	P20	N19	I/O	PCI bus lock. MFUNC7/$\overline{\text{LOCK}}$ can be configured as PCI $\overline{\text{LOCK}}$ and used to gain exclusive access downstream. Since this functionality is not typically used, other functions may be accessed through this terminal. MFUNC7/$\overline{\text{LOCK}}$ defaults to and can be configured through the multifunction routing status register (PCI offset 8Ch, see Section 4.36).
IDSEL/MFUNC7	P20	N19	I	Initialization device select. IDSEL selects the PCI4450 during configuration space accesses. IDSEL can be connected to one of the upper 24 PCI address lines on the PCI bus. If the LATCH terminal (W12/W11) has an external pulldown resistor, then this terminal is configurable as MFUNC7 and IDSEL defaults to the AD23 terminal.
$\overline{\text{IRDY}}$	T17	T19	I/O	PCI initiator ready. $\overline{\text{IRDY}}$ indicates the ability of the PCI bus initiator to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCLK where both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. Until $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are both sampled asserted, wait states are inserted.
$\overline{\text{PERR}}$	W20	V18	I/O	PCI parity error indicator. $\overline{\text{PERR}}$ is driven by a PCI device to indicate that calculated parity does not match PAR when $\overline{\text{PERR}}$ is enabled through bit 6 of the command register (PCI offset 04h, see Section 4.4).
$\overline{\text{REQ}}$	L20	L14	O	PCI bus request. $\overline{\text{REQ}}$ is asserted by the PCI4450 to request access to the PCI bus as an initiator.
$\overline{\text{SERR}}$	Y20	W18	O	PCI system error. $\overline{\text{SERR}}$ is an output that is pulsed from the PCI4450 when enabled through bit 8 of the command register (PCI offset 04h, see Section 4.4), indicating a system error has occurred. The PCI4450 need not be the target of the PCI cycle to assert this signal. When $\overline{\text{SERR}}$ is enabled by bit 1 in the bridge control register (PCI offset 3Eh, see Section 4.24), this signal also pulses, indicating that an address parity error has occurred on a CardBus interface.
$\overline{\text{STOP}}$	V19	V19	I/O	PCI cycle stop signal. $\overline{\text{STOP}}$ is driven by a PCI target to request the initiator to stop the current PCI bus transaction. $\overline{\text{STOP}}$ is used for target disconnects and is commonly asserted by target devices that do not support burst data transfers.
$\overline{\text{TRDY}}$	U18	U18	I/O	PCI target ready. $\overline{\text{TRDY}}$ indicates the ability of the primary bus target to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCLK when both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. Until both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted, wait states are inserted.

Table 2–12. System Interrupt Terminals

TERMINAL			I/O	FUNCTION
NAME	GFN NO.	GJG NO.		
$\overline{\text{INTA}}$ (MFUNC0)	W11	W10	I/O	Parallel PCI interrupt. $\overline{\text{INTA}}$ can be mapped to MFUNC0 when parallel PCI interrupts are used. See Section 3.5, *Programmable Interrupt Subsystem*, for details on interrupt signaling. MFUNC0/$\overline{\text{INTA}}$ defaults to a general-purpose input.
$\overline{\text{INTB}}$ (MFUNC1)	Y11	P10	I/O	Parallel PCI interrupt. $\overline{\text{INTB}}$ can be mapped to MFUNC1 when parallel PCI interrupts are used. See Section 3.5, *Programmable Interrupt Subsystem*, for details on interrupt signaling. MFUNC1/$\overline{\text{INTB}}$ defaults to a general-purpose input.
$\overline{\text{INTC}}$ (MFUNC2)	Y10	P9	I/O	Parallel PCI interrupt. $\overline{\text{INTC}}$ can be mapped to MFUNC2 when parallel PCI interrupts are used. See Section 3.5, *Programmable Interrupt Subsystem*, for details on interrupt signaling. MFUNC2/$\overline{\text{INTC}}$ defaults to a general-purpose input.
IRQSER	W13	P12	I/O	Serial interrupt signal. IRQSER provides the IRQSER-style serial interrupting scheme. Serialized PCI interrupts can also be sent in the IRQSER stream. See Section 3.5, *Programmable Interrupt Subsystem*, for details on interrupt signaling.
MFUNC6 MFUNC5 MFUNC4 MFUNC3 MFUNC2 MFUNC1 MFUNC0	Y4 V5 W9 V10 Y10 Y11 W11	R5 W5 T9 R9 P9 P10 W10	O	Interrupt request/secondary functions multiplexed. The primary function of these terminals is to provide programmable options supported by the PCI4450. These interrupt multiplexer outputs can be mapped to various functions. See Section 4.36, *Multifunction Routing Status Register*, for options. All of these terminals have secondary functions, such as PCI interrupts, PC/PCI DMA, OHCI LEDs, GPE request/grant, ring indicate output, and zoomed video status, that can be selected with the appropriate programming of this register. When the secondary functions are enabled, the respective terminals are not available for multifunction routing. See Section 4.36, *Multifunction Routing Status Register,* for programming options.
$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	Y13	R11	O	Ring indicate out and power management event output. Terminal provides an output to the system for ring-indicate or $\overline{\text{PME}}$ signals. Alternatively, $\overline{\text{RI\_OUT}}$ can be routed on MFUNC7.

Table 2–13. PC/PCI DMA Terminals

TERMINAL			I/O	FUNCTION
NAME	GFN NO.	GJG NO.		
$\overline{\text{PCGNT}}$ (MFUNC2)	Y10	P9	I/O	PC/PCI DMA grant. $\overline{\text{PCGNT}}$ is used to grant the DMA channel to a requester in a system supporting the PC/PCI DMA scheme. $\overline{\text{PCGNT}}$ is available on MFUNC2 or MFUNC3. This terminal is also used for the serial EEPROM interface.
$\overline{\text{PCGNT}}$ (MFUNC3)	V10	R9		
$\overline{\text{PCREQ}}$ (MFUNC7)	P20	N19	O	PC/PCI DMA request. $\overline{\text{PCREQ}}$ is used to request DMA transfers as $\overline{\text{DREQ}}$ in a system supporting the PC/PCI DMA scheme. $\overline{\text{PCREQ}}$ is available on MFUNC7, MFUNC4, or MFUNC0. This terminal is also used for the serial EEPROM interface.
$\overline{\text{PCREQ}}$ (MFUNC4)	W9	T9		
$\overline{\text{PCREQ}}$ (MFUNC0)	W11	W10		

Table 2–14. Zoomed Video Terminals

TERMINAL			I/O AND MEMORY INTERFACE SIGNAL	I/O	FUNCTION
NAME	GFN NO.	GJG NO.			
ZV_HREF	P3	N1	A10	O	Horizontal sync to the zoomed video port
ZV_LRCLK	V4	V3	$\overline{\text{INPACK}}$	O	Audio LRCLK PCM
ZV_MCLK	W4	W3	A6	O	Audio MCLK PCM
ZV_PCLK	Y3	V4	$\overline{\text{IOIS16}}$	O	Pixel clock to the zoomed video port
ZV_SCLK	W3	V2	A7	O	Audio SCLK PCM
ZV_SDATA	U5	W4	$\overline{\text{SPKR}}$	O	Audio SDATA PCM
ZV_UV7	Y2	W2	A25	O	Video data to the zoomed video port in YV:4:2:2 format
ZV_UV6	W2	U2	A12		
ZV_UV5	Y1	V1	A24		
ZV_UV4	W1	T2	A15		
ZV_UV3	V3	U1	A23		
ZV_UV2	U3	R4	A16		
ZV_UV1	V2	T1	A22		
ZV_UV0	T4	R2	A21		
ZV_VSYNC	R2	N2	A11	O	Vertical sync to the zoomed video port
ZV_Y7	V1	R1	A20	O	Video data to the zoomed video port in YV:4:2:2 format
ZV_Y6	U2	P6	A14		
ZV_Y5	T3	P5	A19		
ZV_Y4	U1	P4	A13		
ZV_Y3	T2	P2	A18		
ZV_Y2	R3	N6	A8		
ZV_Y1	P4	N5	A17		
ZV_Y0	T1	N4	A9		

Table 2–15. Miscellaneous Terminals

TERMINAL			I/O	FUNCTION
NAME	GFN NO.	GJG NO.		
MFUNC0	W11	W10	I/O	Multifunction terminal 0. Defaults as a general-purpose input (GPI0), and can be programmed to perform various functions. See Section 4.36, *Multifunction Routing Status Register,* for configuration details.
MFUNC1	Y11	P10	I/O	Multifunction terminal 1. Defaults as a general-purpose input (GPI1), and can be programmed to perform various functions. See Section 4.36, *Multifunction Routing Status Register,* for configuration details.
MFUNC2	Y10	P9	I/O	Multifunction terminal 2. Defaults as a general-purpose input (GPI2), and can be programmed to perform various functions. See Section 4.36, *Multifunction Routing Status Register,* for configuration details.
MFUNC3	V10	R9	I/O	Multifunction terminal 3. Defaults as a general-purpose input (GPI3), and can be programmed to perform various functions. See Section 4.36, *Multifunction Routing Status Register,* for configuration details.
MFUNC4	W9	T9	I/O	Multifunction terminal 4. Defaults as a high-impedance reserved input, and can be programmed to perform various functions. See Section 4.36, *Multifunction Routing Status Register,* for configuration details.
MFUNC5	V5	W5	I/O	Multifunction terminal 5. Defaults as a high-impedance reserved input, and can be programmed to perform various functions. See Section 4.36, *Multifunction Routing Status Register,* for configuration details.
MFUNC6	Y4	R5	I/O	Multifunction terminal 6. Defaults as a high-impedance reserved input, and can be programmed to perform various functions. See Section 4.36, *Multifunction Routing Status Register,* for configuration details.
IDSEL/MFUNC7	P20	N19	I/O	IDSEL and multifunction terminal 7. Defaults as IDSEL, but may be used as a multifunction terminal. See Section 4.36, *Multifunction Routing Status Register* and Section 3.4, *PC Card Applications Overview*, for configuration details.
SCL	W10	V9	I/O	Serial ROM clock. This terminal provides the SCL serial clock signaling in a two-wire serial ROM implementation, and is sensed at reset for serial ROM detection.
SDA	Y9	W9	I/O	Serial ROM data. This terminal provides the SDA serial data signaling in a two-wire serial ROM implementation.
$\overline{\text{SPKROUT}}$	V11	T10	O	Speaker output. SPKROUT is the output to the host system that can carry $\overline{\text{SPKR}}$ or CAUDIO through the PCI4450 from the PC Card interface. SPKROUT is driven as the XOR combination of card $\overline{\text{SPKR}}$//CAUDIO inputs.
$\overline{\text{SUSPEND}}$	U11	R10	I	Suspend. $\overline{\text{SUSPEND}}$ is used to protect the internal registers from clearing when $\overline{\text{PRST}}$ is asserted. See Section 3.6.6, *Suspend Mode* for details.

Table 2–16. 16-Bit PC Card Address and Data Terminals (slots A and B)

TERMINAL					I/O	FUNCTION
NAME	GFN NO.		GJG NO.			
	SLOT A†	SLOT B‡	SLOT A†	SLOT B‡		
ADDR25	G2	A16	G4	E14	O	PC Card address. 16-bit PC Card address lines. A25 is the most significant bit.
ADDR24	F1	C16	G5	A15		
ADDR23	G4	A18	F2	D15		
ADDR22	E2	C17	F5	B17		
ADDR21	D1	B18	E5	A18		
ADDR20	E4	A20	D2	B19		
ADDR19	D2	C18	C2	D16		
ADDR18	B1	C19	B2	D18		
ADDR17	A2	E17	B3	E16		
ADDR16	E3	B17	E2	A17		
ADDR15	E1	D16	F4	B16		
ADDR14	D3	B19	D1	C18		
ADDR13	C2	B20	B1	C19		
ADDR12	F2	A17	G6	E15		
ADDR11	C4	E18	A4	F15		
ADDR10	C5	E20	F6	G15		
ADDR9	B2	C20	D4	E18		
ADDR8	C3	D18	A2	D19		
ADDR7	G3	B16	G2	B15		
ADDR6	H3	D14	H1	B14		
ADDR5	H1	A15	H4	E13		
ADDR4	J4	C14	H5	B13		
ADDR3	J2	A14	J1	D13		
ADDR2	K2	A13	J5	D12		
ADDR1	K3	D12	J6	E12		
ADDR0	K1	C12	K6	D11		
DATA15	C6	G18	A6	G18	I/O	PC Card data. 16-bit PC Card data lines. D15 is the most significant bit.
DATA14	A5	G19	E7	H15		
DATA13	C7	H18	B7	H18		
DATA12	B7	H20	G7	H14		
DATA11	C8	J18	B8	J16		
DATA10	R1	D9	N7	E9		
DATA9	P1	B9	M4	B9		
DATA8	N2	D10	M6	F9		
DATA7	D7	F20	F7	G13		
DATA6	B6	G20	D7	H16		
DATA5	A6	H19	A7	H19		
DATA4	A7	J17	E8	J15		
DATA3	B8	J19	A8	J18		
DATA2	P2	C9	M5	D9		
DATA1	N3	A9	M1	A9		
DATA0	N1	C10	L5	E10		

† Terminal name for slot A is preceded with A_. For example, the full name for GFN terminal G2 is A_ADDR25.
‡ Terminal name for slot B is preceded with B_. For example, the full name for GFN terminal A16 is B_ADDR25.

Table 2–17. 16-Bit PC Card Interface Control Terminals (slots A and B)

TERMINAL					I/O	FUNCTION
NAME	GFN NO.		GJG NO.			
	SLOT A†	SLOT B‡	SLOT A†	SLOT B‡		
BVD1 (STSCHG/RI)	M2	A11	L1	A10	I	Battery voltage detect 1. BVD1 is generated by 16-bit memory PC Cards that include batteries. BVD1 and BVD2 indicate the condition of the batteries on a memory PC Card. Both BVD1 and BVD2 are kept 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 the 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)	M1	C11	L6	F10	I	Battery voltage detect 2. BVD2 is generated by 16-bit memory PC Cards that include batteries. BVD2 and BVD1 indicate 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 the 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 PCI4450 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	A8 M4	J20 B10	F8 L4	J19 D10	I	PC Card detect 1 and PC Card detect 2. CD1 and CD2 are internally connected to ground on the PC Card. When a PC Card is inserted into a socket, CD1 and CD2 are pulled low. For signal status, see Section 5.2, *ExCA Interface Status Register*.
CE1 CE2	A4 B4	F19 G17	D6 A5	G16 G14	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	J3	B14	J4	A13	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 used as a strobe, then the PC Card asserts this signal to indicate a request for a DMA operation.
IORD	D5	D20	D5	F16	O	I/O read. IORD is asserted by the PCI4450 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 PCI4450 asserts IORD during DMA transfers from the PC Card to host memory.
IOWR	B3	D19	B4	F14	O	I/O write. IOWR is driven low by the PCI4450 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 PCI4450 asserts IOWR during transfers from host memory to the PC Card.

† Terminal name for slot A is preceded with A_. For example, the full name for GFN terminal D5 is A_IORD.
‡ Terminal name for slot B is preceded with B_. For example, the full name for GFN terminal D20 is B_IORD.

Table 2–17. 16-Bit PC Card Interface Control Terminals (slots A and B) (continued)

TERMINAL					I/O	FUNCTION
NAME	GFN NO.		GJG NO.			
	SLOT A†	SLOT B‡	SLOT A†	SLOT B‡		
$\overline{OE}$	A3	F18	B5	F19	O	Output enable. $\overline{OE}$ is driven low by the PCI4450 to enable 16-bit memory PC Card data output during host memory read cycles. DMA terminal count. $\overline{OE}$ is used as terminal count (TC) during DMA operations to a 16-bit PC Card that supports DMA. The PCI4450 asserts $\overline{OE}$ to indicate TC for a DMA write operation.
READY ($\overline{IREQ}$)	L2	A12	K2	E11	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. $\overline{IREQ}$ 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. $\overline{IREQ}$ is high (deasserted) when no interrupt is requested.
$\overline{REG}$	J1	C13	J2	A12	O	Attribute memory select. $\overline{REG}$ remains high for all common memory accesses. When $\overline{REG}$ is asserted, access is limited to attribute memory ($\overline{OE}$ or $\overline{WE}$ active) and to the I/O space ($\overline{IORD}$ or $\overline{IOWR}$ active). Attribute memory is a separately accessed section of card memory and is generally used to record card capacity and other configuration and attribute information. DMA acknowledge. $\overline{REG}$ is used as a DMA acknowledge ($\overline{DACK}$) during DMA operations to a 16-bit PC Card that supports DMA. The PCI4450 asserts $\overline{REG}$ to indicate a DMA operation. $\overline{REG}$ is used in conjunction with the DMA read ($\overline{IOWR}$) or DMA write ($\overline{IORD}$) strobes to transfer data.
RESET	H2	B15	H2	F13	O	PC Card reset. RESET forces a hard reset to a 16-bit PC Card.
$\overline{VS1}$ $\overline{VS2}$	L1 G1	B12 C15	K1 H6	B11 A14	I/O	Voltage sense 1 and voltage sense 2. $\overline{VS1}$ and $\overline{VS2}$, when used in conjunction with each other, determine the operating voltage of the 16-bit PC Card.
$\overline{WAIT}$	L3	B11	K4	F11	I	Bus cycle wait. $\overline{WAIT}$ is driven by a 16-bit PC Card to delay the completion of (i.e., extend) the memory or I/O cycle in progress.
$\overline{WE}$	C1	A19	E4	B18	O	Write enable. $\overline{WE}$ is used to strobe memory write data into 16-bit memory PC Cards. $\overline{WE}$ is also used for memory PC Cards that employ programmable memory technologies. DMA terminal count. $\overline{WE}$ is used as TC during DMA operations to a 16-bit PC Card that supports DMA. The PCI4450 asserts $\overline{WE}$ to indicate TC for a DMA read operation.
WP ($\overline{IOIS16}$)	M3	A10	L2	B10	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 cards, WP is used for the 16-bit port ($\overline{IOIS16}$) function. I/O is 16 bits. $\overline{IOIS16}$ applies to 16-bit I/O PC Cards. $\overline{IOIS16}$ 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.

† Terminal name for slot A is preceded with A_. For example, the full name for GFN terminal C1 is A_$\overline{WE}$.
‡ Terminal name for slot B is preceded with B_. For example, the full name for GFN terminal A19 is B_$\overline{WE}$.

Table 2–18. CardBus PC Card Interface System Terminals (slots A and B)

TERMINAL					I/O	FUNCTION
NAME	GFN NO.		GJG NO.			
	SLOT A†	SLOT B‡	SLOT A†	SLOT B‡		
CCLK	E3	B17	E2	A17	O	CardBus PC Card clock. CCLK provides synchronous timing for all transactions on the CardBus interface. All signals except $\overline{\text{CRST}}$, $\overline{\text{CCLKRUN}}$, $\overline{\text{CINT}}$, CSTSCHG, CAUDIO, $\overline{\text{CCD2}}$, $\overline{\text{CCD1}}$, 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.
$\overline{\text{CCLKRUN}}$	M3	A10	L2	B10	O	CardBus PC Card clock run. $\overline{\text{CCLKRUN}}$ is used by a CardBus PC Card to request an increase in the CCLK frequency, and by the PCI4450 to indicate that the CCLK frequency is decreased. CardBus clock run ($\overline{\text{CCLKRUN}}$) follows the PCI clock run ($\overline{\text{CLKRUN}}$).
$\overline{\text{CRST}}$	H2	B15	H2	F13	I/O	CardBus PC Card reset. $\overline{\text{CRST}}$ is used to bring CardBus PC Card-specific registers, sequencers, and signals to a known state. When $\overline{\text{CRST}}$ is asserted, all CardBus PC Card signals must be placed in a high-impedance state, and the PCI4450 drives these signals to a valid logic level. Assertion can be asynchronous to CCLK, but deassertion must be synchronous to CCLK.

† Terminal name for slot A is preceded with A_. For example, the full name for GFN terminal E3 is A_CCLK.
‡ Terminal name for slot B is preceded with B_. For example, the full name for GFN terminal B17 is B_CCLK.

Table 2–19. CardBus PC Card Address and Data Terminals (slots A and B)

TERMINAL					I/O	FUNCTION
NAME	GFN NO.		GJG NO.			
	SLOT A†	SLOT B‡	SLOT A†	SLOT B‡		
CAD31	R1	D9	N7	E9	I/O	PC Card address and data. These signals make up the multiplexed CardBus address and data bus on the CardBus interface. During the address phase of a CardBus cycle, CAD31–CAD0 contain a 32-bit address. During the data phase of a CardBus cycle, CAD31–CAD0 contain data. CAD31 is the most significant bit.
CAD30	P1	B9	M4	B9		
CAD29	N3	A9	M1	A9		
CAD28	N2	D10	M6	F9		
CAD27	N1	C10	L5	E10		
CAD26	K1	C12	K6	D11		
CAD25	K3	D12	J6	E12		
CAD24	K2	A13	J5	D12		
CAD23	J2	A14	J1	D13		
CAD22	J4	C14	H5	B13		
CAD21	H1	A15	H4	E13		
CAD20	H3	D14	H1	B14		
CAD19	G2	A16	G4	E14		
CAD18	G3	B16	G2	B15		
CAD17	F1	C16	G5	A15		
CAD16	A2	E17	B3	E16		
CAD15	B3	D19	B4	F14		
CAD14	B2	C20	D4	E18		
CAD13	D5	D20	D5	F16		
CAD12	C4	E18	A4	F15		
CAD11	A3	F18	B5	F19		
CAD10	B4	G17	A5	G14		
CAD9	C5	E20	F6	G15		
CAD8	C6	G18	A6	G18		
CAD7	D7	F20	F7	G13		
CAD6	C7	H18	B7	H18		
CAD5	B6	G20	D7	H16		
CAD4	B7	H20	G7	H14		
CAD3	A6	H19	A7	H19		
CAD2	C8	J18	B8	J16		
CAD1	A7	J17	E8	J15		
CAD0	B8	J19	A8	J18		
CC/$\overline{BE3}$	J1	C13	J2	A12	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/$\overline{BE0}$ defines 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).
CC/$\overline{BE2}$	F2	A17	G6	E15		
CC/$\overline{BE1}$	C3	D18	A2	D19		
CC/$\overline{BE0}$	A4	F19	D6	G16		
CPAR	C2	B20	B1	C19	I/O	CardBus parity. In all CardBus read and write cycles, the PCI4450 calculates even parity across the CAD and CC/$\overline{BE}$ buses. As an initiator during CardBus cycles, the PCI4450 outputs CPAR with a one-CCLK delay. As a target during CardBus cycles, the calculated parity is compared to the initiator parity indicator; a compare error results in a parity error assertion.

† Terminal name for slot A is preceded with A_. For example, the full name for GFN terminal C2 is A_CPAR.
‡ Terminal name for slot B is preceded with B_. For example, the full name for GFN terminal B20 is B_CPAR.

Table 2–20. CardBus PC Card Interface Control Terminals (slots A and B)

TERMINAL					I/O	FUNCTION
NAME	GFN NO.		GJG NO.			
	SLOT A†	SLOT B‡	SLOT A†	SLOT B‡		
CAUDIO	M1	C11	L6	F10	I	CardBus audio. CAUDIO is a digital input signal from a PC Card to the system speaker. The PCI4450 supports the binary audio mode and outputs a binary signal from the card to SPKROUT.
$\overline{\text{CBLOCK}}$	D2	C18	C2	D16	I/O	CardBus lock. $\overline{\text{CBLOCK}}$ is used to gain exclusive access to a target.
$\overline{\text{CCD1}}$ $\overline{\text{CCD2}}$	A8 M4	J20 B10	F8 L4	J19 D10	I	CardBus detect 1 and CardBus detect 2. $\overline{\text{CCD1}}$ and $\overline{\text{CCD2}}$ are used in conjunction with CVS1 and CVS2 to identify card insertion and interrogate cards to determine the operating voltage and card type.
$\overline{\text{CDEVSEL}}$	D1	B18	E5	A18	I/O	CardBus device select. The PCI4450 asserts $\overline{\text{CDEVSEL}}$ to claim a CardBus cycle as the target device. As a CardBus initiator on the bus, the PCI4450 monitors $\overline{\text{CDEVSEL}}$ until a target responds. If no target responds before timeout occurs, then the PCI4450 terminates the cycle with an initiator abort.
$\overline{\text{CFRAME}}$	G4	A18	F2	D15	I/O	CardBus cycle frame. $\overline{\text{CFRAME}}$ is driven by the initiator of a CardBus bus cycle. $\overline{\text{CFRAME}}$ is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When $\overline{\text{CFRAME}}$ is deasserted, the CardBus bus transaction is in the final data phase.
$\overline{\text{CGNT}}$	C1	A19	E4	B18	I	CardBus bus grant. $\overline{\text{CGNT}}$ is driven by the PCI4450 to grant a CardBus PC Card access to the CardBus bus after the current data transaction has been completed.
$\overline{\text{CINT}}$	L2	A12	K2	E11	I	CardBus interrupt. $\overline{\text{CINT}}$ is asserted low by a CardBus PC Card to request interrupt servicing from the host.
$\overline{\text{CIRDY}}$	E1	D16	F4	B16	I/O	CardBus initiator ready. $\overline{\text{CIRDY}}$ indicates the ability of the CardBus initiator to complete the current data phase of the transaction. A data phase is completed on a rising edge of CCLK when both $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are asserted. Until $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are both sampled asserted, wait states are inserted.
$\overline{\text{CPERR}}$	D3	B19	D1	C18	I/O	CardBus parity error. $\overline{\text{CPERR}}$ reports parity errors during CardBus transactions, except during special cycles. It is driven low by a target two clocks following that data when a parity error is detected.
$\overline{\text{CREQ}}$	J3	B14	J4	A13	I	CardBus request. $\overline{\text{CREQ}}$ indicates to the arbiter that the CardBus PC Card desires use of the CardBus bus as an initiator.
$\overline{\text{CSERR}}$	L3	B11	K4	F11	I	CardBus system error. $\overline{\text{CSERR}}$ reports address parity errors and other system errors that could lead to catastrophic results. $\overline{\text{CSERR}}$ is driven by the card synchronously with CCLK, but deasserted by a weak pullup, and may take several CCLK periods. The PCI4450 can report $\overline{\text{CSERR}}$ to the system by assertion of $\overline{\text{SERR}}$ on the PCI interface.
$\overline{\text{CSTOP}}$	E4	A20	D2	B19	I/O	CardBus stop. $\overline{\text{CSTOP}}$ is driven by a CardBus target to request the initiator to stop the current CardBus transaction. $\overline{\text{CSTOP}}$ is used for target disconnects, and is commonly asserted by target devices that do not support burst data transfers.
CSTSCHG	M2	A11	L1	A10	I	CardBus status change. CSTSCHG alerts the system to a change in the card status and is used as a wake-up mechanism.
$\overline{\text{CTRDY}}$	E2	C17	F5	B17	I/O	CardBus target ready. $\overline{\text{CTRDY}}$ indicates the ability of the CardBus target to complete the current data phase of the transaction. A data phase is completed on a rising edge of CCLK, when both $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are asserted; until this time, wait states are inserted.
CVS1 CVS2	L1 G1	B12 C15	K1 H6	B11 A14	I/O	CardBus voltage sense 1 and CardBus voltage sense 2. CVS1 and CVS2 are used in conjunction with $\overline{\text{CCD1}}$ and $\overline{\text{CCD2}}$ to identify card insertion and interrogate cards to determine the operating voltage and card type.

† Terminal name for slot A is preceded with A_. For example, the full name for GFN terminal M1 is A_CAUDIO.

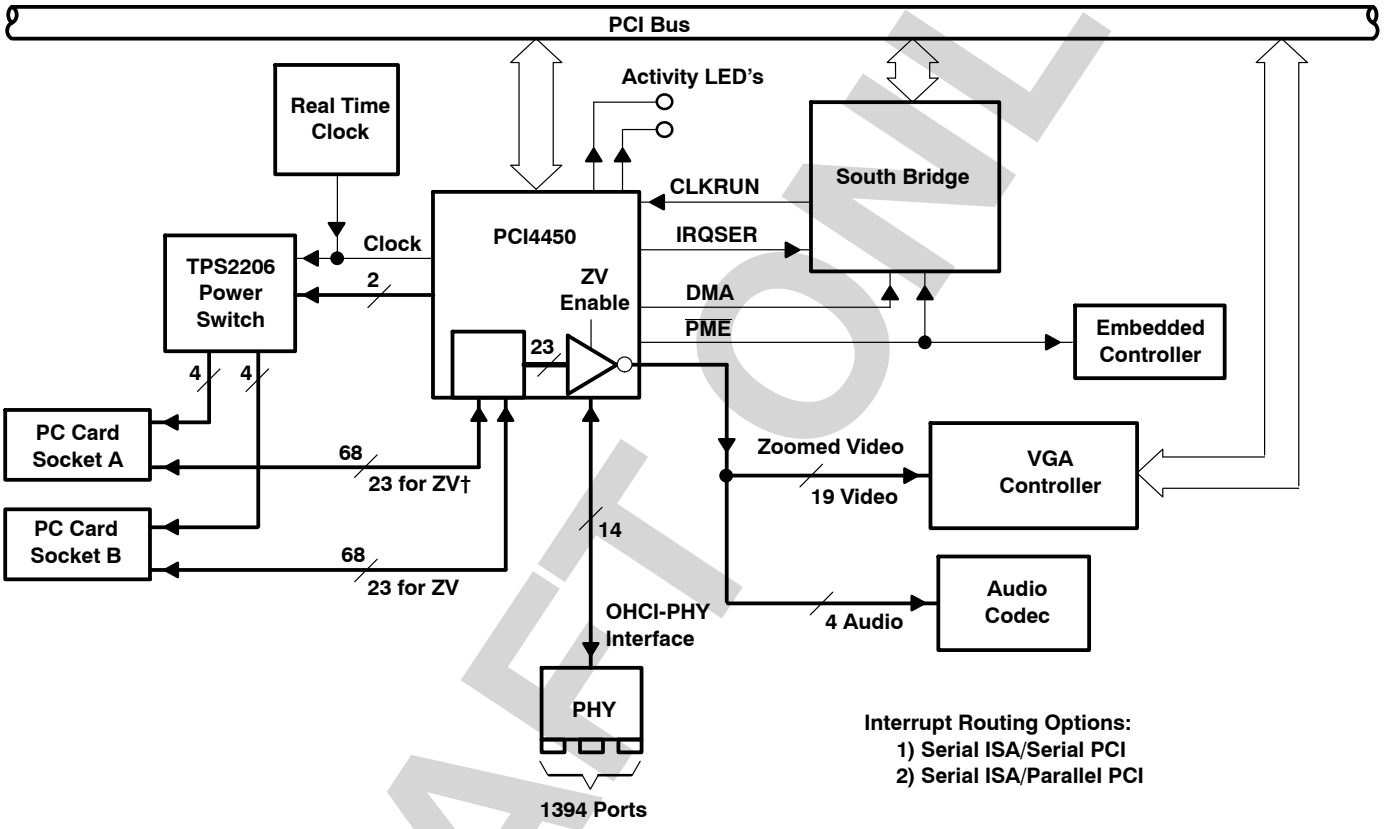
‡ Terminal name for slot B is preceded with B_. For example, the full name for GFN terminal C11 is B_CAUDIO.

Table 2–21. IEEE 1394 PHY/Link Interface Terminals

TERMINAL			I/O	FUNCTION
NAME	GFN NO.	GJG NO.		
LINKON	Y6	P7	I	1394 link on. This input from the PHY indicates that the link should turn on.
LPS	W5	V5	O	Link power status. LPS indicates that link is powered and fully functional.
PHY_CLK	V6	T6	I	System clock. This input provides a 49.152-MHz clock signal for data synchronization.
PHY_CTL1 PHY_CTL0	W6 U7	V6 W6	I/O	PHY-link interface control. These bidirectional signals control passage of information between the PHY and link. The link can only drive these terminals after the PHY has granted permission following a link request (LREQ).
PHY_DATA7 PHY_DATA6 PHY_DATA5 PHY_DATA4 PHY_DATA3 PHY_DATA2 PHY_DATA1 PHY_DATA0	V9 U9 Y8 W8 V8 Y7 W7 V7	R8 T8 V8 W8 P8 W7 V7 R7	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_LREQ	Y5	R6	O	Link request. This signal is driven by the link to initiate a request for the PHY to perform some service.

3 Feature/Protocol Descriptions

Figure 3–1 shows a simplified system implementation example using the PCI4450. The PCI interface includes all address/data and control signals for PCI protocol. Highlighted in this diagram is the functionality supported by the PCI4450. The PCI4450 supports PC/PCI DMA, PCI Way DMA (distributed DMA), $\overline{\text{PME}}$ wake-up from $D3_{cold}$ through D0, 4 interrupt modes, an integrated zoomed video port, and 12 multifunction pins (8 MFUNC, and 4 GPIO pins) that can be programmed for a wide variety of functions.



† The PC Card interface is 68 pins for CardBus and 16-bit PC Cards. In zoomed-video mode 23 pins are used for routing the zoomed video signals to the VGA controller.

Figure 3–1. PCI4450 System Block Diagram

3.1 I/O Characteristics

Figure 3–2 shows a 3-state bidirectional buffer illustration for reference. The table, *recommended operating conditions* provides the electrical characteristics of the inputs and outputs. The PCI4450 meets the ac specifications of the *1995 PC Card Standard* and the *PCI Local Bus Specification*.

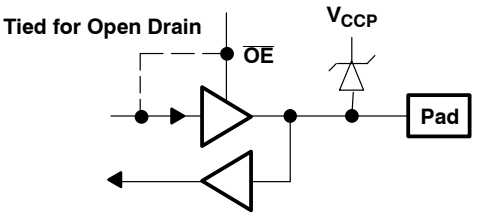


Figure 3–2. 3-State Bidirectional Buffer

3.2 Clamping Voltages

The I/O sites can be pulled through a clamping diode to a voltage rail for protection. The 3.3-V core power supply is independent of the clamping voltages. The clamping (protection) diodes are required if the signaling environment on an I/O is system dependent. For example, PCI signaling can be either 3.3 Vdc or 5.0 Vdc, and the PCI4450 must reliably accommodate both voltage levels. This is accomplished by using a 3.3-V buffer with a clamping diode to V_{CCP}. If a system design requires a 5.0-V PCI bus, then the V_{CCP} would be connected to the 5.0-V power supply.

A standard die has only one clamping voltage for the sites as shown in Figure 3–2. After the terminal assignments are fixed, the fabrication facility will support a design by splitting the clamping voltage for customization. The PCI4450 requires five separate clamping voltages since it supports a wide range of features. The five voltages are listed and defined in the table, *recommended operating conditions*.

3.3 Peripheral Component Interconnect (PCI) Interface

This section describes the PCI interface of the PCI4450, and how the device responds to and participates in PCI bus cycles. The PCI4450 provides all required signals for PCI master/slave devices and may operate in either 5-V or 3.3-V PCI signaling environments by connecting the V_{CCP} terminals to the desired signaling level.

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

The bus locking protocol defined in the PCI Specification is not highly recommended, but is provided on the PCI4450 as an additional compatibility feature. The PCI $\overline{LOCK}$ terminal is multiplexed with GPIO2, and the terminal function defaults to a general-purpose input (GPI). The use of $\overline{LOCK}$ is only supported 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 may proceed to an address that is not currently locked. A grant to start a transaction on the PCI bus does not guarantee control of $\overline{LOCK}$; 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 $\overline{LOCK}$. To avoid confusion with the PCI bus clock, the CardBus signal for this protocol is $\overline{CBLOCK}$.

An agent may need to do an exclusive operation because a critical memory 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 lock protocol defined by PCI 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 PCI4450 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 using devices such as PCI-to-PCI bridges. The potential deadlock can occur if a CardBus target

supports delayed transactions and blocks access as the target until it completes a delayed read. This target characteristic is prohibited by the 2.1 PCI Specification, and the issue is resolved by the PCI master using $\overline{\text{LOCK}}$.

3.3.2 Loading The Subsystem Identification (EEPROM Interface)

The subsystem vendor ID register and subsystem ID register make up a double word of PCI configuration space located at offset 40h for functions 0 and 1. This doubleword register, used for system and option card (mobile dock) identification purposes, is required by some operating systems. Implementation of this unique identifier register is a PC ‘97 requirement.

The PCI4450 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 the access mode may be made read/write by clearing the SUBSYSRW bit in the system control register (bit 5 of the system control register, offset 80h). Once this bit is cleared (0), the BIOS may write a subsystem identification value into the registers at offset 40h. The BIOS must set 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 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 through a serial EEPROM interface. The PCI4450 loads the double-word of data from the serial EEPROM after a reset of the primary bus. The $\overline{\text{SUSPEND}}$ input gates the $\overline{\text{PRST}}$ and $\overline{\text{G\_RST}}$ from the entire PCI4450 core, including the serial EEPROM state machine. Refer to *suspend mode* for details on using $\overline{\text{SUSPEND}}$. The PCI4450 provides a two-line serial bus interface to the serial EEPROM.

The system designer must implement a pulldown resistor on the PCI4450 LATCH terminal to indicate the serial EEPROM mode. Only when this pulldown resistor is present will the PCI4450 attempt to load data through the serial EEPROM interface. The serial EEPROM interface is a two-pin interface with one data signal (SDA) and one clock signal (SCL). Figure 3–3 illustrates a typical PCI4450 application using the serial EEPROM interface.

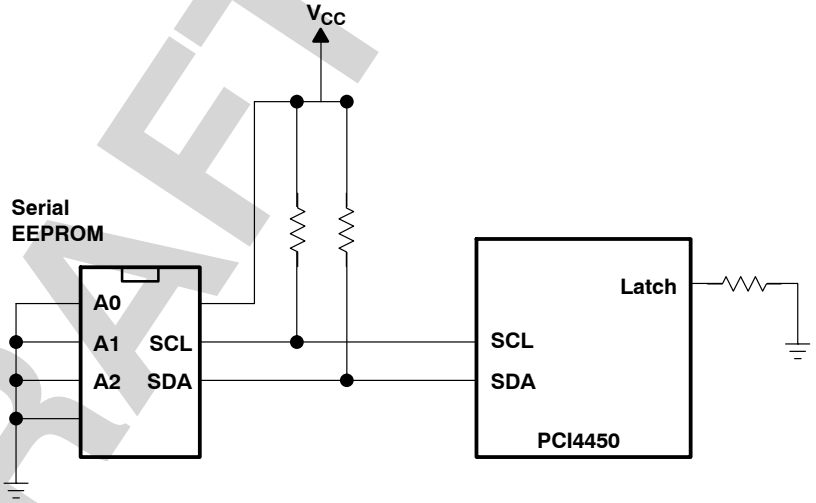


Figure 3–3. Serial EEPROM Application

As stated above, when the PCI4450 is reset by $\overline{\text{G\_RST}}$, the subsystem data is read automatically from the EEPROM. The PCI4450 masters the serial EEPROM bus and reads four bytes as described in Figure 3–4.

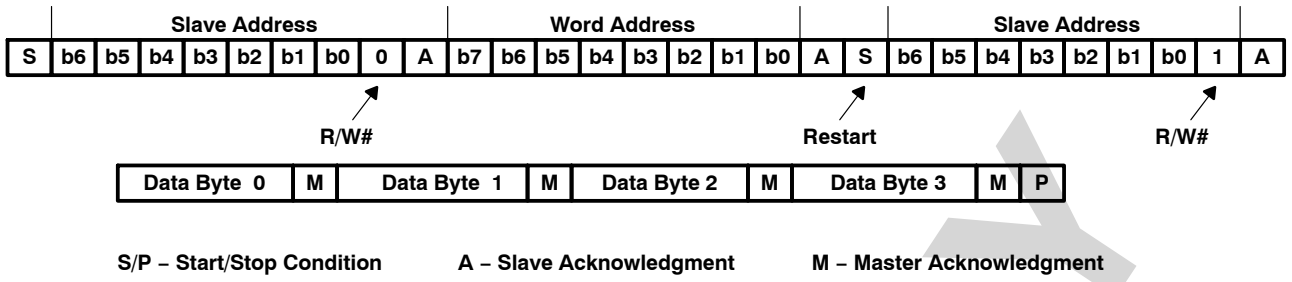


Figure 3–4. EEPROM Interface Subsystem Data Collection

The EEPROM is addressed at slave address A0h (1010 0000b), as indicated in Figure 3–4, and the EEPROM word address auto-increments after each byte transfers according to the protocol. All hardware address bits for the EEPROM should be tied to the appropriate level to achieve this slave address. Thus, to provide the subsystem register with data AABBCCDDh the EEPROM should be programmed with address 0 = AAh, 1 = BBh, 2 = CCh, and 3 = DDh.

The serial EEPROM chip in the sample application circuit, Figure 3–3, 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.

The serial EEPROM interface signals require pullup resistors. The serial EEPROM protocol allows bidirectional transfers. Both the SCL and SDA signals are placed in a high-impedance state and pulled high when the bus is not active. A high-to-low transition of the SDA line defines a start condition (S). A low-to-high transition of SDA while SCL is high is defined as the stop condition (P). One bit is transferred during each clock pulse. The data on the SDA line must remain stable during the high period of the clock pulse, as changes in the data line at this time will be interpreted as a control signal. Data is valid and stable during the clock high period. Figure 3–5 illustrates this protocol.

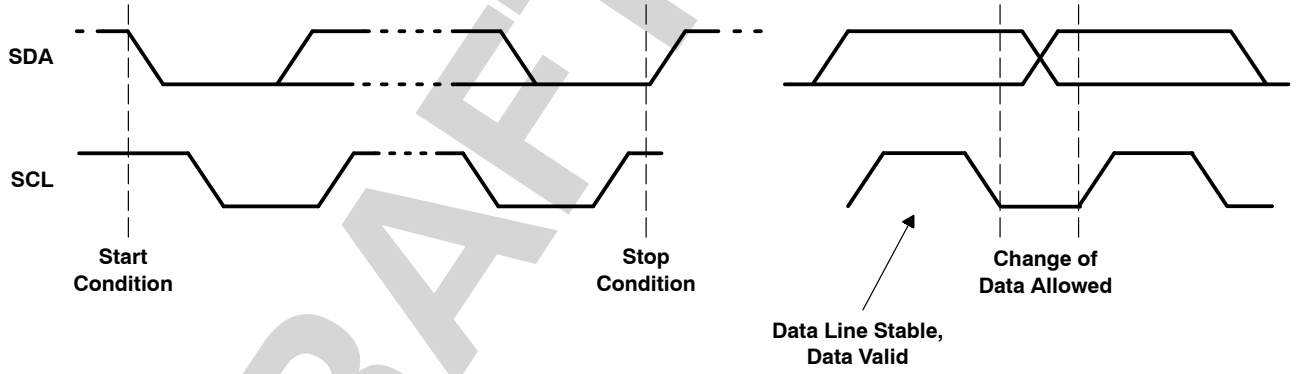


Figure 3–5. Serial EEPROM Start/Stop Conditions and Bit Transfers

Each address byte and data transfer is followed by an acknowledge bit, as indicated in Figure 3–4. When the PCI4450 transmits the addresses, it returns the SDA signal to the high state and places the line in a high-impedance state. The PCI4450 then generates an SCL clock cycle and expects the EEPROM to pull down the SDA line during the acknowledge pulse. This procedure is referred to as a slave acknowledge with the PCI4450 transmitter and the EEPROM receiver. Figure 3–6 illustrates general acknowledges.

During the data byte transfers from the serial EEPROM to the PCI4450, the EEPROM clocks the SCL signal. After the EEPROM transmits the data to the PCI4450, it returns the SDA signal to the high state and places the line in a high-impedance state. The EEPROM then generates an SCL clock cycle and expects the PCI4450 to pull down the SDA line during the acknowledge pulse. This procedure is referred to as a master acknowledge with the EEPROM transmitter and the PCI4450 receiver. Figure 3–6 illustrates general acknowledges.

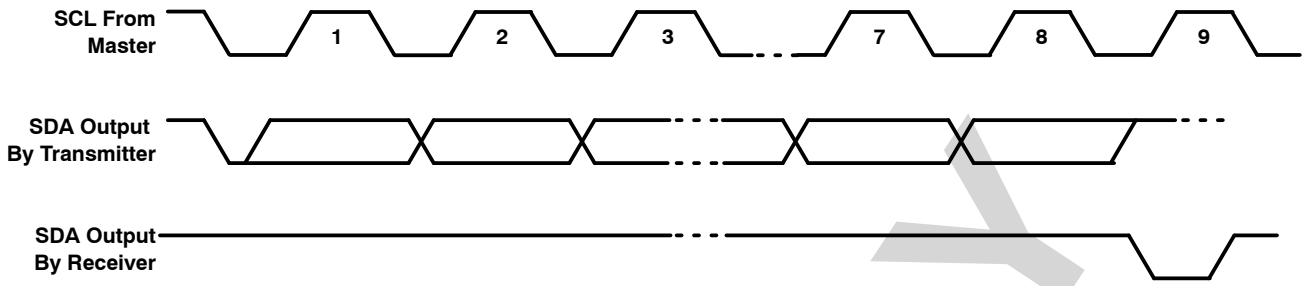


Figure 3–6. Serial EEPROM Protocol – Acknowledge

EEPROM interface status information is communicated through the general status register located at PCI offset 85h. The EEDETECT bit in this register indicates whether or not the PCI4450 serial EEPROM circuitry detects the pulldown resistor on LATCH. An error condition, such as a missing acknowledge, results in the DATAERR bit being set. The EEBUSY bit is set while the subsystem ID register is loading (serial EEPROM interface is busy).

3.3.3 Serial ROM Implementation

A serial ROM interface exists in both the Open HCI function and the PC Card controller functions. The PCI4450 implementation adds a busy indication between the interfaces to allow the function 2 loading to follow the functions 0 and 1 load. All serial ROM addressing uses slave address 8'hA0. The functions 0 and 1 serial EEPROM state machine is modified to provide a busy indication to function 2 and to start loading registers at word address 8'h20 to allow for some serial ROM format flexibility in function 2.

Primarily, the serial ROM is used to preload the PCI4450 registers with data, and only write accessible bits in these registers may be preloaded. Figure 3–7 illustrates the PCI4450 serial ROM data format, which is an expanded version of both the OHCI-Lynx and PCI1450 serial ROM formats.

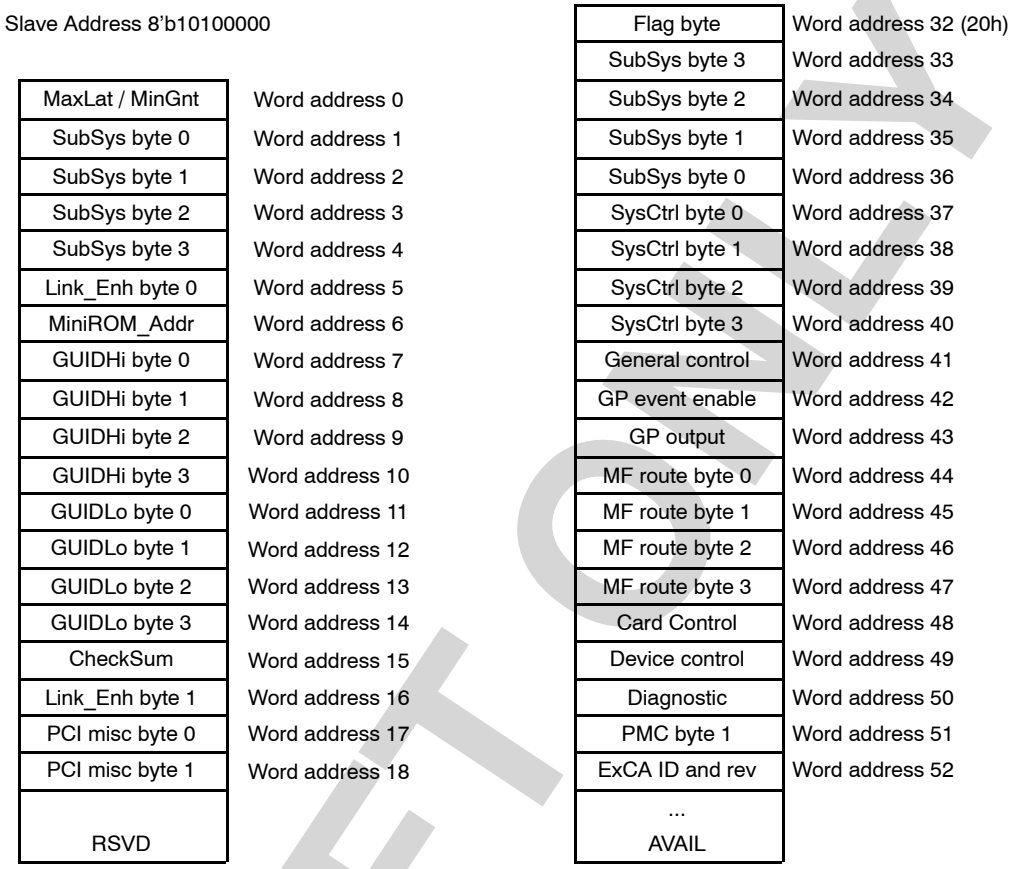


Figure 3–7. Serial ROM Data Format

The flag byte at word address 32 indicates to the PCI4450 whether or not the PC Card controller functions loads the data from word address range 33–52. A flag byte set to 8'hFF indicates to stop loading the serial ROM data for functions 0 and 1, but is independent of the function 2 1394 Open HCI controller load from word address range 0–18.

An additional change in the serial ROM behavior with respect to Open HCI GUIDROM register access is the MiniROM_Addr. The MiniROM_Addr field in the ROM data is loaded from byte location 6 (EEPROM word address 6) and indicates to function 2 where to begin accessing the serial ROM via the GUID ROM register. The GUIDROM.addrReset bit function changes slightly to reset serial ROM access to the byte location indicated by MiniROM_Addr. A MiniROM_Addr value of zero provides identical operation as the OHCI-Lynx.

3.4 PC Card Applications Overview

This section describes the PC Card interfaces of the PCI4450. A discussion on PC Card recognition details the card interrogation procedure. This section discusses the card powering procedure, including the protocol of the P^2C power switch interface. The internal ZV buffering provided by the PCI4450 and programming model is detailed in this section. Also, standard PC Card register models are described, as well as a brief discussion of the PC Card software protocol layers.

3.4.1 PC Card Insertion/Removal and Recognition

The 1995 PC Card Standard addresses the card detection and recognition process through an interrogation procedure that the socket must initiate upon card insertion into a cold, unpowered socket. Through this interrogation, card voltage requirements and interface (16-bit vs. CardBus) are determined.

The scheme uses the $\overline{CD1}$, $\overline{CD2}$, $\overline{VS1}$, and $\overline{VS2}$ signals ($\overline{CCD1}$, $\overline{CCD2}$, CVS1, CVS2 for CardBus). A PC Card designer connects these four pins in a certain configuration depending on the type of card and the supply voltage. The encoding scheme for this, defined in the 1997 PC Card Standard, is shown in Table 3–1.

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

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

3.4.2 P^{2}C Power Switch Interface (TPS2202A/2206)

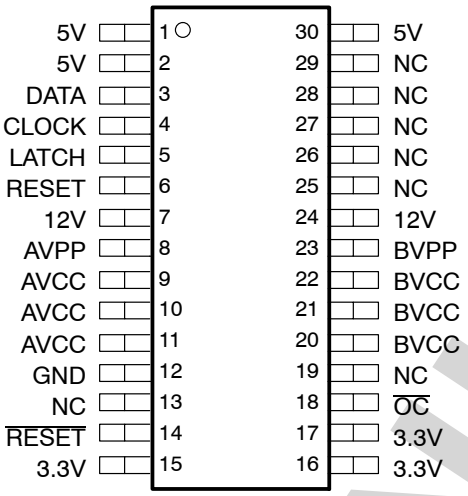
A power switch with a PCMCIA-to-peripheral control (P^{2}C) interface is required for the PC Card powering interface. The TI TPS2206 (or TPS2202A) Dual-Slot PC Card Power-Interface Switch provides the P^{2}C interface to the CLOCK, DATA, and LATCH terminals of the PCI4450. Figure 3–8 shows the terminal assignments of the TPS2206. Figure 3–9 illustrates a typical application where the PCI4450 represents the PCMCIA controller.

There are two ways to provide a clock source to the power switch interface. The first method is to provide an external clock source such as a 32 kHz real time clock to the CLOCK terminal. The second method is to use the internal ring oscillator. If the internal ring oscillator is used, then the PCI4450 provides its own clock source for the PC Card interrogation logic and the power switch interface. The mode of operation is determined by the setting of bit 27 of the system control register (PCI offset 80h). This bit is encoded as follows:

0 = CLOCK terminal (terminal U12) is an input (default).

1 = CLOCK terminal is an output that utilizes the internal oscillator.

A 43 kΩ pulldown resistor should be tied to the CLOCK pin.



NC – No internal connection

Figure 3–8. TPS2206 Terminal Assignments

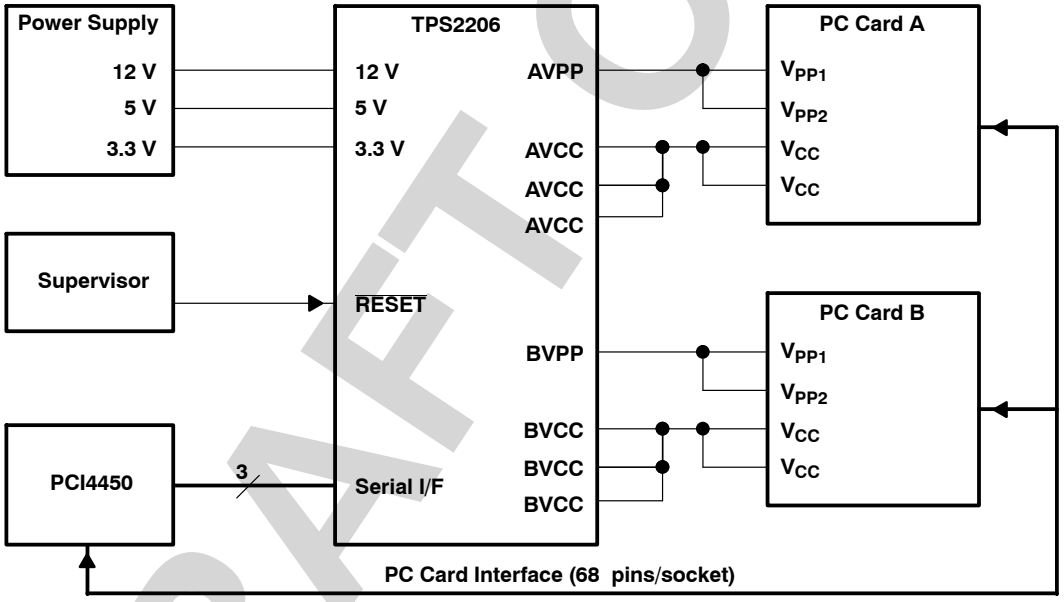
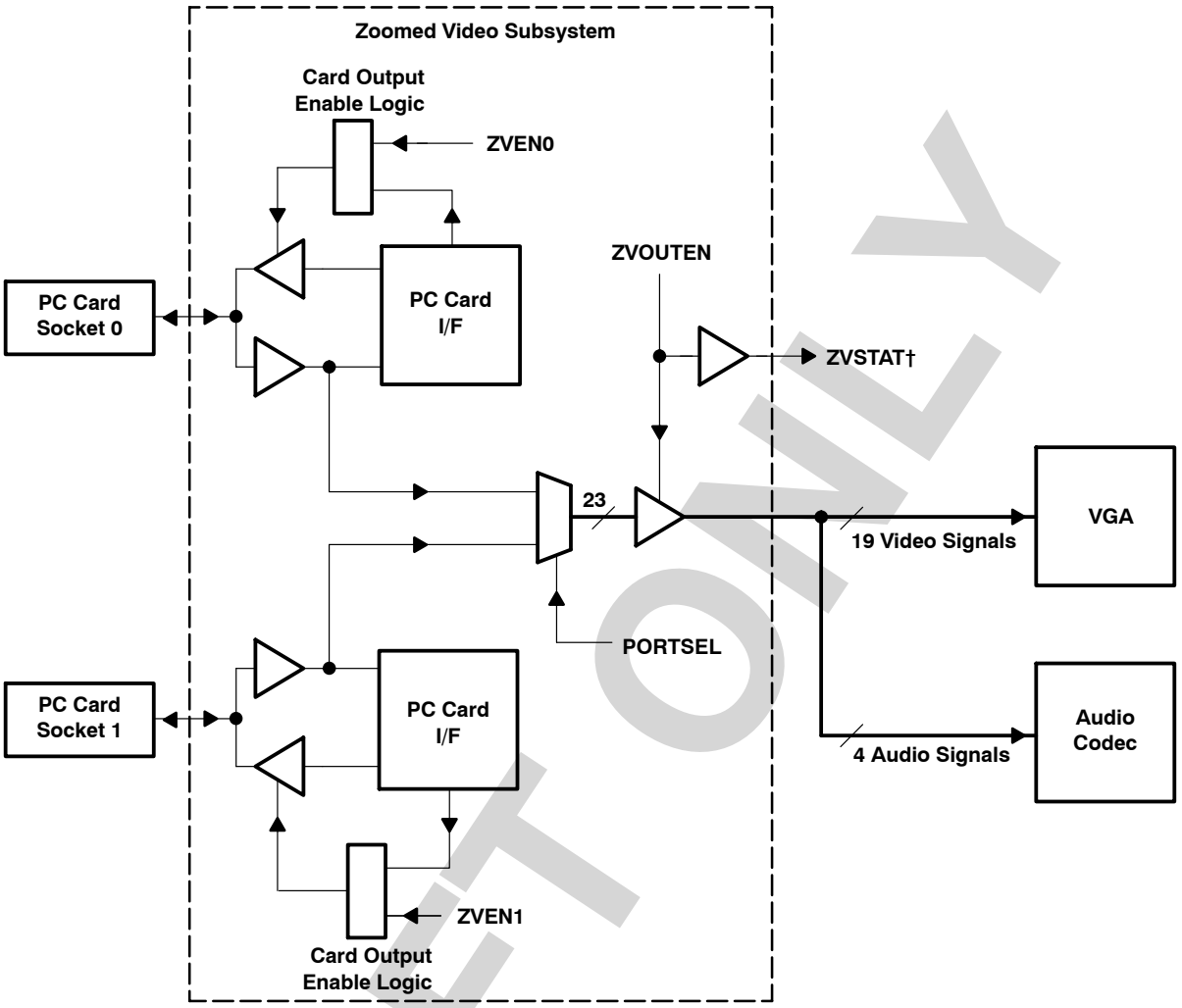


Figure 3–9. TPS2206 Typical Application

3.4.3 Zoomed Video Support

The zoomed video (ZV) port on the PCI4450 provides an internally buffered 16-bit ZV PC Card data path. This internal routing is programmed through the multimedia control register. Figure 3–9 summarizes the zoomed video subsystem implemented in the PCI4450, and details the bit functions found in the multimedia control register.

An output port (PORTSEL) is always selected. The PCI4450 defaults to socket 0 (see the multimedia control register). When ZVOUTEN is enabled, the zoomed video output terminals are enabled and allow the PCI4450 to route the zoomed video data. However, no data is transmitted unless either ZVEN0 or ZVEN1 is enabled in the multimedia control register. If the PORTSEL maps to a card port that is disabled (ZVEN =0 or ZVEN1 = 0), then the zoomed video port is driven low (i.e., no data is transmitted).



† ZVSTAT must be enabled through the GPIO Control Register.

Figure 3–10. Zoomed Video Subsystem

3.4.4 Zoomed Video Auto Detect

Zoomed video auto detect, when enabled, allows the PCI4450 to automatically detect zoomed video data by sensing the pixel clock from each socket and/or from a third zoomed video source that may exist on the motherboard. The PCI4450 automatically switches the internal zoomed video MUX to route the zoomed video stream to the PCI4450's zoomed video output port. This eliminates the need for software to switch the internal MUX using the multimedia control register (PCI offset 84h, bits 6 and 7).

The PCI4450 can be programmed to switch a third zoomed video source by programming MFUNC2 or MFUNC3 as a zoomed video pixel clock sense pin and connecting this pin to the pixel clock of the third zoomed video source. ZVSTAT may then be programmed onto MFUNC4, MFUNC1, or MFUNC0 and this signal may switch the zoomed video buffers from the third zoomed video source. To account for the possibility of several zoomed video sources being enabled at the same time, a programmable priority scheme may be enabled.

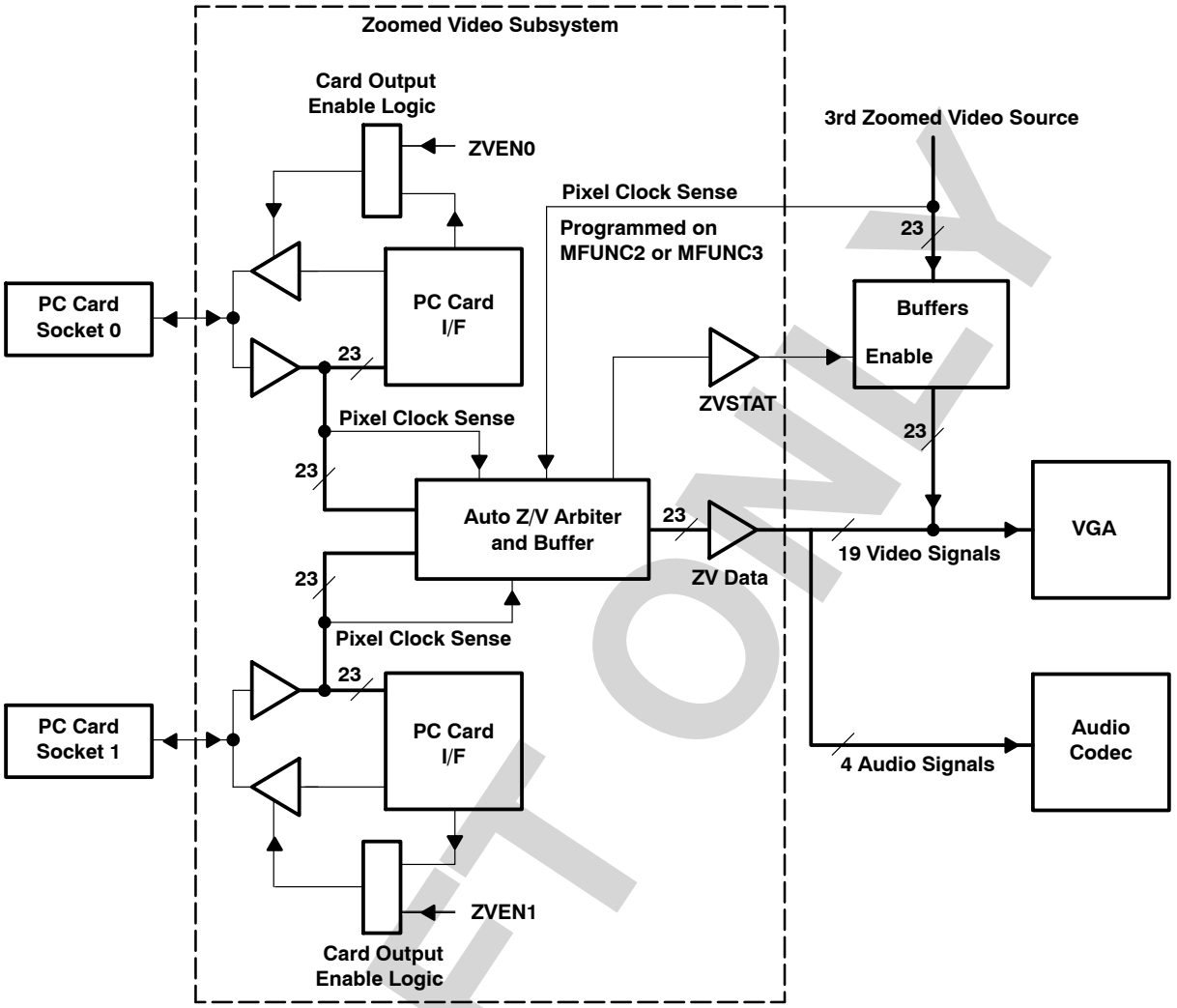


Figure 3–11. Zoomed Video with Auto Detect Enabled

The PCI4450 defaults with zoomed video auto-detect disabled so that it will function exactly like the PCI1250A and PCI1450. To enable zoomed video auto-detect and the programmable priority scheme, the following bits must be set:

- Multimedia control register (PCI offset 84h) bit 5: Writing a 1b enables zoomed video auto-detect
- Multimedia control register (PCI offset 84h) bits 4–2: Set the programmable priority scheme

  000 = Slot A, Slot B, External Source

  001 = Slot A, External Source, Slot B

  010 = Slot B, Slot A, External Source

  011 = Slot B, External Source, Slot A

  100 = External Source, Slot A, Slot B

  101 = External Source, Slot B, Slot A

  110 = External Source, Slot B, Slot A

  111 = Reserved

If it is desired to switch a third zoomed video source, then the following bits must also be set:

- MFUNC routing register (PCI offset 8Ch), bits 14–12 or 10–8: Write 111b to program MFUNC3 or MFUNC2 as a pixel clock input pin.

- MFUNC routing register (PCI ofset 8Ch), bits 18–16, 6–4, or 2–0: Write 111b to program MFUNC4, MFUNC1, or MFUNC0 pin.

3.4.5 Ultra Zoomed Video

Ultra zoomed video is an enhancement to the PCI4450's DMA engine and is intended to improve the 16-bit bandwidth for MPEG I and MPEG II decoder PC Cards. This enhancement allows the 4450 to fetch 32 bits of data from memory versus the 11XX/12XX 16-bit fetch capability. This enhancement allows a higher sustained throughput to the 16-bit PC Card, because the 4450 prefetches an extra 16 bits (32 bits total) during each PCI read transaction. If the PCI Bus becomes busy, then the 4450 has an extra 16 bits of data to perform back-to-back 16-bit transactions to the PC Card before having to fetch more data. This feature is built into the DMA engine and software is not required to enable this enhancement.

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

3.4.6 $\overline{\text{D3\_STAT}}$ Pin

Additional functionality added for the 4450 versus the 1250A/1251 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 and Function 1 are placed in D3
- $\overline{\text{PME}}$ is enabled on either function

The intent of including this feature in the PCI4450 is to use this pin to switch an external V_{CC}/V_{AUX} switch. This feature can be programmed on MFUNC7, MFUNC6, MFUNC2, or MFUNC1 by writing 100b to the appropriate multifunction routing status register bits (PCI offset 8Ch).

3.4.7 Internal Ring Oscillator

The internal ring oscillator provides an internal clock source for the PCI4450 so that neither the PCI clock nor an external clock is required in order for the PCI4450 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 of the system control register (PCI offset 80h) to a 1b. This function is disabled by default.

3.4.8 SPKROUT Usage

The SPKROUT signal carries the digital audio signal from the PC Card to the system. When a 16-bit PC Card is configured for I/O mode, the BVD2 pin becomes $\overline{\text{SPKR}}$. This terminal, also used in CardBus applications, is referred to as CAUDIO. $\overline{\text{SPKR}}$ passes a TTL level digital audio signal to the PCI4450. The CardBus CAUDIO signal also can pass a single amplitude, binary waveform. The binary audio signals from the two PC Card sockets are XOR'ed in the PCI4450 to produce SPKROUT. Figure 3–12 illustrates the SPKROUT connection.

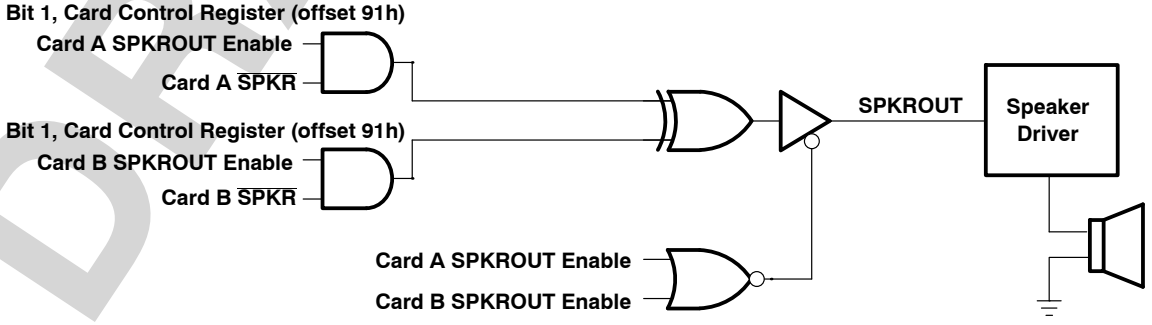


Figure 3–12. SPKROUT Connection to Speaker Driver

The SPKROUT signal is typically driven only by PC modem cards. To verify the SPKROUT on the PCI4450, a sample circuit was constructed, and this simplified schematic is provided below. The PCI1130/1131 required a pullup resistor

on the $\overline{\text{SUSPEND}}$/SPKROUT terminal. Since the PCI4450 does not multiplex any other function on SPKROUT, this terminal does not require a pullup resistor.

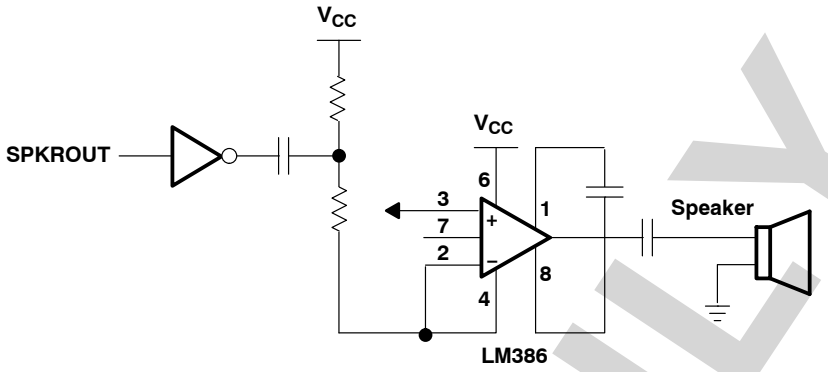


Figure 3–13. Simplified Test Schematic

3.4.9 LED Socket Activity Indicators

The socket activity LEDs indicate when an access is occurring to a PC Card. The LED signals are programmable via the MFUNC register. When configured for LED outputs, these terminals output an active high signal to indicate socket activity. LEDA1 indicates socket 0 (card A) activity, and LEDA2 indicates socket 1 (card B) activity.

The active-high LED signal is driven for 64 ms durations. When the LED is not being driven high, then it is driven to a low state. Either of the two circuits illustrated in Figure 3–14 can be implemented to provide the LED signaling, and it is left for the board designer to implement the circuit to best fit the application.

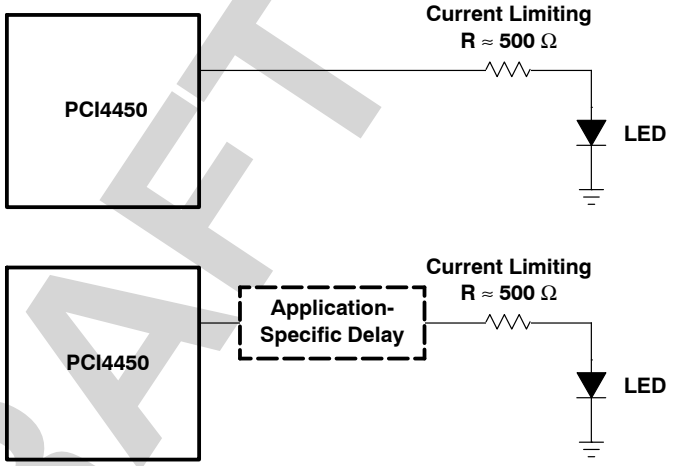


Figure 3–14. Two Sample LED Circuits

As indicated, the LED signals are driven for 64 ms, and this is accomplished by a counter circuit. To avoid the possibility of the LEDs appearing to be stuck when the PCI clock is stopped, the LED signaling is cut off when either the $\overline{\text{SUSPEND}}$ signal is asserted or when the PCI clock is to be stopped per the $\overline{\text{CLKRUN}}$ protocol.

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

3.4.10 PC Card 16 DMA Support

The PCI4450 supports both PC/PCI (centralized) DMA and a distributed DMA slave engine for 16-bit PC Card DMA support. The distributed DMA (DDMA) slave register set provides the programmability necessary for the slave DDMA engine. Table 3–2 provides the DDMA register configuration.

Table 3–2. Distributed DMA Registers

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

3.4.11 CardBus Socket Registers

The PCI4450 contains all registers for compatibility with the latest PCI to PCMCIA CardBus Bridge Specification. These registers exist as the CardBus socket registers, and are listed in Table 3–3.

Table 3–3. 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

3.5 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 PCI4450. The PCI4450 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 upon 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 PCI4450 is therefore backward compatible with existing interrupt control register definitions, and new registers have been defined where required.

The PCI4450 detects PC Card interrupts and events at the PC Card interface and notifies the host controller via one of several interrupt signaling protocols. To simplify the discussion of interrupts in the PCI4450, PC Card interrupts are classified as either card status change (CSC) or as functional interrupts.

The method by which any type of PCI4450 interrupt is communicated to the host interrupt controller varies from system to system. The PCI4450 offers system designers the choice of using parallel PCI interrupt signaling or the serialized ISA and/or PCI interrupt protocol. It is possible to use the parallel PCI interrupts in combination with serialized IRQs via the multifunction routing register at offset 8Ch.

3.5.1 PC Card Functional And Card Status Change Interrupts

PC Card functional interrupts are defined as requests from a PC Card application for interrupt service. They 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, defined as events at the PC Card interface which are detected by the PCI4450, 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 3–4 summarizes the sources of PC Card interrupts and the type of card associated with them. CSC and functional interrupt sources are dependent upon the type of card inserted in the PC Card socket. The three types of cards that may be inserted into any PC Card socket are: 16-bit memory card, 16-bit I/O card, and CardBus cards. 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 3–4. 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 the BVD1 signal indicates a change in the PC Card battery conditions.
		CSC	BVD2 ($\overline{\text{SPKR}}$) // CAUDIO	A transition on the BVD2 signal indicates a change in the PC Card battery conditions.
	Wait states (READY)	CSC	READY ($\overline{\text{IREQ}}$) // $\overline{\text{CINT}}$	A transition on the READY signal 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 the $\overline{\text{STSCHG}}$ signal indicates a status change on the PC Card.
	Interrupt request (IREQ)	Functional	READY ($\overline{\text{IREQ}}$) // $\overline{\text{CINT}}$	The assertion of the $\overline{\text{IREQ}}$ signal indicates an interrupt request from the PC Card.
CardBus	Change in card status (CSTSCHG)	CSC	BVD1 ($\overline{\text{STSCHG}}$) // CSTSCHG	The assertion of the CSTSCHG signal indicates a status change on the PC Card.
	Interrupt request ($\overline{\text{CINT}}$)	Functional	READY ($\overline{\text{IREQ}}$) // $\overline{\text{CINT}}$	The assertion of the $\overline{\text{CINT}}$ signal indicates an interrupt request from the PC Card.
	Power cycle complete	CSC	N/A	An interrupt is generated when a PC Card power-up cycle has completed.
All PC Cards	Card insertion or removal	CSC	CD1 // CCD1, CD2 // CCD2	A transition on either the $\overline{\text{CD1}}$//$\overline{\text{CCD1}}$ signal or the $\overline{\text{CD2}}$//$\overline{\text{CCD2}}$ signal indicates an insertion or removal of a 16-bit // CardBus PC Card.
	Power cycle complete	CSC	N/A	An interrupt is generated when a PC Card power-up cycle has completed.

The signal naming convention for PC Card signals describes the function for 16-bit memory and I/O cards, as well as CardBus. For example, the READY($\overline{\text{IREQ}}$)//$\overline{\text{CINT}}$ signal includes the READY signal for 16-bit memory cards, the $\overline{\text{IREQ}}$ signal for 16-bit I/O cards, and the $\overline{\text{CINT}}$ signal 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 PCI4450 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 PCI4450 interrupt scheme may be used to notify the host system, as in indicated in Table 3–4, denoted by the power cycle complete event. This interrupt source is considered a PCI4450 internal event because it does not depend on a signal change at the PC Card interface, but rather the completion of applying power to the socket.

3.5.2 Interrupt Masks And Flags

Host software may individually mask, or disable, most of the potential interrupt sources listed in Table 3–5 by setting the appropriate bits in the PCI4450. By individually masking the interrupt sources listed in these tables, software can control which events will cause a PCI4450 interrupt. Host software has some control over which system interrupt the PCI4450 will assert by programming the appropriate routing registers. The PCI4450 allows host software to route PC Card CSC and PC Card functional interrupts to separate system interrupts. Interrupt routing is somewhat specific to the interrupt signaling method used. This will be discussed in more detail in the following sections.

When an interrupt is signaled by the PCI4450, the interrupt service routine must be able to discern which of the events in Table 3–5 caused the interrupt. Internal registers in the PCI4450 provide flags which report which of the interrupt sources was the cause of an interrupt. By reading these status bits, the interrupt service routine can determine which action is to be taken.

Table 3–5 details the registers and bits associated with masking and reporting potential interrupts. All interrupts may be masked except the functional PC Card interrupts, and an interrupt status flag is available for all types of interrupts.

Table 3–5. PCI4450 Interrupt Masks and Flags Registers

Card Type	Event	Mask	Flag
16-bit Memory	Battery conditions (BVD1, BVD2)	ExCA Offset 05h/45h/805h Bits 1 & 0	ExCA Offset 04h/44h/804h Bits 1 & 0
	Wait states (READY)	ExCA Offset 05h/45h/805h Bit 2	ExCA Offset 04h/44h/804h Bit 2
16-bit I/O	Change in card status ($\overline{\text{STSCHG}}$)	ExCA Offset 05h/45h/805h Bit 0	ExCA Offset 04h/44h/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/45h/805h Bit 3	ExCA Offset 04h/44h/804h Bit 3
CardBus	Change in card status (CSTSCHG)	Socket mask register Bit 0	Socket event register Bit 0
	Interrupt request ($\overline{\text{CINT}}$)	Always enabled	PCI Configuration Offset 91h Bit 0
	Power cycle complete	Socket mask register Bit 3	Socket event register Bit 3
	Card insertion or removal	Socket mask register Bits 2 & 1	Socket event register Bits 2 & 1

There is no mask bit to stop the PCI4450 from passing PC Card functional interrupts through to the appropriate interrupt scheme. Functional interrupts should not be fired until the PC Card is initialized and powered.

There are various methods of clearing the interrupt flag bits listed in Table 3–5. The flag bits in the ExCA registers (16-bit PC Card related interrupt flags) may be cleared by two different methods. One method is an explicit write of 1 to the flag bit to clear, and the other is a reading of the flag bit register. The selection of flag bit clearing is made by bit 2 in the global control register (ExCA offset 1Eh/5Eh/81Eh), and defaults to the flag cleared on read method.

The CardBus related interrupt flags can only be cleared by an explicit write of 1 to the interrupt flag in the socket event register. 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.

3.5.3 Using Parallel PCI Interrupts

Parallel PCI interrupts are available when in pure parallel PCI interrupt mode and are routed on MFUNC0–MFUNC2. The PCI interrupt signaling is dependent upon the interrupt mode and is summarized in Table 3–6. The interrupt mode is selected in the device control register (92h).

Table 3–6. Interrupt Pin Register Cross Reference

Interrupt Signaling Mode	INTPIN Function 0	INTPIN Function 1
Parallel PCI interrupts only	0x01 ($\overline{\text{INTA}}$)	0x02 ($\overline{\text{INTB}}$)
Reserved	0x01 ($\overline{\text{INTA}}$)	0x02 ($\overline{\text{INTB}}$)
IRQ serialized (IRQSER) & parallel PCI interrupts	0x01 ($\overline{\text{INTA}}$)	0x01 ($\overline{\text{INTA}}$)
IRQ & PCI serialized (IRQSER) interrupts (default)	0x01 ($\overline{\text{INTA}}$)	0x02 ($\overline{\text{INTB}}$)

3.6 Power Management Overview

In addition to the low-power CMOS technology process used for the PCI4450, various features are designed into the device to allow implementation of popular power saving techniques. These features and techniques are discussed in this section.

3.6.1 $\overline{\text{CLKRUN}}$ Protocol

$\overline{\text{CLKRUN}}$ is the primary method of power management on the PCI bus side of the PCI4450. Since some chipsets do not implement $\overline{\text{CLKRUN}}$, this is not always available to the system designer, and alternate power savings features are provided.

If $\overline{\text{CLKRUN}}$ is not implemented, then the $\overline{\text{CLKRUN}}$ pin should be tied low. $\overline{\text{CLKRUN}}$ is enabled by default via bit 1 (KEEPCLK) in the system control register (80h).

3.6.2 CardBus PC Card Power Management

The PCI4450 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 CCLK can also be configured as divide by 16 instead of stopped. The $\overline{\text{CLKRUN}}$ protocol is followed on the CardBus interface to control this clock management.

3.6.3 PCI Bus Power Management

The *PCI Bus Power Management Interface Specification* (PCIPM) establishes the infrastructure required to let the operating system control the power of PCI functions. This is done by defining a standard PCI interface and operations to manage the power of PCI functions on the bus. The PCI bus and the PCI functions can be assigned one of four software visible power management states, which 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, and 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 upstream bridge device.

For the operating system to manage the device power states on the PCI bus, the PCI function should support four power management operations. The four operations are: capabilities reporting; power status reporting; setting the power state; and system wake-up. The operating system identifies the capabilities of the PCI function by traversing the new capabilities list. The presence of new capabilities is indicated by a 1b in bit 4 of the PCI status register (PCI offset 06h). When software determines that the device has a capabilities list by seeing that bit 4 of the PCI status register is set, it will read the capability pointer register at PCI offset 14h. This value in the register points the location in PCI configuration space of the capabilities linked list.

The first byte of each capability register block is required to be a unique ID of that capability. PCI power management has been assigned an ID of 01h. The next byte is a pointer to the next pointer item in the list of capabilities. If there are no more items in the list, then the next item pointer should be set to 0. The registers following the next item pointer are specific to the function's capability. The PCIPM capability implements the following register block:

Power Management Register Block

Power management capabilities (PMC)		Next item pointer	Capability ID	Offset = 0
Data	PMCSR bridge support extensions	Power management control status (CSR)		Offset = 4

The power management capabilities (PMC) register is a static read-only register that provides information on the capabilities of the function, related to power management. The PMCSR register enables control of power management states and enables/monitors power management events. The data register is an optional register that provides a mechanism for state-dependent power measurements such as power consumed or heat dissipation.

3.6.4 CardBus Device Class Power Management

The *PCI Bus 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 Interface Specification for PCI-to-CardBus Bridges* is wake-up from $D3_{hot}$ or $D3_{cold}$ without losing wake-up context (also called $\overline{PME}$ context).

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

- Preservation of device context: The *PCI Power Management Specification* version 1.0 states that $\overline{PRST}$ must be asserted when transitioning from $D3_{cold}$ to D0. Some method to preserve wake-up context must be implemented so that $\overline{PRST}$ does not clear the $\overline{PME}$ context registers.
- Power source in $D3_{cold}$ if wake-up support is required from this state.

The Texas Instruments PCI4450 addresses these D3 wake-up issues in the following manner:

- Preservation of device context: When $\overline{PRST}$ is asserted, bits required to preserve $\overline{PME}$ context are not cleared. To clear all bits in the PCI4450, another reset pin is defined: $\overline{G\_RST}$ (global reset). $\overline{G\_RST}$ is normally only asserted during the initial power-on sequence. After the initial boot, $\overline{PRST}$ should be asserted so that $\overline{PME}$ context is retained for D3-to-D0 transitions. Bits cleared by $\overline{G\_RST}$, but not cleared by $\overline{PRST}$ (if the $\overline{PME}$ enable bit is set), are referred to as $\overline{PME}$ context bits. Please refer to the master list of $\overline{PME}$ context bits in the next section.
- Power source in $D3_{cold}$ if wake-up support is required from this state. Since V_{CC} is removed in $D3_{cold}$, an auxiliary power source must be switched to the PCI4450 V_{CC} pins. This switch should be a *make before break* type of switch, so that V_{CC} to the PCI4450 is not interrupted.

3.6.5 Master List Of $\overline{PME}$ Context Bits And Global Reset Only Bits

$\overline{PME}$ context bit means that the bit is cleared only by the assertion of $\overline{G\_RST}$ when the $\overline{PME}$ enable bit is set (PCI offset A4h, bit 8). If $\overline{PME}$ is not enabled, then these bits are cleared when either $\overline{PRST}$ or $\overline{G\_RST}$ is asserted.

Global reset only bits, as the name implies, are only cleared by $\overline{G\_RST}$. These bits are never cleared by $\overline{PRST}$ regardless of the setting of the $\overline{PME}$ enable bit. (PCI offset A4h, bit 8). The $\overline{G\_RST}$ signal is gated only by the $\overline{SUSPEND}$ signal. This means that assertion of $\overline{SUSPEND}$ blocks the $\overline{G\_RST}$ signal internally, thus preserving all register contents.

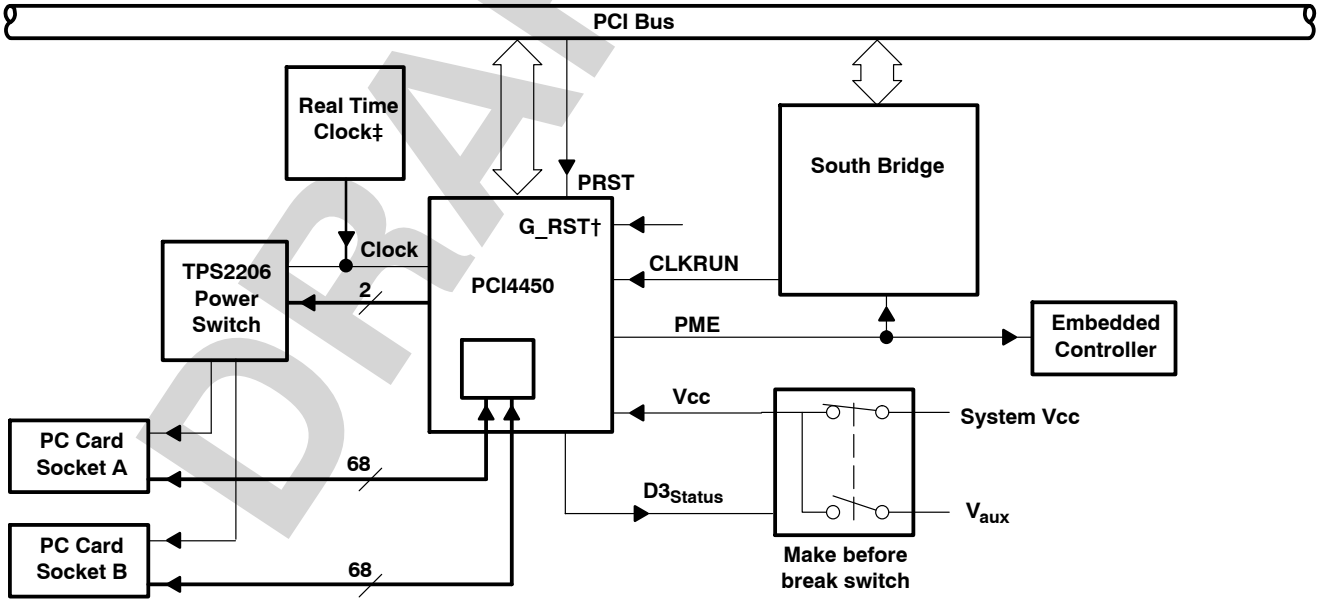
Global reset only bits:

- 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 31–29, 27–24, 22–14, 6–3, 1, 0
- Multimedia control register (PCI offset 84h): bits 7–0
- General status register (PCI offset 85h): bits 2–0
- General-purpose event status register (PCI offset 88h): bits 7, 6, 3–0
- General-purpose event enable register (PCI offset 89h): bits 7, 6, 3–0
- General-purpose input register (PCI offset 8Ah): bits 3–0
- General-purpose output register (PCI offset 8Bh): bits 3–0
- MFUNC routing register (PCI offset 8Ch): bits 31–0
- Retry status register (PCI offset 90h): bits 7–1
- Card control register (PCI offset 91h): bits 7, 6, 2, 1, 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 15–0
- $\overline{\text{GPE}}$ control/status register (PCI offset A8h): bits 10, 9, 8, 2, 1, 0

$\overline{\text{PME}}$ context bits

- Bridge control register (PCI offset 3Eh): bit 6
- Power management capabilities register (PCI offset A2h): bit 15
- Power management control/status register (PCI offset A4h): bits 15, 8
- ExCA power control register (ExCA 802h/842h): bits 7, 4, 3, 1, 0
- ExCA interrupt and general control (ExCA 803h/843h): bit 6, 5
- ExCA card status change register (ExCA 804h/844h): bits 3–0
- ExCA card status change interrupt register (ExCA 805h/845h): 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 control register (CardBus offset 10h): bits 6, 5, 4, 2, 1, 0

System Diagram Implementing CardBus Device Class Power Management



† The system connection to $\overline{\text{GRST}}$ is implementation specific. $\overline{\text{GRST}}$ should be applied whenever V_{cc} is applied to the PCI4450. $\overline{\text{PRST}}$ should be applied for subsequent warm resets.

‡ Not required if internal oscillator is used.

Figure 3–15. System Diagram Implementing CardBus Device Class Power Management

3.6.6 Suspend Mode

The $\overline{\text{SUSPEND}}$ signal, provided for backward compatibility, gates the $\overline{\text{PRST}}$ (PCI reset) signal and the $\overline{\text{G\_RST}}$ (global reset) signal from the PCI4450. Besides gating $\overline{\text{PRST}}$ and $\overline{\text{G\_RST}}$, $\overline{\text{SUSPEND}}$ also gates PCLK inside the PCI4450 in order to minimize power consumption.

Gating PCLK does not create any issues with respect to the power switch interface in the PCI4450. This is because the PCI4450 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 PCI4450:

- Use an external clock to the PCI4450 CLOCK pin
- Use the internal oscillator

It should also 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, then the PCI clock will have to be restarted in order to pass the interrupt, because neither the internal oscillator nor an external clock is routed to the serial interrupt state machine.

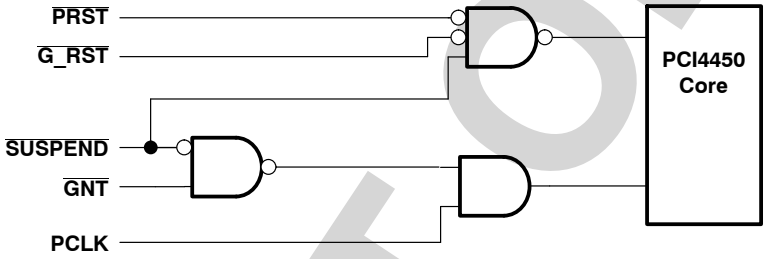


Figure 3–16. $\overline{\text{SUSPEND}}$ Functional Illustration

3.6.7 Requirements For $\overline{\text{SUSPEND}}$

A requirement for implementing suspend mode is that the PCI bus must not be parked on the PCI4450 when $\overline{\text{SUSPEND}}$ is asserted. The PCI4450 responds to $\overline{\text{SUSPEND}}$ being asserted by placing the $\overline{\text{REQ}}$ pin in a high impedance state. The PCI4450 will also gate the internal clock and reset.

The GPIOs, MFUNC signals, and $\overline{\text{RI\_OUT}}$ signals are all active during $\overline{\text{SUSPEND}}$, unless they are disabled in the appropriate PCI4450 registers.

3.6.8 Ring Indicate

The $\overline{\text{RI\_OUT}}$ output is an important feature used in legacy power management. It is used so that a system can go into a suspended mode and wake up on modem rings and other card events. The $\overline{\text{RI\_OUT}}$ signal on the PCI4450 may be asserted under any of the following conditions:

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

A CSTSCHG signal from a powered CardBus card is indicated as a CSC event, not as a CBWAKE event. These two $\overline{\text{RI\_OUT}}$ events are enabled separately. The following figure details various enable bits for the PCI4450 $\overline{\text{RI\_OUT}}$ function; however, it does not illustrate the masking of CSC events. See *interrupt masks and flags* for a detailed description of CSC interrupt masks and flags.

$\overline{\text{RI\_OUT}}$ is multiplexed on the same pin with $\overline{\text{PME}}$. The default is for $\overline{\text{RI\_OUT}}$ to be signaled on this pin. In PCI power managed systems, the $\overline{\text{PME}}$ signal should be enabled by setting bit 0 ($\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$) in the system control register (80h) and clearing bit 7 (RIENB) in the card control register (91h).

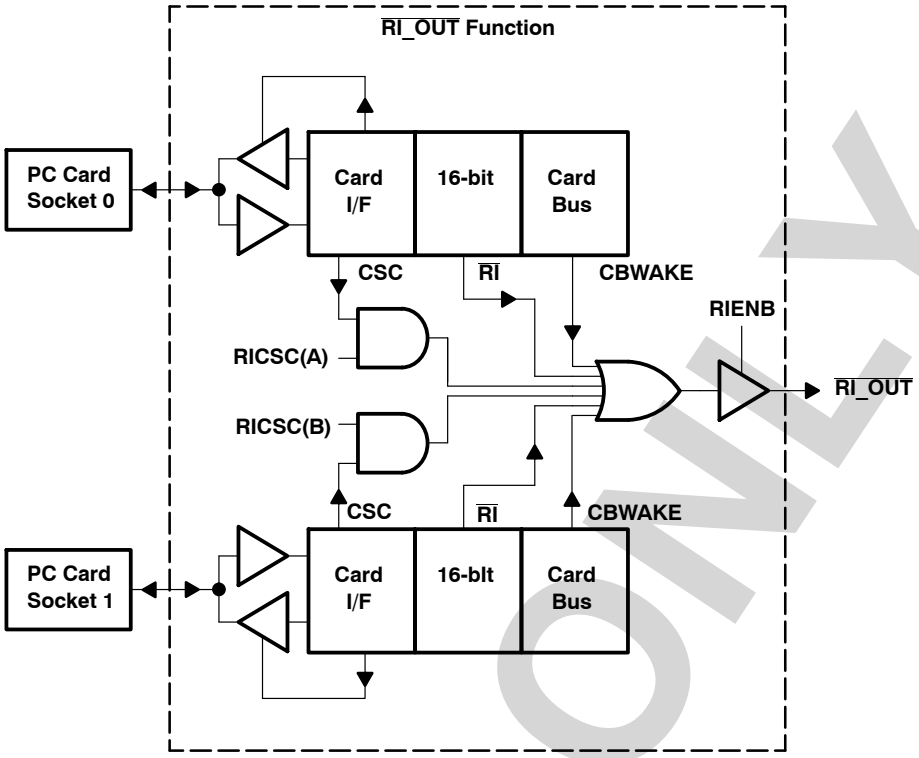


Figure 3–17. $\overline{\text{RI\_OUT}}$ Functional Illustration

Routing of CSC events to the $\overline{\text{RI\_OUT}}$ signal, enabled on a per socket basis, is programmed by the RICSC bit in the card control register. This bit is socket dependent (not shared), as illustrated in Figure 3–17.

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

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

4 PC Card Controller Programming Model

This chapter describes the PCI4450 PCI configuration registers that make up the 256-byte PCI configuration header for each PCI4450 function. As noted below, some bits are global in nature and should be accessed only through function 0.

Registers containing one or more global bits are denoted by a "§."

Any bit followed by a "†" is not cleared by the assertion of $\overline{\text{PRST}}$ (refer to *CardBus device class power management* for more details) if $\overline{\text{PME}}$ is enabled (PCI offset A4h, bit 8). In this case, these bits are only cleared by $\overline{\text{GRST}}$. If $\overline{\text{PME}}$ is not enabled, then these bits are cleared by $\overline{\text{GRST}}$ or $\overline{\text{PRST}}$. These bits are sometimes referred to as PME context bits and are implemented to allow $\overline{\text{PME}}$ context to be preserved when transitioning from $D3_{hot}$ or $D3_{cold}$ to D0. If the PME context $\overline{\text{PRST}}$ functionality is not desired, then the $\overline{\text{PRST}}$ and $\overline{\text{GRST}}$ signals should be tied together.

If a bit is followed by a "‡", then this bit is only cleared by $\overline{\text{GRST}}$ in all cases (not conditional on $\overline{\text{PME}}$ being enabled). These bits are intended to maintain device context such as interrupt routing and MFUNC programming during "warm" resets.

4.1 PCI Configuration Registers (Functions 0 and 1)

The PCI4450 is a multifunction PCI device, and the PC Card controller is integrated as PCI functions 0 and 1. The configuration header, compliant with the PCI Specification as a CardBus bridge header, is PC97/PC98 compliant as well. Table 4–1 illustrates the PCI configuration header, which includes both the predefined portion of the configuration space and the user definable registers.

Table 4–1. Functions 0 and 1 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
CardBus socket registers/ExCA base address				10h
Secondary status		Reserved	Capability pointer	14h
CardBus latency timer	Subordinate bus number	CardBus bus number	PCI bus number	18h
CardBus memory base register 0				1Ch
CardBus memory limit register 0				20h
CardBus memory base register 1				24h
CardBus memory limit register 1				28h
CardBus I/O base register 0				2Ch
CardBus I/O limit register 0				30h
CardBus I/O base register 1				34h
CardBus I/O limit register 1				38h
Bridge control †		Interrupt pin	Interrupt line	3Ch
Subsystem ID ‡		Subsystem vendor ID ‡		40h
PC Card 16-bit I/F legacy mode base address ‡				44h
Reserved				48h–7Fh
System control ‡				80h
Reserved	Reserved	General status †	Multimedia control ‡	84h
GP0 control ‡	GP1 control ‡	GPE enable ‡	GPE status ‡	88h
Multifunction routing †				8Ch

Diagnostic ‡	Device control ‡	Card control ‡	Retry status ‡	90h
Socket DMA register 0 ‡				94h
Socket DMA register 1 ‡				98h
Reserved				9Ch
Power management capabilities †		Next pointer item	Capability ID	A0h
Data (Reserved)	PMCSR bridge support extensions	Power management control/status †		A4h
Reserved		$\overline{\text{GPE}}$ control/status ‡		A8h

† One or more bits in the register are PME context bits and can only be cleared by the assertion of $\overline{\text{GRST}}$ when $\overline{\text{PME}}$ is enabled. If $\overline{\text{PME}}$ is not enabled, then these bits are cleared by the assertion of $\overline{\text{PRST}}$ or $\overline{\text{GRST}}$.

‡ One or more bits in this register are only cleared by the assertion $\overline{\text{GRST}}$.

4.2 Vendor ID Register

The vendor ID register contains a value allocated by the PCI SIG that 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 (Functions 0, 1)
Default: 104Ch

4.3 Device ID Register

The device ID register contains a value assigned to the PCI4450 by Texas Instruments. The device identification for the PCI4450 is AC40.

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	0

Register: **Device ID**
Type: Read-only
Offset: 02h (Functions 0, 1)
Default: AC40h

4.4 Command Register

The command register provides control over the PCI4450 interface to the PCI bus. All bit functions adhere to the definitions in the PCI Local Bus Specification, see Table 4–2. None of the bit functions in this register are shared between the two PCI4450 PCI functions. Two command registers exist in the PCI4450, one for each function. Software manipulates the two PCI4450 functions as separate entities when enabling functionality through the command register. The SERR_EN and PERR_EN enable bits in this register are internally wired OR between the two functions, and these control bits appear separate per function to software.

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

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

Table 4–2. PCI Command Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–10	RSVD	R	Bits 15–10 return 0s when read.
9	??	R	Fast back-to-back enable. The PCI4450 will not generate fast back-to-back transactions; therefore, this bit is read-only. This bit returns a 0 when read.
8	??	R/W	System error ($\overline{\text{SERR}}$) enable. This bit controls the enable for the $\overline{\text{SERR}}$ driver on the PCI interface. $\overline{\text{SERR}}$ can be asserted after detecting an address parity error on the PCI bus. Both this bit and bit 6 must be set for the PCI4450 to report address parity errors. 0 = Disables the $\overline{\text{SERR}}$ output driver (default). 1 = Enables the $\overline{\text{SERR}}$ output driver.
7	??	R	Address/data stepping control. The PCI4450 does not support address/data stepping, and this bit is hardwired to 0. Writes to this bit have no effect.
6	??	R/W	Parity error response enable. This bit controls the PCI4450's response to parity errors through the $\overline{\text{PERR}}$ signal. Data parity errors are indicated by asserting $\overline{\text{PERR}}$, while address parity errors are indicated by asserting $\overline{\text{SERR}}$. 0 = PCI4450 ignores detected parity error (default). 1 = PCI4450 responds to detected parity errors.
5	??	R/W	VGA palette snoop. When set to 1, palette snooping is enabled (i.e., the PCI4450does not respond to palette register writes and snoops the data). When the bit is 0, the PCI4450 will treat all palette accesses like all other accesses.
4	??	R	Memory write and invalidate enable. This bit controls whether a PCI initiator device can generate memory write and invalidate commands. The PCI4450 controller does not support memory write and invalidate commands, it uses memory write commands instead; therefore, this bit is hardwired to 0. This bit returns 0 when read. Writes to this bit have no effect.
3	??	R	Special cycles. This bit controls whether or not a PCI device ignores PCI special cycles. The PCI4450 does not respond to special cycle operations; therefore, this bit is hardwired to 0. This bit returns 0 when read. Writes to this bit have no effect.
2	??	R/W	Bus master control. This bit controls whether or not the PCI4450 can act as a PCI bus initiator (master). The PCI4450 can take control of the PCI bus only when this bit is set. 0 = Disables the PCI4450's ability to generate PCI bus accesses (default). 1 = Enables the PCI4450's ability to generate PCI bus accesses.
1	??	R/W	Memory space enable. This bit controls whether or not the PCI4450 may claim cycles in PCI memory space. 0 = Disables the PCI4450's response to memory space accesses (default). 1 = Enables the PCI4450's response to memory space accesses.
0	??	R/W	I/O space control. This bit controls whether or not the PCI4450 may claim cycles in PCI I/O space. 0 = Disables the PCI4450 from responding to I/O space accesses (default). 1 = Enables the PCI4450 to respond to I/O space accesses.

4.5 Status Register

The status register provides device information to the host system. Bits in this register may be read normally. A bit in the status register is reset when a 1 is written to that bit location; a 0 written to a bit location has no effect. All bit functions adhere to the definitions in the *PCI Bus Specification*, as seen in the bit descriptions. PCI bus status is shown through each function.

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

Register: **Status**
Type: Read-only, Read/Write
Offset: 06h (Functions 0, 1)
Default: 0210h

Table 4–3. Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	PAR_ERR	R/W	Detected parity error. This bit is set when a parity error is detected, either address or data parity errors. Write a 1 to clear this bit.
14	SYS_ERR	R/W	Signaled system error. This bit is set when $\overline{\text{SERR}}$ is enabled and the PCI4450 signaled a system error to the host. Write a 1 to clear this bit.
13	MABORT	R/W	Received master abort. This bit is set when a cycle initiated by the PCI4450 on the PCI bus has been terminated by a master abort. Write a 1 to clear this bit.
12	TABT_REC	R/W	Received target abort. This bit is set when a cycle initiated by the PCI4450 on the PCI bus was terminated by a target abort. Write a 1 to clear this bit.
11	TABT_SIG	R/W	Signaled target abort. This bit is set by the PCI4450 when it terminates a transaction on the PCI bus with a target abort. Write a 1 to clear this bit.
10–9	PCI_SPEED	R	DEVSEL timing. These bits encode the timing of $\overline{\text{DEVSEL}}$ and are hardwired to 01b indicating that the PCI4450 asserts this signal at a medium speed on nonconfiguration cycle accesses.
8	DATAPAR	R/W	Data parity error detected. Write a 1 to clear this bit. 0 = The conditions for setting this bit 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 PCI4450. b. The PCI4450 was the bus master during the data parity error. c. The parity error response bit is set in the command register.
7	FBB_CAP	R	Fast back-to-back capable. The PCI4450 cannot accept fast back-to-back transactions; thus, this bit is hardwired to 0.
6	UDF	R	UDF supported. The PCI4450 does not support the user definable features; therefore, this bit is hardwired to 0.
5	??	R	66 MHz capable. The PCI4450 operates at a maximum PCLK frequency of 33 MHz; therefore, this bit is hardwired to 0.
4	??	R	Capabilities list. This bit 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	These bits return 0s when read.

4.6 Class Code And Revision ID Register

The class code and revision ID register recognizes the PCI4450 functions 0 and 1 as a bridge device (06h) and CardBus bridge device (07h) with a 00h programming interface. Furthermore, the TI chip revision is indicated in the lower byte (00h).

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	0	1	1	0	0	0	0	0	0	1	1	1
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	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Class code and revision ID**
Type: Read-only
Offset: 08h (Functions 0, 1)
Default: 0607 0000h

4.7 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 (Functions 0, 1)
Default: 00h

4.8 Latency Timer Register

The latency timer register specifies the latency timer for the PCI4450, in units of PCI clock cycles. When the PCI4450 is a PCI bus initiator and asserts $\overline{\text{FRAME}}$, the latency timer begins counting from zero. If the latency timer expires before the PCI4450 transaction has terminated, then the PCI4450 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

4.9 Header Type Register

The header type register returns 82h when read, indicating that the PCI4450 functions 0 and 1 configuration spaces adhere to the CardBus bridge PCI header. The CardBus bridge PCI header ranges from PCI register 0 to 7Fh, and 80h–FFh is 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 (Functions 0, 1)
Default: 82h

4.10 BIST Register

Since the PCI4450 does not support a built-in self-test (BIST), this register returns the value of 00h when read. This register returns 0s for the two PCI4450 functions.

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 (Functions 0, 1)
Default: 00h

4.11 CardBus Socket Registers / ExCA Registers Base Address Register

This 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 ones to this register, the value read back will be FFFF F000h, indicating that at least 4K bytes of memory address space are required. The CardBus registers start at offset 000h, and the memory mapped ExCA registers begin at offset 800h. This register is not shared by functions 0 and 1, mapping each socket control register separately.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CardBus socket registers/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 registers/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 registers/ExCA base address**
Type: Read-only, Read/Write
Offset: 10h
Default: 0000 0000h

4.12 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. Each socket has its own capability pointer register. This register is read-only and 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

4.13 Secondary Status Register

The secondary status register is compatible with the PCI-PCI bridge secondary status register. It indicates CardBus related device information to the host system. This register is very similar to the PCI status register (offset 06h), and status bits are cleared by a writing a 1. This register is not shared by the two socket functions, but is accessed on a per socket basis.

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

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

Table 4–4. Secondary Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	CBPARITY	R/WC	Detected parity error. This bit is set when a CardBus parity error is detected, either address or data parity errors. Write a 1 to clear this bit.
14	CBSERR	R/WC	Signaled system error. This bit is set when $\overline{\text{CSERR}}$ is signaled by a CardBus card. The PCI4450 does not assert the $\overline{\text{CSERR}}$ signal. Write a 1 to clear this bit.
13	CBMABORT	R/WC	Received master abort. This bit is set when a cycle initiated by the PCI4450 on the CardBus bus has been terminated by a master abort. Write a 1 to clear this bit.
12	REC_CBTA	R/WC	Received target abort. This bit is set when a cycle initiated by the PCI4450 on the CardBus bus was terminated by a target abort. Write a 1 to clear this bit.
11	SIG_CBTA	R/WC	Signaled target abort. This bit is set by the PCI4450 when it terminates a transaction on the CardBus bus with a target abort. Write a 1 to clear this bit.
10–9	CB_SPEED	R	CDEVSEL timing. These bits encode the timing of $\overline{\text{CDEVSEL}}$ and are hardwired to 01b indicating that the PCI4450 asserts this signal at a medium speed.
8	CB_DPAR	R/WC	CardBus data parity error detected. Write a 1 to clear this bit. 0 = The conditions for setting this bit have not been met. 1 = A data parity error occurred and the following conditions were met: a. $\overline{\text{CPERR}}$ was asserted on the CardBus interface. b. The PCI4450 was the bus master during the data parity error. c. The parity error response bit is set in the bridge control register.
7	CBFBB_CAP	R	Fast back-to-back capable. The PCI4450 cannot accept fast back-to-back transactions; therefore, this bit is hardwired to 0.
6	CB_UDF	R	User definable feature support. The PCI4450 does not support the user definable features; therefore, this bit is hardwired to 0.
5	CB66MHZ	R	66 MHz capable. The PCI4450 CardBus interface operates at a maximum CCLK frequency of 33 MHz; therefore, this bit is hardwired to 0.
4–0	RSVD	R	These bits return 0s when read.

4.14 PCI Bus Number Register

The PCI bus number register is programmed by the host system to indicate the bus number of the PCI bus to which the PCI4450 is connected. The PCI4450 uses this register, in conjunction with the CardBus bus number and subordinate bus number registers, to determine when to forward PCI configuration cycles to its secondary buses.

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

Register: **PCI bus number**
Type: Read/Write
Offset: 18h (Functions 0, 1)
Default: 00h

4.15 CardBus Bus Number Register

The CardBus bus number register is programmed by the host system to indicate the bus number of the CardBus bus to which the PCI4450 is connected. The PCI4450 uses this register, in conjunction with the PCI bus number and subordinate bus number registers, to determine when to forward PCI configuration cycles to its secondary buses. This register is separate for each PCI4450 controller function.

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

4.16 Subordinate Bus Number Register

The subordinate bus number register is programmed by the host system to indicate the highest numbered bus below the CardBus bus. The PCI4450 uses this register, in conjunction with the PCI bus number and CardBus bus number registers, to determine when to forward PCI configuration cycles to its secondary buses. This register is separate for each CardBus controller function.

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

4.17 CardBus Latency Timer Register

The CardBus latency timer register is programmed by the host system to specify the latency timer for the PCI4450 CardBus interface, in units of CCLK cycles. When the PCI4450 is a CardBus initiator and asserts $\overline{\text{CFRAME}}$, the CardBus latency timer begins counting. If the latency timer expires before the PCI4450 transaction has terminated, then the PCI4450 terminates the transaction at the end of the next data phase. A recommended minimum value for this register of 20h 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 (Functions 0, 1)
Default: 00h

4.18 Memory Base Registers 0, 1

These registers indicate the lower address of a PCI memory address range. They are used by the PCI4450 to determine when to forward a memory transaction to the CardBus bus, and likewise, 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. Writes to these bits have no effect. Bits 8 and 9 of the bridge control register specify whether memory windows 0 and 1 are prefetchable or nonprefetchable. The memory base register or the memory limit register must be nonzero in order for the PCI4450 to claim any memory transactions through CardBus memory windows (i.e., these windows are not enabled by default to pass the first 4K bytes of memory to CardBus).

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

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

4.19 Memory Limit Registers 0, 1

These registers indicate the upper address of a PCI memory address range. They are used by the PCI4450 to determine when to forward a memory transaction to the CardBus bus, and likewise, 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. Writes to these bits have no effect. Bits 8 and 9 of the bridge control register specify whether memory windows 0 and 1 are prefetchable or nonprefetchable. The memory base register or the memory limit register must be nonzero in order for the PCI4450 to claim any memory transactions through CardBus memory windows (i.e., these windows are not enabled by default to pass the first 4K bytes of memory to CardBus).

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

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

4.20 I/O Base Registers 0, 1

These registers indicate the lower address of a PCI I/O address range. They are used by the PCI4450 to determine when to forward an I/O transaction to the CardBus bus, and likewise, 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 all 0s which locate this 64-Kbyte page in the first page of the 32-bit PCI I/O address space. Bits 31–16 and bits 1–0 are read-only and always return 0s, forcing I/O windows to be aligned on a natural doubleword boundary in the first 64-Kbyte page of PCI I/O address space. These I/O windows are enabled when either the I/O base register or the I/O limit register are nonzero. The I/O windows are not enabled by default to pass the first doubleword of I/O to CardBus.

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	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 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

4.21 I/O Limit Registers 0, 1

These registers indicate the upper address of a PCI I/O address range. They are used by the PCI4450 to determine when to forward an I/O transaction to the CardBus bus, and likewise, 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 which 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 1–0 are read-only and always return 0s, forcing I/O windows to be aligned on a natural doubleword boundary. Writes to read-only bits have no effect. The PCI4450 assumes that the lower two bits of the limit address are ones.

These I/O windows are enabled when either the I/O base register or the I/O limit register are nonzero. The I/O windows are not enabled by default to pass the first doubleword of I/O to CardBus.

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 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

4.22 Interrupt Line Register

The interrupt line register communicates interrupt line routing information to the host system. This register is not used by the PCI4450, since there are many programmable interrupt signaling options. This register is considered reserved; however, host software may read and write to this register. Each PCI4450 function has an interrupt line register.

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

4.23 Interrupt Pin Register

The value read from this register is function dependent. The value depends on two interrupt tie bits (INTRTIE and TIEALL) in the system control register. INTRTIE is compatible with other TI CardBus controllers and ties $\overline{\text{INTA}}$ to $\overline{\text{INTB}}$ internally. The TIEALL bit ties $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, and $\overline{\text{INTC}}$ together internally. The internal interrupt connections set by INTRTIE and TIEALL are communicated to host software through this standard register interface. Refer to Table 4–5 for a complete description of the register contents.

PCI function 0

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

PCI function 1

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

Register: **Interrupt pin**
Type: Read-only
Offset: 3Dh
Default: The default depends on the interrupt signaling mode.

Table 4–5. Interrupt Pin Register Cross Reference

INTRTIE BIT	TIEALL BIT	INTPIN FUNCTION 0	INTPIN FUNCTION 1	INTPIN FUNCTION 2
0	0	0x01 ($\overline{\text{INTA}}$)	0x02 ($\overline{\text{INTB}}$)	0x03 ($\overline{\text{INTC}}$)
1	0	0x01 ($\overline{\text{INTA}}$)	0x01 ($\overline{\text{INTA}}$)	0x03 ($\overline{\text{INTC}}$)
x	1	0x01 ($\overline{\text{INTA}}$)	0x01 ($\overline{\text{INTA}}$)	0x01 ($\overline{\text{INTA}}$)

4.24 Bridge Control Register

The bridge control register provides control over various PCI4450 bridging functions. Some bits in this register are global in nature and should be accessed only through function 0.

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 (Function 0, 1)
Default: 0340h

Table 4–6. Bridge Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–11	RSVD	R	These bits 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 will inhibit performance on burst cycles. Note that bursted write data can be posted, but various write transactions may not. This bit is socket dependent and is not shared between functions 0 and 1.
9	PREFETCH1	R/W	Memory window 1 type. This bit specifies whether or not memory window 1 is prefetchable. This bit is socket dependent. This bit is encoded as: 0 = Memory window 1 is nonprefetchable. 1 = Memory window 1 is prefetchable (default).
8	PREFETCH0	R/W	Memory window 0 type. This bit 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	??	R/W	PCI Interrupt – IREQ routing enable. This bit is used to select 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 by ExCA registers.
6	CRST	R/W	CardBus reset. When this bit is set, the $\overline{\text{CRST}}$ signal is asserted on the CardBus interface. The $\overline{\text{CRST}}$ signal may also be asserted by passing a $\overline{\text{PRST}}$ assertion to CardBus. 0 = $\overline{\text{CRST}}$ is deasserted. 1 = $\overline{\text{CRST}}$ is asserted (default). This bit will not be cleared by the assertion of $\overline{\text{PRST}}$. It will only be cleared by the assertion of $\overline{\text{GRST}}$.
5†	MABTMODE	R/W	Master abort mode. This bit controls how the PCI4450 responds to a master abort when the PCI4450 is an initiator on the CardBus interface. This bit is common between each socket. 0 = Master aborts not reported (default). 1 = Signal target abort on PCI and signal $\overline{\text{SERR}}$, if enabled.
4	RSVD	R	This bit returns 0 when read.
3	VGAEN	R/W	VGA enable. This bit affects how the PCI4450 responds to VGA addresses. When this bit is set, accesses to VGA addresses will be forwarded.
2	ISAEN	R/W	ISA mode enable. This bit affects how the PCI4450 passes I/O cycles within the 64-Kbyte ISA range. This bit is not common between sockets. When this bit is set, the PCI4450 will not forward the last 768 bytes of each 1K I/O range to CardBus.
1	CSERREN	R/W	$\overline{\text{CSERR}}$ enable. This bit controls the response of the PCI4450 to $\overline{\text{CSERR}}$ signals on the CardBus bus. This bit is separate for each socket. 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. This bit controls the response of the PCI4450 to CardBus parity errors. This bit is separate for each socket. 0 = CardBus parity errors are ignored. 1 = CardBus parity errors are reported using $\overline{\text{CPERR}}$.

† This bit is global in nature and should be accessed only through function 0.

4.25 Subsystem Vendor ID Register

The subsystem vendor ID register, used for system and option card identification purposes, 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. When bit 5 is 0, this register is read/write; when bit 5 is 1, this register is read-only. The default mode is read-only.

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 (when bit 5 in the system control register is 0.)
Offset: 40h (Functions 0, 1)
Default: 0000h

4.26 Subsystem ID Register

The subsystem ID register, used for system and option card identification purposes, may be required for certain operating systems. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register (see Section). When bit 5 is 0, this register is read/write; when bit 5 is 1, this register is read-only. The default mode is read-only.

If an EEPROM is present, then the subsystem ID and subsystem vendor ID will be loaded from EEPROM after a reset.

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 (when bit 5 in the system control register is 0.)
Offset: 42h (Functions 0, 1)
Default: 0000h

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

The PCI4450 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. As specified in the Yenta specification, this register is shared by functions 0 and 1. Refer to the ExCA register set description 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 (Functions 0, 1)
Default: 0000 0001h

4.28 System Control Register

System level initializations are performed through programming this doubleword register. Some of the bits are global in nature and should be accessed only through function 0.

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/W	R/W	R	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	System control															
Type	R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Register: **System control**
Type: Read-only, Read/Write
Offset: 80h (Functions 0, 1)
Default: 0000 0020h

Table 4–7. System Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–30§	SER_STEP	R/W	Serialized PCI interrupt routing step. These bits are used to configure the serialized PCI interrupt stream signaling and accomplish an even distribution of interrupts signaled on the four PCI interrupt slots. These bits are global to both PCI4450 functions. 00 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$/$\overline{\text{INTC}}$ signal in $\overline{\text{INTA}}$/$\overline{\text{INTB}}$/$\overline{\text{INTC}}$ slots (default) 01 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$/$\overline{\text{INTC}}$ signal in $\overline{\text{INTB}}$/$\overline{\text{INTC}}$/$\overline{\text{INTD}}$ slots 10 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$/$\overline{\text{INTC}}$ signal in $\overline{\text{INTC}}$/$\overline{\text{INTD}}$/$\overline{\text{INTA}}$ slots 11 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$/$\overline{\text{INTC}}$ signal in $\overline{\text{INTD}}$/$\overline{\text{INTA}}$/$\overline{\text{INTB}}$ slots
29§	INTRTIE	R/W	Tie internal PCI interrupts. When this bit is set, the $\overline{\text{INTA}}$ and $\overline{\text{INTB}}$ signals are tied together internally and are signaled as $\overline{\text{INTA}}$. $\overline{\text{INTA}}$ may then be shifted by using the SER_STEP bits. This bit is global to both PCI4450 functions. This bit has no effect on $\overline{\text{INTC}}$. 0 = $\overline{\text{INTA}}$ and $\overline{\text{INTB}}$ are not tied together internally (default). 1 = $\overline{\text{INTA}}$ and $\overline{\text{INTB}}$ are tied together internally.
28	TIEALL	R/W	This bit ties $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, and $\overline{\text{INTC}}$ internally (to $\overline{\text{INTA}}$) and reports this through the interrupt pin register.
27§	P2CCLK	R/W	P2C power switch CLOCK. This bit determines whether the CLOCK terminal (terminal U12) is an input that requires an external clock source or if this terminal is an output that uses the internal oscillator. 0 = CLOCK terminal (terminal U12) is an input (default) (disabled). 1 = CLOCK terminal is an output, the PCI4450 generated CLOCK. A 43kΩ pulldown resistor should be tied to this terminal.
26§	SMIROUTE	R/W	SMI interrupt routing. This bit is shared between functions 0 and 1, and selects whether IRQ2 or CSC is signaled when a write occurs to power a PC Card socket. 0 = PC Card power change interrupts routed to IRQ2 (default). 1 = A CSC interrupt is generated on PC Card power changes.
25	SMISTATUS	R/W	SMI interrupt status. This socket dependent bit is set when a write occurs to set the socket power, and the SMIENB bit is set. Writing a 1 to this bit clears the status. 0 = SMI interrupt is signaled. 1 = SMI interrupt is not signaled.
24§	SMIENB	R/W	. SMI interrupt mode enable. When this bit is set, the SMI interrupt signaling generates an interrupt when a write to the socket power control occurs. This bit is shared and defaults to 0 (disabled). 0 = SMI interrupt mode is disabled (default). 1 = SMI interrupt mode is enabled.
23	RSVD	R	Reserved

§ These bits are global in nature and should be accessed only through function 0.

Table 4–7. System Control Register Description (continued)

BIT	SIGNAL	TYPE	FUNCTION
22	CBRSVD	R/W	CardBus reserved terminals signaling. When this bit is set, the RSVD CardBus terminals are driven low when a CardBus card is inserted. When this bit is low, as default, these signals are placed in a high-impedance state. 0 = Place the CardBus RSVD terminals in a high-impedance state 1 = Drive the Cardbus RSVD terminals low (default).
21	VCCPROT	R/W	V_{CC} protection enable. This bit is socket dependent. 0 = V_{CC} protection is enabled for 16-bit cards (default). 1 = V_{CC} protection is disabled for 16-bit cards.
20	??	R/W	Reduced zoomed video enable. When this bit is enabled, A25–22 of the card interface for PC Card 16 cards is placed in the high impedance state. 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 this bit is set, the PCI4450 allows 16-bit PC Cards to request PC/PCI DMA using the $\overline{\text{DREQ}}$ signaling. $\overline{\text{DREQ}}$ is selected through the socket DMA register 0. 0 = Ignore $\overline{\text{DREQ}}$ signaling from PC Cards (default). 1 = Signal DMA request on $\overline{\text{DREQ}}$.
18–16	CDMACHAN	R/W	PC/PCI DMA channel assignment. These bits 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 this bit is set, memory read transactions are allowed to burst downstream. 0 = MRBURSTDN downstream is disabled. 1 = MRBURSTDN downstream is enabled (default).
14§	MRBURSTUP	R/W	Memory read burst enable upstream. When this bit is set, the PCI4450 allows memory read transactions to burst upstream. 0 = MRBURSTUP upstream is disabled (default). 1 = MRBURSTUP upstream is enabled.
13	SOCACTIVE	R	Socket activity status. When set, this bit indicates access has been performed to or from a PC Card, and is cleared upon read of this status bit. This bit is socket dependent. 0 = No socket activity (default) 1 = Socket activity
12	RSVD	R	Reserved. This bit returns 1 when read. This is the power rail bit in functions 0 and 1.
11	PWRSTREAM	R	Power stream in progress status bit. When set, this bit indicates that a power stream to the power switch is in progress and a powering change has been requested. When this bit is clear, it indicates that the power stream is complete. 0 = Power stream is complete, delay has expired. 1 = Power stream is in progress.
10	DELAYUP	R	Power-up delay in progress status bit. When set, this bit 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. 0 = Power-up delay has expired. 1 = Power-up stream sent to switch. Power might not be stable.
9	DELAYDOWN	R	Power-down delay in progress status bit. When set, this bit 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. 0 = Power-down delay has expired. 1 = Power-down stream sent to switch. Power might not be stable.
8	INTERROGATE	R	Interrogation in progress. When set, this bit indicates an interrogation is in progress, and clears when the interrogation completes. This bit is socket dependent. 0 = Interrogation not in progress (default) 1 = Interrogation in progress
7	RSVD	R	Reserved. This bit returns 0 when read.

§ These bits are global in nature and should be accessed only through function 0.

Table 4–7. System Control Register Description (continued)

BIT	SIGNAL	TYPE	FUNCTION
6§	PWRSAVINGS	R/W	Power savings mode enable. When this bit is set, the PCI4450 will consume less power with no performance loss. This bit is shared between the two PCI4450 functions. 0 = Power savings mode disabled 1 = Power savings mode enabled (default)
5§	SUBSYSRW	R/W	Subsystem ID (SS ID), subsystem vendor ID (SS VID), and the ExCA identification and revision registers read/write enable. This bit is shared by functions 0 and 1. This bit does not control read/write of function 2, subsystem ID register. 0 = Subsystem ID, subsystem vendor ID, and the ExCA identification and revision registers are read/write. 1 = Subsystem ID, subsystem vendor ID, and the ExCA identification and revision registers are read-only (default).
4§	CB_DPAR	R/W	CardBus data parity SERR signaling enable. 0 = CardBus data parity not signaled on PCI $\overline{\text{SERR}}$ signal (default) 1 = CardBus data parity signaled on PCI $\overline{\text{SERR}}$ signal
3§	CDMA_EN	R/W	PC/PCI DMA enable. Enables PC/PCI DMA when set. When PC/PCI DMA is enabled, $\overline{\text{PCREQ}}$ and $\overline{\text{PCGNT}}$ should be routed to a multifunction routing terminal. See multifunction routing status register for options. 0 = Centralized DMA disabled (default) 1 = Centralized DMA enabled
2	RSVD	R	Reserved. This bit returns 0 when read.
1§	KEEPCLK	R/W	Keep clock. When this bit is set, the PCI4450 will always follow $\overline{\text{CLKRUN}}$ protocol to maintain the system PCLK and the CCLK (CardBus clock). This bit is global to the PCI4450 functions. 0 = Allow system PCLK and CCLK to stop (default) 1 = Never allow system PCLK or CCLK clock to stop Note that the functionality of this bit has changed versus the PCI12XX series of TI CardBus controllers. In these CardBus controllers, setting this bit would only maintain the PCI clock, not the CCLK. In the PCI4450, setting this bit will maintain both the PCI clock and the CCLK.
0§	??	R/W	$\overline{\text{PME}}$/$\overline{\text{RI\_OUT}}$ select bit. When this bit is 1, the PME signal is routed on to pin Y13 ($\overline{\text{PME}}$/$\overline{\text{RI\_OUT}}$ pin). When this bit is 0 and bit 7 (RIENB) of the card control register is 1, the $\overline{\text{RI\_OUT}}$ signal is routed on to pin Y13 (GFN) or pin R11 (GJG). If this bit is 0 and bit 7 (RIENB) of the card control register is 0, then the output (Y13 or R11) will be placed in a high-impedance state. This pin is encoded as: 0 = $\overline{\text{RI\_OUT}}$ signal is routed to pin Y13 (GFN) or pin R11 (GJG) if bit 7 of the card control register is 1*. (default) 1 = $\overline{\text{PME}}$ signal is routed on pin Y13 (GFN) or pin R11 (GJG) of the PCI4450 controller. NOTE: If this bit (bit 0) is 0 and bit 7 of the card control register is 0, then the output on pin Y13 (GFN) or pin R11 (GJG) is placed in a high-impedance state.

§ These bits are global in nature and should be accessed only through function 0.

4.29 Multimedia Control Register

The multimedia control register provides port mapping for the PCI4450 zoomed video/data ports. See *zoomed video support* for details on the PCI4450 zoomed video support. Access this register only through function 0.

BIT	7	6	5	4	3	2	1	0
Name	Multimedia 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: **Multimedia control**
Type: Read/Write
Offset: 84h (Functions 0, 1)
Default: 00h

Table 4–8. Multimedia Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	ZVOUTEN	R/W	ZV output enable. This bit enables the output for the PCI4450 outsourcing ZV terminals. When this bit is reset, '0', these terminals are in a high impedance state. 0 = PCI4450 ZV output terminals disabled (default) 1 = PCI4450 ZV output terminals enabled
6	PORTSEL	R/W	ZV port select. This bit controls the multiplexing control over which PC Card ZV port data will be driven to the outsourcing PCI4450 ZV port. 0 = Output card 0 ZV if enabled (default) 1 = Output card 1 ZV if enabled
5	??	R/W	Zoomed video auto-detect. This bit enables the zoomed video auto-detect feature. This bit is encoded as: 0 = Zoomed video auto detect disabled (default) 1 = Zoomed video auto detect enabled
4–2	??	R/W	Auto-detect priority encoding. These bits have meaning only if zoomed video auto-detect is enabled in bit 5 of this register. If auto-detect is enabled, then bits 4–2 are encoded as follows: 000 = Slot A, Slot B, External Source 001 = Slot A, External Source, Slot B 010 = Slot B, Slot A, External Source 011 = Slot B, External Source, Slot A 100 = External Source, Slot A, Slot B 101 = External Source, Slot B, Slot A 110 = Reserved 111 = Reserved
1	ZVEN1	R/W	PC Card 1 ZV mode enable. Enables the zoomed video mode for socket 1. When set, the PCI4450 inputs ZV data from the PC Card interface, and disables output drivers on ZV terminals. 0 = PC Card 1 ZV disabled (default) 1 = PC Card 1 ZV enabled
0	ZVEN0	R/W	PC Card 0 ZV mode enable. Enables the zoomed video mode for socket 0. When set, the PCI4450 inputs ZV data from the PC Card interface, and disables output drivers on ZV terminals. 0 = PC Card 0 ZV disabled (default) 1 = PC Card 0 ZV enabled

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.

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

Register: **General status**
Type: Read/Update
Offset: 85h (Functions 0)
Default: 00h

Table 4–9. General Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	IDSEL_DET	R/U	When this bit is set, the IDSEL/MFUNC7 terminal functions as an IDSEL input.
6–3	RSVD	R	Reserved. These bits return 0s when read.
2§	EEDETECT	R/U	Serial EEPROM detect. Serial EEPROM is detected by sampling a logic high on SCL on $\overline{\text{PRST}}$. When this bit is set, the serial ROM is detected. This status bit is encoded as: 0 = EEPROM not detected (default) 1 = EEPROM detected
1§	DATAERR	R	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 writing a 1. 0 = No error detected. (default) 1 = Data error detected.
0§	EEBUSY	R	Serial EEPROM busy status. This bit indicates the status of the PCI4450 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.

§ This bit is global in nature and should only be accessed through function 0.

4.31 General Control Register

The general control register provides top level PCI arbitration control.

Bit	7	6	5	4	3	2	1	0
Name	General control							
Type	R	R	R	R	R/W	R/W	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 4–10. General Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–4	RSVD	R	Reserved. These bits return 0s when read.
3	DISABLE_OHCI	R/W	When this bit is set, the open HCI 1394 controller function is completely nonaccessible and nonfunctional.
2	GP2IIC	R/W	When this bit is set, the GPO0 and GPO1 signals are routed to SDA and SCL, respectively.
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 General-Purpose Event Status Register

The general-pupose event status register contains status bits that are set when general events occur and may be programmed to generate general-purpose event signalling through $\overline{\text{GPE}}$.

Bit	7	6	5	4	3	2	1	0
Name	General-purpose event status							
Type	RCU	RCU	R	R	RCU	RCU	RCU	RCU
Default	0	0	0	0	0	0	0	0

Register: **General-purpose event status**
Type: Read/Clear/Update
Offset: 88h
Default: 00h

Table 4–11. General-Purpose Event Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	PWR_STS	RCU	Power change status. This bit is set when software changes the V_{CC} or V_{PP} power state of either socket.
6	VPP12_STS	RCU	12V V_{PP} request status. This bit is set when software has changed the requested V_{PP} level to or from 12 V for either socket.
5–4	RSVD	R	Reserved. This bit returns 0 when read. A write has no effect.
3	GP3_STS	RCU	GPI3 status. This bit is set on a change in status of the MFUNC3 terminal input level if configured as a general-purpose input, GPI3.
2	GP2_STS	RCU	GPI2 status. This bit is set on a change in status of the MFUNC2 terminal input level if configured as a general-purpose input, GPI2.
1	GP1_STS	RCU	GPI1 status. This bit is set on a change in status of the MFUNC1 terminal input level if configured as a general-purpose input, GPI1.
0	GP0_STS	RCU	GPI0 status. This bit is set on a change in status of the MFUNC0 terminal input level if configured as a general-purpose input, GPI0.

4.33 General-Purpose Event Enable Register

The general-purpose event enable register contains bits that are set to enable $\overline{\text{GPE}}$ signals.

Bit	7	6	5	4	3	2	1	0
Name	General-purpose event enable							
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: **General-purpose event enable**
Type: Read-only, Read/Write
Offset: 89h
Default: 00h

Table 4–12. General-Purpose Event Enable Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	PWR_EN	R/W	Power change $\overline{\text{GPE}}$ enable. When this bit is set, $\overline{\text{GPE}}$ is signaled on PWR_STS events.
6	VPP12_EN	R/W	12-Volt V_{PP} $\overline{\text{GPE}}$ enable. When this bit is set, $\overline{\text{GPE}}$ is signaled on VPP12_STS events.
5–4	RSVD	R	Reserved. This bit returns 0 when read. A write has no effect.
3	GP3_EN	R/W	GPI3 $\overline{\text{GPE}}$ enable. When this bit is set, $\overline{\text{GPE}}$ is signaled on GP3_STS events.
2	GP2_EN	R/W	GPI2 $\overline{\text{GPE}}$ enable. When this bit is set, $\overline{\text{GPE}}$ is signaled on GP2_STS events.
1	GP1_EN	R/W	GPI1 $\overline{\text{GPE}}$ enable. When this bit is set, $\overline{\text{GPE}}$ is signaled on GP1_STS events.
0	GP0_EN	R/W	GPI0 $\overline{\text{GPE}}$ enable. When this bit is set, $\overline{\text{GPE}}$ is signaled on GP0_STS events.

4.34 General-Purpose Input Register

The general-purpose input register contains GPI terminal status.

Bit	7	6	5	4	3	2	1	0
Name	General-purpose input							
Type	R	R	R	R	RU	RU	RU	RU
Default	0	0	0	0	x	x	x	x

Register: **General-purpose input**
Type: Read/Update
Offset: 8Ah
Default: 00h

Table 4–13. General-Purpose Input Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–4	RSVD	R	Reserved. These bits return 0s when read. Writes have no effect.
3	GPI3_DATA	RU	GPI3 data input. This bit represents the logical value of the data input from GPI3.
2	GPI2_DATA	RU	GPI2 data input. This bit represents the logical value of the data input from GPI2.
1	GPI1_DATA	RU	GPI1 data input. This bit represents the logical value of the data input from GPI1.
0	GPI0_DATA	RU	GPI0 data input. This bit represents the logical value of the data input from GPI0.

4.35 General-Purpose Output Register

The general-purpose output register is used to drive the GPO3–GPO0 outputs.

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

Register: **General-purpose output**
Type: Read-only, Read/Write
Offset: 8Bh
Default: 00h

Table 4–14. General-Purpose Output Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–4	RSVD	R	Reserved. These bits return 0s when read. Writes have no effect.
3	GPO3_DATA	R/W	This bit represents the logical value of the data driven to GPO3.
2	GPO2_DATA	R/W	This bit represents the logical value of the data driven to GPO2.
1	GPO1_DATA	R/W	This bit represents the logical value of the data driven to GPO1.
0	GPO0_DATA	R/W	This bit represents the logical value of the data driven to GPO0.

4.36 Multifunction Routing Status Register

The multifunction routing status register is used to configure MFUNC7–MFUNC0 terminals. These terminals may be configured for various functions. This register is intended to be programmed once at power-on initialization. The default value for this register may also be loaded through a serial ROM.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Multifunction routing status															
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	Multifunction routing status															
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 status**
Type: Read/Write
Offset: 8Ch
Default: 0000 0000h

Table 4–15. Multifunction Routing Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
31	RSVD	R	Reserved. This bit returns 0 when read. Writes have no effect.
30–28	MFUNC7_SEL	R/W	MFUNC7 select. These bits control the mapping of MFUNC7 as follows: 000 = IDSEL 001 = $\overline{\text{RI\_OUT}}$ 010 = OHCI_LED 011 = $\overline{\text{PCREQ}}$ 100 = $\overline{\text{D3\_STAT}}$ 101 = $\overline{\text{LOCK}}$ 110 = RSVD 111 = RSVD
27	RSVD	R	Reserved. This bit returns 0 when read. Writes have no effect.
26–24	MFUNC6_SEL	R/W	MFUNC6 select. These bits control the mapping of MFUNC6 as follows: 000 = RSVD 001 = RSVD 010 = OHCI_LED 011 = RSVD 100 = $\overline{\text{D3\_STAT}}$ 101 = RSVD 110 = CAUDPWM 111 = RSVD
23	RSVD	R	Reserved. This bit returns 0 when read. Writes have no effect.
22–20	MFUNC5_SEL	R/W	MFUNC5 select. These bits control the mapping of MFUNC5 as follows: 000 = RSVD 001 = RSVD 010 = OHCI_LED 011 = RSVD 100 = RSVD 101 = $\overline{\text{GPE}}$ 110 = CAUDPWM 111 = RSVD
19	RSVD	R	Reserved. This bit returns 0 when read. Writes have no effect.
18–16	MFUNC4_SEL	R/W	MFUNC4 select. These bits control the mapping of MFUNC4 as follows: 000 = RSVD 001 = RSVD 010 = LEDA1 011 = $\overline{\text{PCREQ}}$ 100 = RSVD 101 = $\overline{\text{GPE}}$ 110 = RSVD 111 = ZV_STAT
15	RSVD	R	Reserved. This bit returns 0 when read. Writes have no effect.
14–12	MFUNC3_SEL	R/W	MFUNC3 select. These bits control the mapping of MFUNC3 as follows: 000 = GPI3 001 = GPO3 010 = LEDA2 011 = $\overline{\text{PCGNT}}$ 100 = RSVD 101 = $\overline{\text{LOCK}}$ 110 = RSVD 111 = C_ZVCLK

Table 4–15. Multifunction Routing Status Register Description (continued)

BIT	SIGNAL	TYPE	FUNCTION
11	RSVD	R	Reserved. This bit returns 0 when read. Writes have no effect.
10–8	MFUNC2_SEL	R/W	MFUNC2 select. These bits control the mapping of MFUNC2 as follows: 000 = GPI2 001 = GPO2 010 = $\overline{\text{INTC}}$ 011 = $\overline{\text{PCGNT}}$ 100 = $\overline{\text{D3\_STAT}}$ 101 = RSVD 110 = RSVD 111 = C_ZVCLK
7	RSVD	R	Reserved. This bit returns 0 when read. Writes have no effect.
6–4	MFUNC1_SEL	R/W	MFUNC1 select. These bits control the mapping of MFUNC1 as follows: 000 = GPI1 001 = GPO1 010 = $\overline{\text{INTB}}$ 011 = TEST_MUX 100 = $\overline{\text{D3\_STAT}}$ 101 = $\overline{\text{LOCK}}$ 110 = CAUDPWM 111 = ZV_STAT
3	RSVD	R	Reserved. This bit returns 0 when read. Writes have no effect.
2–0	MFUNC0_SEL	R/W	MFUNC0 select. These bits control the mapping of MFUNC0 as follows: 000 = GPI0 001 = GPO0 010 = $\overline{\text{INTA}}$ 011 = $\overline{\text{PCREQ}}$ 100 = RSVD 101 = $\overline{\text{GPE}}$ 110 = RSVD 111 = ZV_STAT

4.37 Retry Status Register

The contents of the retry status register enable the retry time-out counters and display the retry expiration status. The flags are set when the PCI4450 retries a PCI or CardBus master request, and the master does not return within 2^{15} PCI clock cycles. The flags are cleared by writing a 1 to the bit. These bits are expected to be incorporated into the PCI command register, PCI status register, and bridge control register by the PCI SIG. Access this register only through function 0.

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

Register: **Retry status**
Type: Read-only, Read/Write
Offset: 90h (Functions 0, 1)
Default: C0h

Table 4–16. Retry Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	PCIRETRY	R/W	PCI retry time-out counter enable. This bit is encoded as: 0 = PCI retry counter disabled 1 = PCI retry counter enabled (default)
6§	CBRETRY	R/W	CardBus retry time-out counter enable. This bit is encoded as: 0 = CardBus retry counter disabled 1 = CardBus retry counter enabled (default)
5	TEXP_CBB	R/WC	CardBus target B retry expired. Write a 1 to clear this bit. 0 = Inactive (default) 1 = Retry has expired.
4	RSVD	R	Reserved. This bit returns 0 when read.
3§	TEXP_CBA	R/WC	CardBus target A retry expired. Write a 1 to clear this bit. 0 = Inactive (default) 1 = Retry has expired.
2	RSVD	R	Reserved. This bit returns 0 when read.
1	TEXP_PCI	R/WC	PCI target retry expired. Write a 1 to clear this bit. 0 = Inactive (default) 1 = Retry has expired.
0	RSVD	R	Reserved. This bit returns 0 when read.

§ These bits are global in nature and should be accessed only through function 0.

4.38 Card Control Register

The card control register is provided for PCI1130 compatibility. The contents provide the PC Card function interrupt flag (IFG) and an alias for the ZVEN0 and ZVEN1 bits found in the PCI4450 multimedia control register. When this register is accessed by function 0, the ZVEN0 bit will alias with ZVENABLE. When this register is accessed by function 1, the ZVEN1 bit will alias with ZVENABLE. Setting ZVENABLE only places the PC Card socket interface ZV terminals in a high impedance state, but does not enable the PCI4450 to drive ZV data onto the ZV terminals.

The $\overline{\text{RI\_OUT}}$ signal is enabled through this register, and the enable bit is shared between functions 0 and 1.

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/W
Default	0	0	0	0	0	0	0	0

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

Table 4–17. Card Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7§	RIENB	R/W	Ring indicate enable. When this bit is 1, the $\overline{\text{RI\_OUT}}$ output is enabled. This bit is global in nature and should be accessed only through function 0. This bit defaults to 0.
6	ZVENABLE	R/W	Compatibility ZV mode enable. When this bit is 1, the corresponding PC Card socket interface ZV terminals will enter a high impedance state. This bit defaults to 0.
5	RSVD	R/W	Reserved.
4–3	RSVD	R	Reserved. These bits default to 0.
2	AUD2MUX	R/W	CardBus Audio-to-MFUNC. When this bit is set, the CAUDIO CardBus signal must be routed through an MFUNC terminal. If this bit is set for both functions, then function 0 gets routed. 0 = CAUDIO set to CAUDPWM on MFUNC pin (default) 1 = CAUDIO is not routed.
1	SPKROUTEN	R/W	Speaker output enable. When this bit is 1, it enables $\overline{\text{SPKR}}$ on the PC Card and routes it to SPKROUT on the PCI bus. The $\overline{\text{SPKR}}$ signal from socket 0 is XOR'ed with the $\overline{\text{SPKR}}$ signal from socket 1 and sent to SPKROUT. The SPKROUT terminal only drives data then either functions SPKROUTEN bit is set. This bit is encoded as: 0 = $\overline{\text{SPKR}}$ to SPKROUT not enabled (default) 1 = $\overline{\text{SPKR}}$ to SPKROUT enabled
0	IFG	R/W	Interrupt flag. This bit is the interrupt flag for 16-bit I/O PC Cards and for CardBus cards. This bit is set when a functional interrupt is signaled from a PC Card interface, and is socket dependent (i.e., not global). Write back a '1' to clear this bit. 0 = No PC Card functional interrupt detected (default) 1 = PC Card functional interrupt detected

§ These bits are global in nature and should be accessed only through function 0.

4.39 Device Control Register

The device control register is provided for PCI1130 compatibility. It contains bits which are shared between functions 0 and 1. The interrupt mode select is programmed through this register. The socket capable force bits are also programmed through this register.

Bit	7	6	5	4	3	2	1	0
Name	Device control							
Type	R/W	R/W	R/W	R	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 (Functions 0, 1)
Default: 66h

Table 4–18. Device Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	??	R/W	Socket power lock bit. When this bit is set to 1, software will not be able to power down the PC Card socket while in D3. This may be necessary to support Wake on LAN or RING if the operating system is programmed to power down a socket when the CardBus controller is placed in the D3 state.
6§	3VCAPABLE	R/W	3-V socket capable force bit. 0 = Not 3-V capable 1 = 3-V capable (default)
5	IO16R2	R/W	Diagnostic bit. This bit defaults to 1.
4	RSVD	R	Reserved. This bit returns 0 when read. A write has no effect.
3§	TEST	R/W	TI test bit. Write only 0 to this bit. This bit can be set to shorten the interrogation counter.
2–1§	INTMODE	R/W	Interrupt mode. These bits select the interrupt signaling mode. The interrupt mode bits are encoded: 00 = Parallel PCI interrupts only 01 = Reserved 10 = IRQ serialized interrupts & parallel PCI interrupts $\overline{\text{INTA}}$ and $\overline{\text{INTB}}$ 11 = IRQ & PCI serialized interrupts (default)
0§	RSVD	R/W	Reserved. NAND tree enable bit. There is a NAND tree diagnostic structure in the PCI4450, and it tests only the pins that are inputs or I/Os. Any output only terminal on the PCI4450 is excluded from the NAND tree test.

§ These bits are global in nature and should be accessed only through function 0.

4.40 Diagnostic Register

The diagnostic register is provided for internal Texas Instruments test purposes.

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	1	1	0	0	0	0	1

Register: **Diagnostic**
Type: Read/Write
Offset: 93h (Functions 0, 1)
Default: 61h

Table 4–19. Diagnostic Register Description

BIT	SIGNAL	TYPE	FUNCTION
7§	??	R/W	This bit defaults to 0. This bit is encoded as: 0 = Reads true values in PCI vendor ID and PCI device ID registers (default). 1 = Reads all ones in reads to the PCI vendor ID and PCI device ID registers.
6		R/W	Reserved.
5	??	R/W	CSC interrupt routing control 0 = CSC interrupts routed to PCI if ExCA 803 bit 4 = 1. 1 = CSC Interrupts routed to PCI if ExCA 805 bits 7–4 = 0000b. (Default) In this case, the setting of ExCA 803 bit 4 is a “don’t care.”
4§	DIAG	R/W	Diagnostic RETRY_DIS. Delayed transaction disable.
3§	DIAG	R/W	Diagnostic RETRY_EXT. Extends the latency from 16 to 64.
2§	DIAG	R/W	Diagnostic DISCARD_TIM_SEL_CB. Set = 2^{10}, Reset = 2^{15}
1§	DIAG	R/W	Diagnostic DISCARD_TIM_SEL_PCI. Set = 2^{10}, Reset = 2^{15}
0§	ASYNC_CSC	R/W	Asynchronous interrupt generation. 0 = CSC interrupt not generated asynchronously 1 = CSC interrupt is generated asynchronously (default)

§ These bits are global in nature and should be accessed only through function 0.

4.41 Socket DMA Register 0

This register provides control over the PC Card $\overline{\text{DREQ}}$ (DMA request) signaling.

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: **DMA socket register 0**
Type: Read-only, Read/Write
Offset: 94h (Functions 0, 1)
Default: 0000 0000h

Table 4–20. Socket DMA Register 0 Description

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

4.42 Socket DMA Register 1

The contents of this register provide 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. Note that 32-bit transfers to the 16-bit PC Card interface are not supported; the maximum transfer possible to the PC Card interface is 16-bits. However, 32 bits of data are prefetched from the PCI bus, thus allowing back-to-back 16-bit transfers to the PC Card interface.

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: **DMA socket register 1**
Type: Read-only, Read/Write
Offset: 98h (Functions 0, 1)
Default: 0000 0000h

Table 4–21. Socket DMA Register 1 Description

BIT	SIGNAL	TYPE	FUNCTION
31–16	RSVD	R	Reserved. These bits 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 four 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 PCI4450, and always returns a 0.
2–1	XFERSIZE	R/W	Transfer size. These bits 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 upon the value of DMABASE. 0 = Disabled (default) 1 = Enabled

4.43 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

4.44 Next Item Pointer Register

The contents of this register indicate the next item in the linked list of the PCI power management capabilities. Since the PCI4450 functions only include 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

4.45 Power Management Capabilities Register

The power management capabilities register contains information on the capabilities of the PC Card function related to power management. Both PCI4450 CardBus bridge functions support D0, D1, D2, and D3 power states.

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	0	1	0	0	0	1

Register: **Power management capabilities**
Type: Read-only, Read/Write
Offset: A2h (Functions 0, 1)
Default: FE11h

Table 4–22. Power Management Capabilities Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	PME support	R/W	This 5-bit field indicates the power states from which the PCI4450 device functions may assert $\overline{PME}$. A 0b (zero) for any bit indicates that the function cannot assert the $\overline{PME}$ signal while in that power state. These five bits return 0Fh when read. Each of these bits is described below: Bit 15 – defaults to a 1 indicating the $\overline{PME}$ signal can be asserted from the $D3_{cold}$ state. This bit is read/write because wake-up support from $D3_{cold}$ is contingent on the system providing an auxiliary power source to the V_{CC} terminals. If the system designer chooses not to provide an auxiliary power source to the V_{CC} terminals for $D3_{cold}$ wake-up support, then BIOS should write a 0 to this bit.
14–11		R	Bit 14 – contains the value 1 to indicate that the $\overline{PME}$ signal can be asserted from the $D3_{hot}$ state. Bit 13 – contains the value 1 to indicate that the $\overline{PME}$ signal can be asserted from the D2 state. Bit 12 – contains the value 1 to indicate that the $\overline{PME}$ signal can be asserted from the D1 state. Bit 11 – contains the value 1 to indicate that the $\overline{PME}$ signal can be asserted from the D0 state.
10	D2_Support	R	This bit returns a 1 when read, indicating that the function supports the D2 device power state.
9	D1_Support	R	This bit returns a 1 when read, indicating that the function supports the D1 device power state.
8–6	RSVD	R	Reserved. These bits return 000b when read.
5	DSI	R	Device specific initialization. This bit returns 0 when read.
4	AUX_PWR	R	Auxiliary power source. This bit is meaningful only if bit 15 ($D3_{cold}$ supporting $\overline{PME}$) is set. When this bit 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. A 0 (zero) in this bit field indicates that the function supplies its own auxiliary power source. If the function does not support PME while in the $D3_{cold}$ state (bit 15=0), then this field must always return 0.
3	PMECLK	R	When this bit is 1, it indicates that the function relies on the presence of the PCI clock for $\overline{PME}$ operation. When this bit is 0, it indicates that no PCI clock is required for the function to generate $\overline{PME}$. Functions that do not support PME generation in any state must return 0 for this field.
2–0	Version	R	These 3 bits return 001b when read, indicating that there are 4 bytes of general-purpose power management (PM) registers as described in the draft revision 1.0 *PCI Bus Power Management Interface Specification*.

4.46 Power Management Control/Status Register

The power management control/status register determines and changes the current power state of the PCI4450 CardBus function. The contents of this register are not affected by the internally generated reset caused by the transition from the $D3_{hot}$ to D0 state.

All PCI registers, ExCA registers, and CardBus registers are reset as a result of a $D3_{hot}$-to-D0 state transition, with the exception of the $\overline{PME}$ context bits (if $\overline{PME}$ is enabled) and the $\overline{GRST}$ only bits.

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

Register: **Power management control/status**
Type: Read-only, Read/Write, Read/Write to Clear
Offset: A4h (Functions 0, 1)
Default: 0000h

Table 4–23. Power Management Control/Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	PMESTAT	R/WC	PME status. This bit is set when the CardBus function would normally assert the $\overline{PME}$ signal, independent of the state of the PME_EN bit. This bit is cleared by a write back 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	This 2-bit field returns 0s when read. The CardBus function does not return any dynamic data, as indicated by the DYN_DATA bit.
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 the DYN_DATA bit.
8	PME enable	R/W	This bit enables the function to assert $\overline{PME}$. If this bit is cleared, then assertion of $\overline{PME}$ is disabled. This bit will not be cleared by the assertion of $\overline{PRST}$. It will only be cleared by the assertion of $\overline{GRST}$.
7–2	RSVD	R	Reserved. These bits return 0s when read.
1–0	PWRSTATE	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 = $D3_{hot}$

4.47 Power Management Control/Status Register Bridge Support Extensions

This register supports PCI bridge specific functionality. It is required for all PCI-to-PCI bridges.

Bit	7	6	5	4	3	2	1	0
Name	Power management control/status register bridge support extensions							
Type	R	R	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 (Functions 0, 1)
Default: C0h

Table 4–24. Power Management Control/Status Register Bridge Support Extensions

BIT	SIGNAL	TYPE	FUNCTION
7	BPCC_Enable	R	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 Power Management specification are disabled. When the bus power/clock control enable mechanism is disabled, the bridge's PMCSR powerstate field 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. When bus power/clock control is disabled, the bridge's PMCSR power state field cannot be used by the system software to control power or the clock of the bridge's secondary bus.
6	$\overline{B2\_B3}$	R	B2/B3 support for $D3_{hot}$. The state of this bit determines the action that is to occur as a direct result of programming the function to $D3_{hot}$. This bit is only meaningful if bit 7 (BPCC_Enable) 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 is stopped (B2). (Default)
5–0	RSVD	R	Reserved. These bits return 0s when read.

4.48 $\overline{\text{GPE}}$ Control/Status Register

If the $\overline{\text{GPE}}$ (general-purpose event) function is programmed onto the MFUNC5 pin by writing 101b to bits 22–20 of the multifunction routing register (PCI offset 8Ch, see Section), then this register may be used to program which events will cause $\overline{\text{GPE}}$ to be asserted and report the status.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	$\overline{\text{GPE}}$ control/status															
Type	R	R	R	R	R	R/WC	R/WC	R/WC	R	R	R	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: **$\overline{\text{GPE}}$ control/status**
Type: Read-only, Read/Write, Read/Write to Clear
Offset: A8h
Default: 0001h

Table 4–25. $\overline{\text{GPE}}$ Control/Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–11	RSVD	R	Reserved. These bits return 0s when read.
10	ZV1_STS	R/WC	PC Card socket 1 status. This bit is set on a change in status of the ZVENABLE bit in function 1.
9	ZV0_STS	R/WC	PC Card socket 0 status. This bit is set on a change in status of the ZVENABLE bit in function 0.
8	VPP12_STS	R/WC	12-volt V_{PP} request status. This bit is set when software has changed the requested V_{PP} level to or from 12 volts from either socket.
7–3	RSVD	R	Reserved. These bits return 0s when read.
2	ZV1_EN	R/W	PC Card socket 1 zoomed video event enable. When this bit is set, $\overline{\text{GPE}}$ is signaled on a change in status of the ZVENABLE bit in function 1 of the PC Card controller.
1	ZV0_EN	R/W	PC Card socket 0 zoomed video event enable. When this bit is set, $\overline{\text{GPE}}$ is signaled on a change in status of the ZVENABLE bit in function 0 of the PC Card controller.
0	VPP12_EN	R/W	12 Volt V_{PP} request event enable. When this bit is set, a $\overline{\text{GPE}}$ is signaled when software has changed the requested V_{PP} level to or from 12 Volts for either socket.

5 ExCA Compatibility Registers (Functions 0 and 1)

The ExCA (exchangeable card architecture) registers implemented in the PCI4450 are register-compatible with the Intel 82365SL-DF PCMCIA controller. ExCA registers are identified by an offset value, which 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, which is shared by both card sockets. The offsets from this base address run contiguous from 00h to 3Fh for socket A, and from 40h to 7Fh for socket B. Refer to Figure 5–1 for an ExCA I/O mapping illustration. Table 5–1 identifies each ExCA register and its respective ExCA offset.

The TI PCI4450 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 registers/ExCA registers base address register (PCI register 10h) at memory offset 800h. Each socket has a separate base address programmable by function. Refer to Figure 5–2 for an ExCA memory mapping illustration. Note that memory offsets are 800h–844h for both functions 0 and 1. This illustration also identifies the CardBus socket register mapping, which is mapped into the same 4K window at memory offset 0h.

The interrupt registers, as defined by the 82365SL Specification, 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 PCI4450 to ensure that all possible PCI4450 interrupts can potentially be routed to the programmable interrupt controller. The ExCA registers that are critical to the interrupt signaling are at memory address ExCA offset 803h and 805h.

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. Memory windows have 4K byte granularity.

A bit location followed by a ‡ means that this bit is not cleared by the assertion of $\overline{\text{PRST}}$. This bit will only be cleared by the assertion of $\overline{\text{GRST}}$. This is necessary to retain device context when transitioning from D3 to D0.

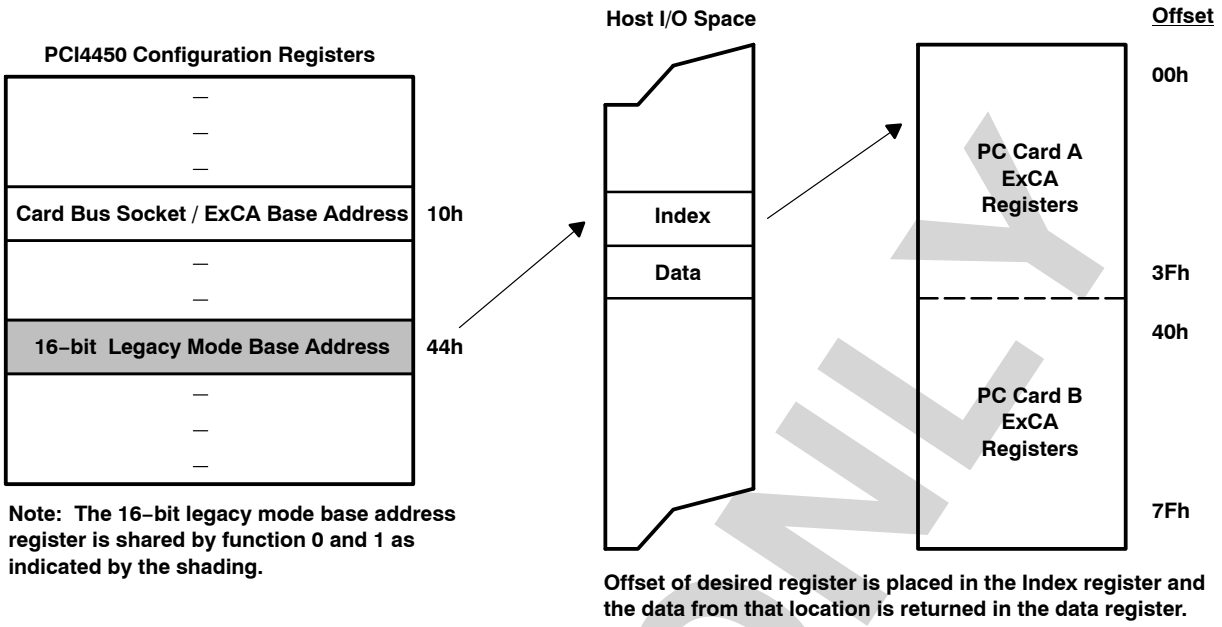


Figure 5–1. ExCA Register Access Through I/O

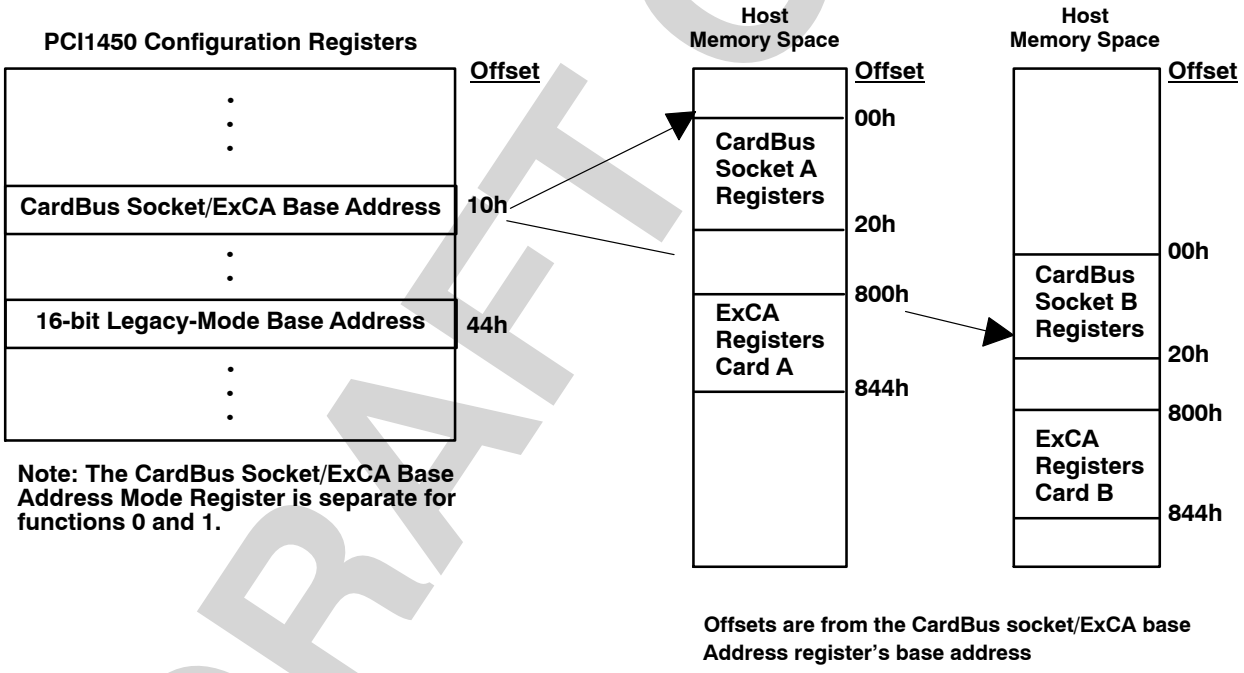


Figure 5–2. ExCA Register Access Through Memory

Table 5–1. ExCA Registers and Offsets

REGISTER NAME	PCI MEMORY ADDRESS OFFSET	EXCA OFFSET (CARD A)	EXCA OFFSET (CARD B)
Identification and revision	800	00	40
Interface status	801	01	41
Power control †	802†	02	42
Interrupt and general control †	803†	03	43
Card status change †	804†	04	44
Card status change interrupt configuration †	805†	05	45
Address window enable	806	06	46
I / O window control	807	07	47
I / O window 0 start-address low-byte	808	08	48
I / O window 0 start-address high-byte	809	09	49
I / O window 0 end-address low-byte	80A	0A	4A
I / O window 0 end-address high-byte	80B	0B	4B
I / O window 1 start-address low-byte	80C	0C	4C
I / O window 1 start-address high-byte	80D	0D	4D
I / O window 1 end-address low-byte	80E	0E	4E
I / O window 1 end-address high-byte	80F	0F	4F
Memory window 0 start-address low-byte	810	10	50
Memory window 0 start-address high-byte	811	11	51
Memory window 0 end-address low-byte	812	12	52
Memory window 0 end-address high-byte	813	13	53
Memory window 0 offset-address low-byte	814	14	54
Memory window 0 offset-address high-byte	815	15	55
Card detect and general control	816	16	56
Reserved	817	17	57
Memory window 1 start-address low-byte	818	18	58
Memory window 1 start-address high-byte	819	19	59
Memory window 1 end-address low-byte	81A	1A	5A
Memory window 1 end-address high-byte	81B	1B	5B
Memory window 1 offset-address low-byte	81C	1C	5C
Memory window 1 offset-address high-byte	81D	1D	5D
Global control	81E	1E	5E
Reserved	81F	1F	5F
Memory window 2 start-address low-byte	820	20	60
Memory window 2 start-address high-byte	821	21	61
Memory window 2 end-address low-byte	822	22	62
Memory window 2 end-address high-byte	823	23	63
Memory window 2 offset-address low-byte	824	24	64
Memory window 2 offset-address high-byte	825	25	65

† One or more bits in this register are cleared only by the assertion of $\overline{\text{GRST}}$ when $\overline{\text{PME}}$ is enabled. If $\overline{\text{PME}}$ is NOT enabled, then this bit is cleared by the assertion of $\overline{\text{PRST}}$ or $\overline{\text{GRST}}$.

Table 5–1. ExCA Registers and Offsets (continued)

REGISTER NAME	PCI MEMORY ADDRESS OFFSET	EXCA OFFSET (CARD A)	EXCA OFFSET (CARD B)
Reserved	826	26	66
Reserved	827	27	67
Memory window 3 start-address low-byte	828	28	68
Memory window 3 start-address high-byte	829	29	69
Memory window 3 end-address low-byte	82A	2A	6A
Memory window 3 end-address high-byte	82B	2B	6B
Memory window 3 offset-address low-byte	82C	2C	6C
Memory window 3 offset-address high-byte	82D	2D	6D
Reserved	82E	2E	6E
Reserved	82F	2F	6F
Memory window 4 start-address low-byte	830	30	70
Memory window 4 start-address high-byte	831	31	71
Memory window 4 end-address low-byte	832	32	72
Memory window 4 end-address high-byte	833	33	73
Memory window 4 offset-address low-byte	834	34	74
Memory window 4 offset-address high-byte	835	35	75
I/O window 0 offset-address low-byte	836	36	76
I/O window 0 offset-address high-byte	837	37	77
I/O window 1 offset-address low-byte	838	38	78
I/O window 1 offset-address high-byte	839	39	79
Reserved	83A	3A	7A
Reserved	83B	3B	7B
Reserved	83C	3C	7C
Reserved	83D	3D	7D
Reserved	83E	3E	7E
Reserved	83F	3F	7F
Memory window page register 0	840	-	-
Memory window page register 1	841	-	-
Memory window page register 2	842	-	-
Memory window page register 3	843	-	-
Memory window page register 4	844	-	-

5.1 ExCA Identification and Revision Register (Index 00h)

This register provides host software with information on 16-bit PC Card support and 82365SL-DF compatibility.

NOTE: If bit 5 (SUBSYRW) in the system control register is 1, then this register is read-only.

Bit	7	6	5	4	3	2	1	0
Name	ExCA identification and revision							
Type	R/W	R/W	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/Write
Offset: CardBus Socket Address + 800h: Card A ExCA Offset 00h
Card B ExCA Offset 40h
Default: 84h

Table 5–2. ExCA Identification and Revision Register Description

BIT	TYPE	FUNCTION
7–6	R/W	IFTYPE. Interface type. These bits, which are hardwired as 10b, identify the 16-bit PC Card support provided by the PCI4450. The PCI4450 supports both I/O and memory 16-bit PC Cards.
5–4	R/W	Reserved. These bits can be used for 82365SL emulation.
3–0	R/W	365REV. 82365SL revision. This field stores the 82365SL revision supported by the PCI4450. Host software may read this field to determine compatibility to the 82365SL register set. This field defaults to 0100b upon reset.

5.2 ExCA Interface Status Register (Index 01h)

This register provides information on current status of the PC Card interface. An x in the default bit values indicates that the value of the bit after reset depends on the state of the PC Card interface.

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: Card A ExCA Offset 01h
Card B ExCA Offset 41h
Default: 00XX XXXXb

Table 5–3. ExCA Interface Status Register Description

BIT	TYPE	FUNCTION
7	R	Reserved. This bit returns 0 when read. A write has no effect.
6	R	CARDPWR. Card power. This bit indicates the current power status of the PC Card socket. This bit reflects how the ExCA power control register has been programmed. The bit is encoded as: 0 = V_{CC} and V_{PP} to the socket is turned off (default). 1 = V_{CC} and V_{PP} to the socket is turned on.
5	R	READY. This bit indicates the current status of the READY signal at the PC Card interface. 0 = PC Card is not ready for a data transfer. 1 = PC Card is ready for a data transfer.
4	R	CARDWP. Card write protect. This bit indicates the current status of the WP signal at the PC Card interface. This signal reports to the PCI4450 whether or not the memory card is write protected. Further, write protection for an entire PCI4450 16-bit memory window is available by setting the appropriate bit in the ExCA memory window offset-address high byte register. 0 = WP signal is 0. PC Card is R/W. 1 = WP signal is 1. PC Card is read-only.
3	R	CDETECT2. Card detect 2. This bit indicates the status of the CD2 signal at the PC Card interface. Software may use this and CDETECT1 to determine if a PC Card is fully seated in the socket. 0 = CD2 signal is 1. No PC Card inserted. 1 = CD2 signal is 0. PC Card at least partially inserted.
2	R	CDETECT1. Card detect 1. This bit indicates the status of the CD1 signal at the PC Card interface. Software may use this and CDETECT2 to determine if a PC Card is fully seated in the socket. 0 = CD1 signal is 1. No PC Card inserted. 1 = CD1 signal is 0. PC Card at least partially inserted.
1–0	R	BVDSTAT. 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 0 reflects the BVD1 status, and bit 1 reflects BVD2. 00 = Battery is dead. 01 = Battery is dead. 10 = Battery is low; warning. 11 = Battery is good. When a 16-bit I/O card is inserted, this field indicates the status of the $\overline{\text{SPKR}}$ (bit 1) signal and the $\overline{\text{STSCHG}}$ (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 (Index 02h)

This register provides PC Card power control. Bit 7 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.

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: Card A ExCA Offset 02h
Card B ExCA Offset 42h
Default: 00h

Table 5–4. ExCA Power Control Register Description

BIT	TYPE	FUNCTION
7†	R/W	COE. Card output enable. This bit controls the state of all of the 16-bit outputs on the PCI4450. This bit is encoded as: 0 = 16-bit PC Card outputs are disabled (default). 1 = 16-bit PC Card outputs are enabled.
6–5	R	Reserved. These bits return 0s when read. Writes have no effect.
4–3†	R/W	EXCAVCC. V_{CC}. These bits 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	R	Reserved. This bit returns 0 when read. A write has no effect.
1–0†	R/W	EXCAVPP. V_{PP}. These bits are used to request changes to card V_{PP}. The PCI4450 ignores this field unless V_{CC} to the socket is enabled (i.e., 5 Vdc or 3.3 Vdc). This field is encoded as: 00 = 0 V (default) 01 = V_{CC} 10 = 12 V 11 = 0 V Reserved

† This bit is cleared only by the assertion of $\overline{\text{GRST}}$ when $\overline{\text{PME}}$ is enabled. If $\overline{\text{PME}}$ is not enabled, then this bit is cleared by the assertion of $\overline{\text{PRST}}$ or $\overline{\text{GRST}}$.

5.4 ExCA Interrupt and General Control Register (Index 03h)

This register controls interrupt routing for I/O interrupts as well as other critical 16-bit PC Card functions.

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: Card A ExCA Offset 03h
Card B ExCA Offset 43h
Default: 00h

Table 5–5. ExCA Interrupt and General Control Register Description

BIT	TYPE	FUNCTION
7	R/W	RINGEN. Card ring indicate enable. Enables the ring indicate function of the BVD1/RI pins. This bit is encoded as: 0 = Ring indicate disabled (default) 1 = Ring indicate enabled
6†	R/W	Card reset. This bit controls the 16-bit PC Card RESET signal, and allows host software to force a card reset. This bit affects 16-bit cards only. This bit is encoded as: 0 = RESET signal asserted (default) 1 = RESET signal deasserted.
5†	R/W	CARDTYPE. Card type. This bit indicates the PC Card type. This bit is encoded as: 0 = Memory PC Card is installed (default) 1 = I/O PC Card is installed
4	R/W	CSCROUTE. PCI interrupt – CSC routing enable bit. This bit has meaning only if the CSC interrupt routing control bit (PCI offset 93h, bit 5) is 0b. In this case, when this bit 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 in the ExCA card status change interrupt configuration register. This bit is encoded as: 0 = CSC interrupts routed by ExCA registers (default) 1 = CSC interrupts routed to PCI interrupts If the CSC interrupt routing control bit (PCI offset 93h, bit 5) is set to 1b, this bit has no meaning which is the default case.
3–0	R/W	INTSELECT. Card interrupt select for I/O PC Card functional interrupts. These bits select the interrupt routing for I/O PC Card functional interrupts. This field is encoded as: 0000 = No ISA interrupt routing (default). CSC interrupts routed to PCI Interrupts. 0001 = IRQ1 enabled 0010 = SMI enabled 0011 = IRQ3 enabled 0100 = IRQ4 enabled 0101 = IRQ5 enabled 0110 = 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

† This bit is cleared only by the assertion of $\overline{\text{GRST}}$ when $\overline{\text{PME}}$ is enabled. If $\overline{\text{PME}}$ is not enabled, then this bit is cleared by the assertion of $\overline{\text{PRST}}$ or $\overline{\text{GRST}}$.

5.5 ExCA Card Status-Change Register (Index 04h)

This register reflects the status of PC Card CSC interrupt sources. The ExCA card status change interrupt configuration 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 as 0. When an interrupt source is enabled and that particular event occurs, the corresponding bit in this register is set to indicate the interrupt source. 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 write back of 1 to the status bit. The choice of these two methods is based on the interrupt flag clear mode select, bit 2, in the ExCA global control register.

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: Card A ExCA Offset 04h
Card B ExCA Offset 44h

Default: 00h

Table 5–6. ExCA Card Status-Change Register Description

BIT	TYPE	FUNCTION
7–4	R	Reserved. These bits return 0s when read. Writes have no effect.
3†	R	CDCHANGE. Card detect change. This bit indicates whether a change on the CD1 or CD2 signals occurred at the PC Card interface. A read of this bit or writing a 1 to this bit clears it. This bit is encoded as: 0 = No change detected on either CD1 or CD2 1 = A change was detected on either CD1 or CD2
2†	R	READYCHANGE. Ready change. When a 16-bit memory is installed in the socket, this bit includes whether the source of a PCI4450 interrupt was due to a change on the READY signal at the PC Card interface indicating that PC Card is now ready to accept new data. A read of this bit or writing a 1 to this bit clears it. This bit is encoded as: 0 = No low-to-high transition detected on READY (default) 1 = Detected a low-to-high transition on READY When a 16-bit I/O card is installed, this bit is always 0.
1†	R	BATWARN. Battery warning change. When a 16-bit memory card is installed in the socket, this bit indicates whether the source of a PCI4450 interrupt was due to a battery low warning condition. A read of this bit or writing a 1 to this bit clears it. This bit is encoded as: 0 = No battery warning condition (default) 1 = Detected a battery warning condition When a 16-bit I/O card is installed, this bit is always 0.
0†	R	BATDEAD. Battery dead or status change. When a 16-bit memory card is installed in the socket, this bit indicates whether the source of a PCI4450 interrupt was due to a battery dead condition. A read of this bit or writing a 1 to this bit clears it. This bit is encoded as: 0 = STSCHG deasserted (default) 1 = STSCHG asserted Ring indicate. When the PCI4450 is configured for ring indicate operation this bit indicates the status of the RI pin.

† This bit is cleared only by the assertion of $\overline{\text{GRST}}$ when $\overline{\text{PME}}$ is enabled. If $\overline{\text{PME}}$ is not enabled, then this bit is cleared by the assertion of $\overline{\text{PRST}}$ or $\overline{\text{GRST}}$.

5.6 ExCA Card Status-Change Interrupt Configuration Register (Index 05h)

This register controls interrupt routing for CSC interrupts, as well as masks/unmasks CSC interrupt sources.

Bit	7	6	5	4	3†	2†	1†	0†
Name	ExCA card 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: Card A ExCA Offset 05h
Card B ExCA Offset 45h
Default: 00h

Table 5–7. ExCA Card Status-Change Interrupt Register Description

BIT	TYPE	FUNCTION
7–4	R/W	CSCSELECT. Interrupt select for card status change. These bits select the interrupt routing for card status change interrupts. This field is encoded as: 0000 = CSC interrupts routed to PCI interrupts if bit 5 of the diagnostic register (PCI offset 93h) is set to 1b. In this case bit 4 of ExCA 803 is a “don’t care.” This is the default setting. 0000 = No ISA interrupt routing if bit 5 of the diagnostic register (PCI offset 93h) is set to 0b. In this case, CSC interrupts are routed to PCI interrupts by setting bit 4 of ExCA 803h to 1b 0001 = IRQ1 enabled 0010 = SMI enabled 0011 = IRQ3 enabled 0100 = IRQ4 enabled 0101 = IRQ5 enabled 0110 = 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
3†	R/W	CDEN. Card detect enable. Enables interrupts on CD1 or CD2 changes. This bit is encoded as: 0 = Disables interrupts on CD1 or CD2 line changes (default) 1 = Enable interrupts on CD1 or CD2 line changes
2†	R/W	READYEN. Ready enable. This bit enables/disables a low-to-high transition on the PC Card READY signal 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†	R/W	BATWARNEN. Battery warning enable. This bit 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†	R/W	BATDEADEN. Battery dead enable. This bit enables/disables a battery dead condition on a memory PC Card or assertion of the STSCHG I/O PC Card signal to generate a CSC interrupt. 0 = Disables host interrupt generation (default) 1 = Enables host interrupt generation

† This bit is cleared only by the assertion of $\overline{\text{GRST}}$ when $\overline{\text{PME}}$ is enabled. If $\overline{\text{PME}}$ is not enabled, then this bit is cleared by the assertion of $\overline{\text{PRST}}$ or $\overline{\text{GRST}}$.

5.7 ExCA Address Window Enable Register (Index 06h)

This 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 PCI4450 will 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 ExCA memory and I/O window start/end/offset address registers.

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: Card A ExCA Offset 06h
Card B ExCA Offset 46h
Default: 00h

Table 5–8. ExCA Address Window Enable Register Description

BIT	TYPE	FUNCTION
7	R/W	IOWIN1EN. I/O window 1 enable. This bit enables/disables I/O window 1 for the card. This bit is encoded as: 0 = I/O window 1 disabled (default) 1 = I/O window 1 enabled
6	R/W	IOWIN0EN. I/O window 0 enable. This bit enables/disables I/O window 0 for the card. This bit is encoded as: 0 = I/O window 0 disabled (default) 1 = I/O window 0 enabled
5	R	Reserved. This bit returns 0 when read. A write has no effect.
4	R/W	MEMWIN4EN. Memory window 4 enable. This bit enables/disables memory window 4 for the card. This bit is encoded as: 0 = memory window 4 disabled (default) 1 = memory window 4 enabled
3	R/W	MEMWIN3EN. Memory window 3 enable. This bit enables/disables memory window 3 for the card. This bit is encoded as: 0 = memory window 3 disabled (default) 1 = memory window 3 enabled
2	R/W	MEMWIN2EN. Memory window 2 enable. This bit enables/disables memory window 2 for the card. This bit is encoded as: 0 = memory window 2 disabled (default) 1 = memory window 2 enable
1	R/W	MEMWIN1EN. Memory window 1 enable. This bit enables/disables memory window 1 for the PC Card. This bit is encoded as: 0 = memory window 1 disabled (default) 1 = memory window 1 enabled
0	R/W	MEMWIN0EN. Memory window 0 enable. This bit enables/disables memory window 0 for the PC Card. This bit is encoded as: 0 = memory window 0 disabled (default) 1 = memory window 0 enabled

5.8 ExCA I/O Window Control Register (Index 07h)

This register contains parameters related to I/O window sizing and cycle timing.

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: Card A ExCA Offset 07h
Card B ExCA Offset 47h
Default: 00h

Table 5–9. ExCA I/O Window Control Register Description

BIT	TYPE	FUNCTION
7	R/W	WAITSTATE1. I/O window 1 wait-state. This bit controls the I/O window 1 wait-state for 16-bit I/O accesses. This bit has no effect on 8-bit accesses. This wait-state timing emulates the ISA wait-state used by the 82365SL-DF. This bit is encoded as: 0 = 16-bit cycles have standard length (default) 1 = 16-bit cycles extended by one equivalent ISA wait state
6	R/W	ZEROWS1. I/O window 1 zero wait-state. This bit controls the I/O window 1 wait-state for 8-bit I/O accesses. NOTE: This bit has no effect on 16-bit accesses. This wait-state timing emulates the ISA wait-state used by the 82365SL-DF. 0 = 8-bit cycles have standard length (default) 1 = 8-bit cycles reduced to equivalent of three ISA cycles
5	R/W	IOSIS16W1. I/O window 1 IOIS16 source. This bit controls the I/O window automatic data sizing feature which used the $\overline{\text{IOIS16}}$ signal from the PC Card to determine the data width of the I/O data transfer. 0 = Data width determined by DATASIZE1, bit 4 (default) 1 = Window data width determined by $\overline{\text{IOIS16}}$
4	R/W	DATASIZE1. I/O window 1 data size. This bit controls the I/O window 1 data size. This bit is ignored if the I/O window 1 IOIS16 source bit (bit 5) is set. This bit is encoded as: 0 = Window data width is 8 bits (default) 1 = Window data width is 16 bits
3	R/W	WAITSTATE0. I/O window 0 wait-state. This bit controls the I/O window 0 wait-state for 16-bit I/O accesses. This bit has no effect on 8-bit accesses. This wait-state timing emulates the ISA wait-state used by the 82365SL-DF. This bit is encoded as: 0 = 16-bit cycles have standard length (default) 1 = 16-bit cycles extended by one equivalent ISA wait state
2	R/W	ZEROWS0. I/O window 0 zero wait-state. This bit controls the I/O window 0 wait-state for 8-bit I/O accesses. NOTE: This bit has no effect on 16-bit accesses. This wait-state timing emulates the ISA wait-state used by the 82365SL-DF. 0 = 8-bit cycles have standard length (default) 1 = 8-bit cycles reduced to equivalent of three ISA cycles
1	R/W	IOIS16W0. I/O window 0 IOIS16 source. This bit controls the I/O window automatic data sizing feature which used the $\overline{\text{IOIS16}}$ signal from the PC Card to determine the data width of the I/O data transfer. 0 = Data width determined by DATASIZE0, bit 0 (default) 1 = Window data width determined by $\overline{\text{IOIS16}}$
0	R/W	DATASIZE0. I/O window 0 data size. This bit controls the I/O window 1 data size. This bit is ignored if the I/O window 1 IOIS16 Source bit (bit 1) 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 & 1 Start-Address Low-Byte Registers (Index 08h, 0Ch)

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 & 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: Card A ExCA Offset 08h
Card B ExCA Offset 48h
Register: **ExCA I/O window 1 start-address low-byte**
Offset: CardBus Socket Address + 80Ch: Card A ExCA Offset 0Ch
Card B ExCA Offset 4Ch
Type: Read/Write
Default: 00h
Size: One byte

5.10 ExCA I/O Windows 0 & 1 Start-Address High-Byte Registers (Index 09h, ODh)

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 & 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: Card A ExCA Offset 09h
Card B ExCA Offset 49h
Register: **ExCA I/O window 1 start-address high-byte**
Offset: CardBus Socket Address + 80Dh: Card A ExCA Offset 0Dh
Card B ExCA Offset 4Dh
Type: Read/Write
Default: 00h
Size: One byte

5.11 ExCA I/O Windows 0 & 1 End-Address Low-Byte Registers (Index 0Ah, 0Eh)

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 start address.

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O windows 0 & 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: Card A ExCA Offset 0Ah
Card B ExCA Offset 4Ah
Register: **ExCA I/O window 1 end-address low-byte**
Offset: CardBus Socket Address + 80Eh: Card A ExCA Offset 0Eh
Card B ExCA Offset 4Eh
Type: Read/Write
Default: 00h
Size: One byte

5.12 ExCA I/O Windows 0 & 1 End-Address High-Byte Registers (Index 0Bh, 0Fh)

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

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O windows 0 & 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: Card A ExCA Offset 0Bh
Card B ExCA Offset 4Bh
Register: **ExCA I/O window 1 end-address high-byte**
Offset: CardBus Socket Address + 80Fh: Card A ExCA Offset 0Fh
Card B ExCA Offset 4Fh
Type: Read/Write
Default: 00h
Size: One byte

5.13 ExCA Memory Windows 0–4 Start-Address Low-Byte Registers (Index 10h/18h/20h/28h/30h)

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: Card A ExCA Offset 10h
Card B ExCA Offset 50h

Register: **ExCA memory window 1 start-address low-byte**
Offset: CardBus Socket Address + 818h: Card A ExCA Offset 18h
Card B ExCA Offset 58h

Register: **ExCA memory window 2 start-address low-byte**
Offset: CardBus Socket Address + 820h: Card A ExCA Offset 20h
Card B ExCA Offset 60h

Register: **ExCA memory window 3 start-address low-byte**
Offset: CardBus Socket Address + 828h: Card A ExCA Offset 28h
Card B ExCA Offset 68h

Register: **ExCA memory window 4 start-address low-byte**
Offset: CardBus Socket Address + 830h: Card A ExCA Offset 30h
Card B ExCA Offset 70h

Type: Read/Write
Default: 00h
Size: One byte

5.14 ExCA Memory Windows 0–4 Start-Address High-Byte Registers (Index 11h/19h/21h/29h/31h)

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.

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: Card A ExCA Offset 11h
Card B ExCA Offset 51h

Register: **ExCA memory window 1 start-address high-byte**
Offset: CardBus Socket Address + 819h: Card A ExCA Offset 19h
Card B ExCA Offset 59h

Register: **ExCA memory window 2 start-address high-byte**
Offset: CardBus Socket Address + 821h: Card A ExCA Offset 21h
Card B ExCA Offset 61h

Register: **ExCA memory window 3 start-address high-byte**
Offset: CardBus Socket Address + 829h: Card A ExCA Offset 29h
Card B ExCA Offset 69h

Register: **ExCA memory window 4 start-address high-byte**
Offset: CardBus Socket Address + 831h: Card A ExCA Offset 31h
Card B ExCA Offset 71h

Type: Read/Write
Default: 00h
Size: One byte

Table 5–10. ExCA Memory Windows 0–4 Start-Address High-Byte Registers Description

BIT	TYPE	FUNCTION
7	R/W	DATASIZE. This bit 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	R/W	ZEROWAIT. Zero wait-state. This bit controls the memory window wait state for 8- and 16-bit accesses. This wait state timing emulates the ISA wait-state used by the 82365SL-DF. This bit is encoded as: 0 = 8- and 16-bit cycles have standard length (default) 1 = 8-bit cycles reduced to equivalent of three ISA cycles 16-bit cycles reduce to the equivalent of two ISA cycles.
5–4	R/W	SCRATCH. Scratch pad bits. These bits have no effect on memory window operation.
3–0	R/W	STAHN. Start address high-nibble. These bits 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 (Index 12h/1Ah/22h/2Ah/32h)

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: Card A ExCA Offset 12h
Card B ExCA Offset 52h

Register: **ExCA memory window 1 end-address low-byte**
Offset: CardBus Socket Address + 81Ah: Card A ExCA Offset 1Ah
Card B ExCA Offset 5Ah

Register: **ExCA memory window 2 end-address low-byte**
Offset: CardBus Socket Address + 822h: Card A ExCA Offset 22h
Card B ExCA Offset 62h

Register: **ExCA memory window 3 end-address low-byte**
Offset: CardBus Socket Address + 82Ah: Card A ExCA Offset 2Ah
Card B ExCA Offset 68h

Register: **ExCA memory window 4 end-address low-byte**
Offset: CardBus Socket Address + 832h: Card A ExCA Offset 32h
Card B ExCA Offset 72h

Type: Read/Write
Default: 00h
Size: One byte

5.16 ExCA Memory Windows 0–4 End-Address High-Byte Registers (Index 13h/1Bh/23h/2Bh/33h)

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.

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: Card A ExCA Offset 13h
Card B ExCA Offset 53h
Register: **ExCA memory window 1 end-address high-byte**
Offset: CardBus Socket Address + 81Bh: Card A ExCA Offset 1Bh
Card B ExCA Offset 5Bh
Register: **ExCA memory window 2 end-address high-byte**
Offset: CardBus Socket Address + 823h: Card A ExCA Offset 23h
Card B ExCA Offset 63h
Register: **ExCA memory window 3 end-address high-byte**
Offset: CardBus Socket Address + 82Bh: Card A ExCA Offset 2Bh
Card B ExCA Offset 6Bh
Register: **ExCA Memory window 4 end-address high-byte**
Offset: CardBus Socket Address + 833h: Card A ExCA Offset 33h
Card B ExCA Offset 73h

Type: Read/Write
Default: 00h
Size: One byte

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

BIT	TYPE	FUNCTION
7–6	R/W	MEMWS. Wait state. These bits 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	R	Reserved. These bits return 0s when read. Writes have no effect.
3–0	R/W	ENDHN. End address high-nibble. These bits represent the upper address bits A23–A20 of the memory window and address.

5.17 ExCA Memory Windows 0–4 Offset-Address Low-Byte Registers (Index 14h/1Ch/24h/2Ch/34h)

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: Card A ExCA Offset 14h
Card B ExCA Offset 54h

Register: **ExCA memory window 1 offset-address low-byte**
Offset: CardBus Socket Address + 81Ch: Card A ExCA Offset 1Ch
Card B ExCA Offset 5Ch

Register: **ExCA memory window 2 offset-address low-byte**
Offset: CardBus Socket Address + 824h: Card A ExCA Offset 24h
Card B ExCA Offset 64h

Register: **ExCA memory window 3 offset-address low-byte**
Offset: CardBus Socket Address + 82Ch: Card A ExCA Offset 2Ch
Card B ExCA Offset 6Ch

Register: **ExCA memory window 4 offset-address low-byte**
Offset: CardBus Socket Address + 834h: Card A ExCA Offset 34h
Card B ExCA Offset 74h

Type: Read/Write
Default: 00h
Size: One byte

5.18 ExCA Memory Windows 0–4 Offset-Address High-Byte Registers (Index 15h/1Dh/25h/2Dh/35h)

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.

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory window 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: Card A ExCA Offset 15h
Card B ExCA Offset 55h

Register: **ExCA memory window 1 offset-address high-byte**
Offset: CardBus Socket Address + 81Dh: Card A ExCA Offset 1Dh
Card B ExCA Offset 5Dh

Register: **ExCA memory window 2 offset-address high-byte**
Offset: CardBus Socket Address + 825h: Card A ExCA Offset 25h
Card B ExCA Offset 65h

Register: **ExCA memory window 3 offset-address high-byte**
Offset: CardBus Socket Address + 82Dh: Card A ExCA Offset 2Dh
Card B ExCA Offset 6Dh

Register: **ExCA memory window 4 offset-address high-byte**
Offset: CardBus Socket Address + 835h: Card A ExCA Offset 35h
Card B ExCA Offset 75h

Type: Read/Write
Default: 00h
Size: One byte

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

BIT	TYPE	FUNCTION
7	R/W	WINWP. Write protect. This bit 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	R/W	REG. This bit 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 attribute memory
5–0	R/W	OFFHB. Offset address high-byte. These bits represent the upper address bits A25–A20 of the memory window offset address.

5.19 ExCA I/O Windows 0 & 1 Offset-Address Low-Byte Registers (Index 36h, 38h)

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

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O windows 0 & 1 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 I/O window 0 offset-address low-byte**
Offset: CardBus Socket Address + 836h: Card A ExCA Offset 36h
Card B ExCA Offset 76h
Register: **ExCA I/O window 1 offset-address low-byte**
Offset: CardBus Socket Address + 838h: Card A ExCA Offset 38h
Card B ExCA Offset 78h
Type: Read/Write
Default: 00h
Size: One byte

5.20 ExCA I/O Windows 0 & 1 Offset-Address High-Byte Registers (Index 37h, 39h)

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 & 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: Card A ExCA Offset 37h
Card B ExCA Offset 77h
Register: **ExCA I/O window 1 offset-address high-byte**
Offset: CardBus Socket Address + 839h: Card A ExCA Offset 39h
Card B ExCA Offset 79h
Type: Read/Write
Default: 00h
Size: One byte

5.21 ExCA Card Detect and General Control Register (Index 16h)

This register controls how the ExCA registers for the socket respond to card removal. It also reports the status of the $\overline{VS1}$ and $\overline{VS2}$ signals at the PC Card interface. Table 5–13 describes each bit in the ExCA card detect and general control register.

Bit	7	6	5	4	3	2	1	0
Name	ExCA card detect and general control							
Type	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, Write-only, Read/Write
Offset: CardBus Socket Address + 816h: Card A ExCA Offset 16h
Card B ExCA Offset 56h
Default: XX00 0000b

Table 5–13. ExCA Card Detect and General Control Register Description

BIT	TYPE	FUNCTION
7	R	VS2STAT. VS2. This bit reports the current state of the VS2 signal at the PC Card interface, and, therefore, does not have a default value. 0 = VS2 is low 1 = VS2 is high
6	R	VS1STAT. VS1. This bit reports the current state of the VS1 signal at the PC Card interface, and, therefore, does not have a default value. 0 = VS1 is low 1 = VS1 is high
5	W	SWCSC. Software card detect interrupt. If the card detect enable bit in the ExCA card status change interrupt configuration register is set, then writing a 1 to this bit causes a card detect card status change interrupt for the associated card socket. If the card detect enable bit is cleared to 0 in the ExCA card status change interrupt configuration register, then writing a 1 to the software card detect interrupt bit has no effect. This bit is write-only. A read operation of this bit always returns 0. Writing a 1 to this bit also clears it. If bit 2 of the ExCA global control register is set and a 1 is written to clear bit 3 of the ExCA card status change interrupt register, then this bit also gets cleared.
4	R/W	CDRESUME. Card detect resume enable. If this bit is set to 1 and once a card detect change has been detected on the CD1 and CD2 inputs, then the $\overline{RI\_OUT}$ output will go from high to low. The $\overline{RI\_OUT}$ remains low until the card status change bit in the ExCA card status change register is cleared. If this bit is a 0, then the card detect resume functionality is disabled. 0 = Card detect resume disabled (default) 1 = Card detect resume enabled
3–2	R	Reserved. These bits return 0s when read. Writes have no effect.
1	R/W	REGCONFIG. Register configuration upon card removal. This bit 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 upon card removal (default) 1 = Reset ExCA registers upon card removal
0	R	Reserved. This bit returns 0 when read. A write has no effect.

5.22 ExCA Global Control Register (Index 1Eh)

This register controls both PC Card sockets, and is not duplicated for each socket. The host interrupt mode bits in this register are retained for 82365SL compatibility.

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: Card A ExCA Offset 1Eh
Card B ExCA Offset 5Eh
Default: 00h

Table 5–14. ExCA Global Control Register Description

BIT	TYPE	FUNCTION
7–5	R	Reserved. These bits return 0s when read. Writes have no effect.
4	R/W	INTMODEB. Level/edge interrupt mode select – card B. This bit selects the signaling mode for the PCI4450 host interrupt for Card B interrupts. This bit is encoded as: 0 = Host interrupt is edge mode (default) 1 = Host interrupt is level mode
3	R/W	INTMODEA. Level/edge interrupt mode select – card A. This bit selects the signaling mode for the PCI4450 host interrupt for card A interrupts. This bit is encoded as: 0 = Host interrupt is edge mode (default) 1 = Host interrupt is level mode
2	R/W	IFCMODE. Interrupt flag clear mode select. This bit selects the interrupt flag clear mechanism for the flags in the ExCA card status change register. This bit is encoded as: 0 = Interrupt flags cleared by read of CSC register (default) 1 = Interrupt flags cleared by explicit write back of 1
1	R/W	CSCMODE. Card status change level/edge mode select. This bit selects the signaling mode for the PCI4450 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	R/W	PWRDWN. PWRDWN mode select. When the bit is set to 1, the PCI4450 is in power-down mode. In power-down mode the PCI4450 card outputs are placed in a high-impedance state until an active cycle is executed on the card interface. Following an active cycle the outputs are again placed in a high-impedance state. The PCI4450 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 disabled (default) 1 = Power-down mode enabled

5.23 ExCA Memory Windows 0–4 Page Registers (Index 40h, 41h, 42h, 43h, 44h)

The upper 8 bits of a 4-byte PCI memory address are compared to the contents of this register when decoding addresses for 16-bit memory windows. Each window has its own page register, all of which default to 00h. By programming this register to a nonzero value, host software may locate 16-bit memory windows in any one of 256 16M-byte regions in the 4 Gigabyte PCI address space. These registers are only accessible when the ExCA registers are memory-mapped, that is, these registers may not 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 (Functions 0 and 1)

The PCMCIA CardBus Specification requires a CardBus socket controller to provide five 32-bit registers which report and control the socket-specific functions. The PCI4450 provides the CardBus socket/ExCA base address register (PCI offset 10h) 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 below. Table 6–1 illustrates the location of the socket registers in relation to the CardBus socket/ExCA base address.

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

† One or more bits in this register are cleared only by the assertion of $\overline{\text{GRST}}$ when $\overline{\text{PME}}$ is enabled. If $\overline{\text{PME}}$ is not enabled, then this bit is cleared by the assertion of $\overline{\text{PRST}}$ or $\overline{\text{GRST}}$.

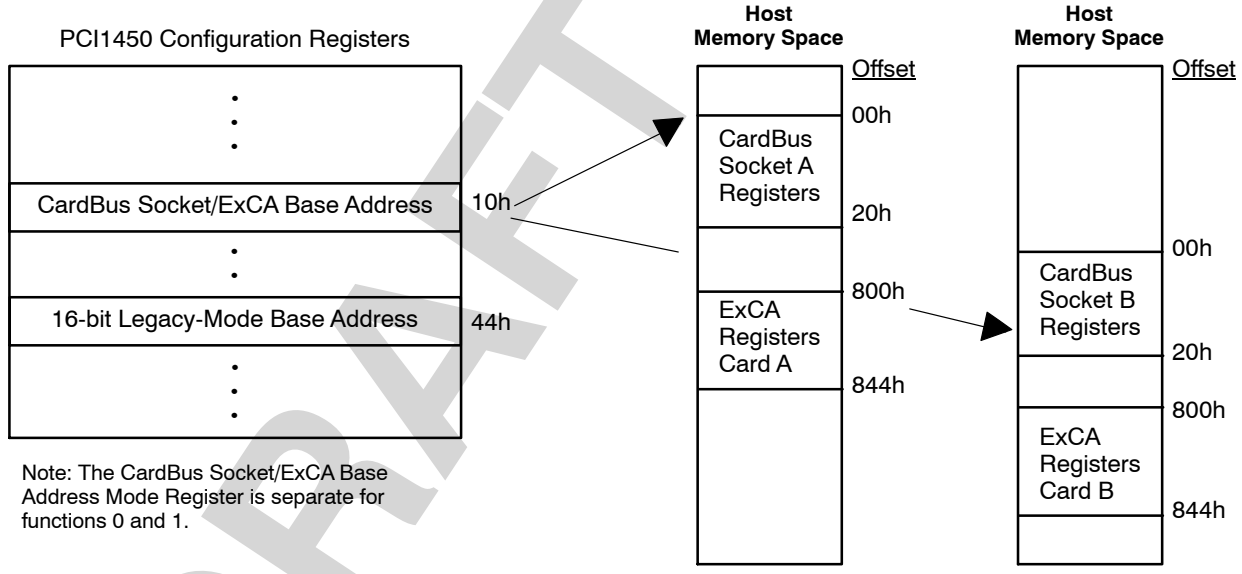


Figure 6–1. Accessing CardBus Socket Registers Through PCI Memory

6.1 Socket Event Register

This 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 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 through writing a 1 to the corresponding bit in the socket force event register. All bits in this register are cleared by PCI reset. They may be immediately set again, if, when coming out of PC Card reset, the bridge finds the status unchanged (i.e., CSTSCHG reasserted or card detect is still true). Software needs to clear this register before enabling interrupts. If it is not cleared and interrupts are enabled, then an interrupt is generated based on any bit set and not masked.

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/WC	R/WC	R/WC	R/WC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Table 6–2. Socket Event Register Description

BIT	TYPE	FUNCTION
31–4	R	Reserved. These bits return 0s when read.
3†	R/WC	PWREVENT. Power cycle. This bit is set when the PCI4450 detects that the PWRCYCLE bit in the socket present state register has changed. This bit is cleared by writing a 1.
2†	R/WC	CD2EVENT. $\overline{\text{CCD2}}$. This bit is set when the PCI4450 detects that the CDETECT2 field in the socket present state register has changed. This bit is cleared by writing a 1.
1†	R/WC	CD1EVENT. $\overline{\text{CCD1}}$. This bit is set when the PCI4450 detects that the CDETECT1 field in the socket present state register has changed. This bit is cleared by writing a 1.
0†	R/WC	CSTSEVENT. CSTSCHG. This bit is set when the CARDSTS field in the socket present state register has changed state. For CardBus cards, this bit is set on the rising edge of the CSTSCHG signal. For 16-bit PC Cards, this bit is set on both transitions of the CSTSCHG signal. This bit is reset by writing a 1.

† This bit is cleared only by the assertion of $\overline{\text{GRST}}$ when $\overline{\text{PME}}$ is enabled. If $\overline{\text{PME}}$ is not enabled, then this bit is cleared by the assertion of $\overline{\text{PRST}}$ or $\overline{\text{GRST}}$.

6.2 Socket Mask Register

This register allows software to control the CardBus card events which generate a status change interrupt. Table 6–3 below describes each bit in this register. The state of these mask bits does not prevent the corresponding bits from reacting in the socket event register.

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	TYPE	FUNCTION
31–4	R	Reserved. These bits return 0s when read.
3†	R/W	PWRMASK. Power cycle. This bit masks the PWRCYCLE bit in the socket present state register from causing a status change interrupt. 0 = PWRCYCLE event will not cause a CSC interrupt (default) 1 = PWRCYCLE event will cause a CSC interrupt
2–1†	R/W	CDMASK. Card detect mask. These bits mask the CDETECT1 and CDETECT2 bits in the socket present state register from causing a CSC interrupt. 00 = Insertion/removal will not cause CSC interrupt (default) 01 = Reserved (undefined) 10 = Reserved (undefined) 11 = Insertion/removal will cause CSC interrupt
0†	R/W	CSTSMASK. CSTSCHG mask. This bit masks the CARDSTS field in the socket present state register from causing a CSC interrupt. 0 = CARDSTS event will not cause CSC interrupt (default) 1 = CARDSTS event will cause CSC interrupt

† This bit is cleared only by the assertion of $\overline{\text{GRST}}$ when $\overline{\text{PME}}$ is enabled. If $\overline{\text{PME}}$ is not enabled, then this bit is cleared by the assertion of $\overline{\text{PRST}}$ or $\overline{\text{GRST}}$.

6.3 Socket Present State Register

This register reports information about the socket interface. Writes to the socket force event register are reflected here as well as general socket interface status. Information about PC Card V_{CC} support and card type is only updated at each insertion. Also note that the PCI4450 uses the $\overline{CCD1}$ and $\overline{CCD2}$ signals during card identification, and changes on these signals during this operation are not reflected in this register.

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	TYPE	FUNCTION
31	R	YVSOCKET. YV socket. This bit indicates whether or not the socket can supply V_{CC} = Y.YV to PC Cards. The PCI4450 does not support Y.YV V_{CC}; therefore, this bit is always reset unless overridden by the socket force event register. This bit is hardwired to 0.
30	R	XVSOCKET. XV socket. This bit indicates whether or not the socket can supply V_{CC} = X.XV to PC Cards. The PCI4450 does not support X.XV V_{CC}; therefore, this bit is always reset unless overridden by the socket force event register. This bit is hardwired to 0.
29	R	3VSOCKET. 3-V socket. This bit indicates whether or not the socket can supply V_{CC} = 3.3 Vdc to PC Cards. The PCI4450 does support 3.3 V V_{CC}; therefore, this bit is always set unless overridden by the socket force event register.
28	R	5VSOCKET. 5-V socket. This bit indicates whether or not the socket can supply V_{CC} = 5.0 Vdc to PC Cards. The PCI4450 does support 5.0 V V_{CC}; therefore, this bit is always 1 unless overridden by the device control register (bit 6).
27–14	R	Reserved. These bits return 0s when read.
13	R	YVCARD. YV card. This bit indicates whether or not the PC Card inserted in the socket supports V_{CC} = Y.Y Vdc. This bit can be set by writing to the corresponding bit in the socket force event register.
12	R	XVCARD. XV card. This bit indicates whether or not the PC Card inserted in the socket supports V_{CC} = X.X Vdc. This bit can be set by writing to the corresponding bit in the socket force event register.
11	R	3VCARD. 3-V card. This bit indicates whether or not the PC Card inserted in the socket supports V_{CC} = 3.3 Vdc. This bit can be set by writing to the corresponding bit in the socket force event register.
10	R	5VCARD. 5-V card. This bit indicates whether or not the PC Card inserted in the socket supports V_{CC} = 5.0 Vdc.
9	R	BADVCCREQ. Bad V_{CC} request. This bit 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	R	DATALOST. Data lost. This bit 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 PCI4450. 0 = Normal operation (default) 1 = Potential data loss due to card removal
7	R	NOTACARD. Not a card. This bit indicates that an unrecognizable PC Card has been inserted in the socket. This bit is not updated until a valid PC Card is inserted into the socket. 0 = Normal operation (default) 1 = Unrecognizable PC Card detected

Table 6–4. Socket Present State Register Description (continued)

BIT	TYPE	FUNCTION

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

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

6.4 Socket Force Event Register

This register is used to force changes to the socket event register and the socket present state register. The CVSTEST bit in this register must be written when forcing changes that require card interrogation.

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 XXXXh

Table 6–5. Socket Force Event Register Description

BIT	TYPE	FUNCTION
31–15	R	Reserved. These bits return 0s when read.
14	W	CVSTEST. Card VS test. When this bit is set, the PCI4450 reinterrogates the PC Card, updates the socket present state register, and re-enables the socket power control.
13	W	FYVCARD. Force YV card. Writes to this bit cause the YVCARD bit in the socket present state register to be written. When set, this bit disables the socket power control.
12	W	FXVCARD. Force XV card. Writes to this bit cause the XVCARD bit in the socket present state register to be written. When set, this bit disables the socket power control.
11	W	F3VCARD. Force 3-V card. Writes to this bit cause the 3VCARD bit in the socket present state register to be written. When set, this bit disables the socket power control.
10	W	F5VCARD. Force 5-V card. Writes to this bit cause the 5VCARD bit in the socket present state register to be written. When set, this bit disables the socket power control.
9	W	FBADVCCREQ. Force BadVccReq. Changes to the BADVCCREQ bit in the socket present state register can be made by writing this bit.
8	W	FDATALOST. Force data lost. Writes to this bit cause the DATALOST bit in the socket present state register to be written.
7	W	FNOTACARD. Force not a card. Writes to this bit cause the NOTACARD bit in the socket present state register to be written.
6	R	Reserved. This bit returns 0 when read.
5	W	FCBCARD. Force CardBus card. Writes to this bit cause the CBCARD bit in the socket present state register to be written.
4	W	F16BITCARD. Force 16-bit card. Writes to this bit cause the 16BITCARD bit in the socket present state register to be written.
3	W	FPWRCYCLE. Force power cycle. Writes to this bit cause the PWREVENT bit in the socket event register to be written, and the PWRCYCLE bit in the socket present state register is unaffected.
2	W	FCDETECT2. Force $\overline{\text{CCD2}}$. Writes to this bit cause the CD2EVENT bit in the socket event register to be written, and the CDETECT2 bit in the socket present state register is unaffected.
1	W	FCDETECT1. Force $\overline{\text{CCD1}}$. Writes to this bit cause the CD1EVENT bit in the socket event register to be written, and the CDETECT1 bit in the socket present state register is unaffected.
0	W	FCARDSTS. Force CSTSCHG. Writes to this bit cause the CSTSEVENT bit in the socket event register to be written. The CARDSTS bit in the socket present state register is unaffected.

6.5 Socket Control Register

This register provides control of the voltages applied to the socket's V_{PP} and V_{CC}. The PCI4450 ensures that the socket is powered up only at acceptable voltages when a CardBus card is inserted.

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	TYPE	FUNCTION
31–8	R	Reserved. These bits return 0s when read.
7	R/W	STOPCLK. This bit controls how the CardBus clock run state machine decides when to stop the CardBus clock to the CardBus Card: 0 = The PCI4450 Clock run master will try to stop the clock to the CardBus Card under the following two conditions: The CardBus interface is idle for 8 clocks and There is a request from the PCI master to stop the PCI clock. 1 = The PCI4450 clock run master will try to stop the clock to the CardBus card under the following condition: The CardBus interface is idle for 8 clocks. In summary, if this bit is set to1, then the CardBus controller will try to stop the clock to the CardBus card independent of the PCI clock run signal. The only condition that has to be satisfied in this case is the CardBus interface sampled idle for 8 clocks.
6–4†	R/W	VCCCTRL. V_{CC} control. These bits are used to request card V_{CC} changes. 000 = Request power off (default) 001 = Reserved 010 = Request V_{CC} = 5.0 V 011 = Request V_{CC} = 3.3 V 100 = Request V_{CC} = X.XV 101 = Request V_{CC} = Y.YV 110 = Reserved 111 = Reserved
3	R	Reserved. This bit returns 0 when read.
2–0†	R/W	VPPCTRL. V_{PP} control. These bits are used to request card VPP changes. 000 = Request power off (default) 001 = Request V_{PP} = 12.0 V 010 = Request V_{PP} = 5.0 V 011 = Request V_{PP} = 3.3 V 100 = Request V_{PP} = X.XV 101 = Request V_{PP} = Y.YV 110 = Reserved 111 = Reserved

† This bit is cleared only by the assertion of $\overline{\text{GRST}}$ when $\overline{\text{PME}}$ is enabled. If $\overline{\text{PME}}$ is not enabled, then this bit is cleared by the assertion of $\overline{\text{PRST}}$ or $\overline{\text{GRST}}$.

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.

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	TYPE	FUNCTION
31–26	R	Reserved. These bits return 0s when read.
25	R	SKTACCES. Socket access status. This bit provides information on when a socket access has occurred. This bit is cleared by a read access. 0 = No PC Card access has occurred (default) 1 = PC Card has been accessed
24	R	SKTMODE. Socket mode status. This bit provides clock mode information. 0 = Normal clock operation 1 = Clock frequency has changed
23–17	R	Reserved. These bits return 0s when read.
16	R/W	CLKCTRLEN. CardBus clock control enable. This bit, when set, enables clock control according to bit 0 (CLKCTRL). 0 = Clock control disabled (default) 1 = Clock control enabled
15–1	R	Reserved. These bits return 0s when read.
0	R/W	This bit determines whether the CardBus clock run master stops the CCLK or slows the clock when it detects idle activity on the CardBus interface. 0 = Stop the CardBus clock using the clock run protocol when there is no activity on the CardBus interface (default). 1 = Divide PCI clock by 16 using the clock run protocol when there is no activity on the CardBus interface.

7 Distributed DMA (DDMA) Registers

The DMA base address, programmable in PCI configuration space at offset 98h, points to a 16-byte region in PCI I/O space where the DDMA registers reside. Table 7–1 summarizes the names and locations of these registers. These registers are identical in function, but different in location from the Intel 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 forwarding legacy DMA writes to DMA channels.

These PCI4450 DMA register definitions are identical to those registers of the same name in the 8237 DMA controller; however, some register bits defined in the 8237 do not apply to distributed DMA in a PCI environment. In such cases, the PCI4450 will implement these obsolete register bits as nonfunctional, read-only bits. The reserved registers shown in Table 7–1 are implemented as read-only, and return 0s when read. Writes to reserved registers have no effect.

Table 7–1. Distributed DMA Registers

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

7.1 DMA Current Address / Base Address Register

This register is used to set the starting (base) memory address of a DMA transfer. Reads from this register indicate the current memory address of a direct memory transfer.

For the 8-bit DMA transfer mode, the DMA current address register contents are presented on AD15–0 of the PCI bus during the address phase. Bits 7–0 of the DMA page register are presented on AD23–AD16 of the PCI bus during the address phase.

For the 16-bit DMA transfer mode, the DMA 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 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: 00 0000h
Size: Two bytes

7.2 DMA Page Register

This register is used to set the upper byte of the address of a DMA transfer. Details of the address represented by this register are explained in the DMA current address/base address register, above.

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: 0000h
Size: One byte

7.3 DMA Current Count / Base Count Register

This register is used to set the total transfer count, in bytes, of a direct memory transfer. Reads from 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. Likewise, the count is decremented by 2 in 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
Size: Two bytes

7.4 DMA Command Register

This register is used to enable and disable the controller; all other bits are reserved.

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: 0000h
Size: One byte

Table 7–2. DDMA Command Register Description

BIT	TYPE	FUNCTION
7–3	R	Reserved. These bits return 0s when read.
2	R/W	DMA controller enable. This bit enables and disables the distributed DMA slave controller in the PCI4450, and defaults to the enabled state. 0 = DMA controller enabled (default) 1 = DMA controller disabled
1–0	R	Reserved. These bits return 0s when read.

7.5 DMA Status Register

This register indicates the terminal count and DMA request ($\overline{\text{DREQ}}$) status.

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: 0000h
Size: One byte

Table 7–3. DMA Status Register Description

BIT	TYPE	FUNCTION
7–4	R	DREQSTAT. Channel request. In the 8237, these bits indicate the status of the $\overline{\text{DREQ}}$ signal of each DMA channel. In the PCI4450, these bits indicate the $\overline{\text{DREQ}}$ status of the single socket being serviced by this register. All four bits are set when the PC Card asserts its $\overline{\text{DREQ}}$ signal, and are reset when $\overline{\text{DREQ}}$ is deasserted. The status of the mask bit in the DMA multichannel mask register has no effect on these bits.
3–0	R	TC. Channel terminal count. The 8327 uses these bits to indicate the TC status of each of its four DMA channels. In the PCI4450, these bits report information about just 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 when the terminal count (TC) is reached by the DMA channel. These bits are reset when read or when the DMA channel is reset

7.6 DMA Request Register

This register is used to request a DDMA transfer through software. Any write to this register enables software requests. 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: 0000h
Size: One byte

7.7 DMA Mode Register

This register is used to set the DMA transfer mode.

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/W	R
Default	0	0	0	0	0	0	0	0

Register: **DMA mode**
Type: Read-only, Read/Write
Offset: DMA Base Address + 0Bh
Default: 0000h
Size: One byte

Table 7–4. DDMA Mode Register Description

BIT	TYPE	FUNCTION
7–6	R/W	DMAMODE. Mode select bits. The PCI4450 uses these bits to determine the transfer mode. 00 = Demand mode select (default) 01 = Single mode select 10 = Block mode select 11 = Reserved
5	R/W	INCDEC. Address increment/decrement. The PCI4450 uses this register bit 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 use of this register bit, and is encoded as follows: 0 = Addresses increment (default) 1 = Addresses decrement
4	R/W	AUTOINIT. Auto-initialization bit. 0 = Auto-initialization disabled (default) 1 = Auto-initialization enabled
3–2	R/W	XFERTYPE. Transfer type. These bits select the type of direct memory transfer to be performed. A memory write transfer moves data from the PCI4450 PC Card interface to memory, and a memory read transfer moves data from memory to the PCI4450 PC Card interface. The field is encoded as: 00 = No transfer selected (default) 01 = Write transfer 10 = Read transfer 11 = Reserved
1–0	R	Reserved. These bits return 0s when read.

7.8 DMA Master Clear Register

This register is used to reset the DDMA controller, and resets all DDMA 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: 0000h
Size: One byte

7.9 DMA Multichannel Mask Register

The PCI4450 uses only the least significant bit of this register to mask the PC Card DMA channel. The PCI4450 sets the mask bit when the PC Card is removed. Host software is responsible for either resetting the socket's DMA controller or re-enabling the mask bit.

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	1

Register: **DMA multichannel mask**
Type: Read-only, Read/Write
Offset: DMA Base Address + 0Fh
Default: 0000h
Size: One byte

Table 7–5. DDMA MultiChannel Mask Register Description

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

8 OHCI-Lynx Controller Programming Model

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

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

Table 8–1. Bit Field Access Tag Descriptions

ACCESS TAG	NAME	MEANING
R	Read	Field may be read by software.
W	Write	Field may be written by software to any value.
S	Set	Field may be set by a write of 1. Writes of 0 have no effect.
C	Clear	Field may be cleared by a write of 1. Writes of 0 have no effect.
U	Update	Field may be autonomously updated by the PCI4450.

8.1 PCI Configuration Registers

The PCI4450 link function configuration header is compliant with the *PCI Specification* as a standard header. Table 8–2 illustrates the PCI configuration header which includes both the predefined portion of the configuration space and the user definable registers. The registers that are labeled Reserved are read-only returning 0 when read and are not applicable to the link function or have been reserved by the *PCI Specification* for future use.

Table 8–2. PCI Configuration Register Map

REGISTER NAME				OFFSET
Device ID		Vendor ID		00h
Status		Command		04h
Class code			Revision ID	08h
BIST	Header type	Latency timer	Cache line size	0Ch
Open HCI registers base address				10h
TI extension registers base address				14h
Reserved				18h
Reserved				1Ch
Reserved				20h
Reserved				24h
Reserved				28h
Subsystem ID		Subsystem vendor ID		2Ch
Reserved				30h
Reserved			Capabilites pointer	34h
Reserved				38h
Max latency	Min grant	Interrupt pin	Interrupt line	3Ch
PCI OHCI control				40h
Power management capabilities		Next item pointer	Capability ID	44h
PM data	PMCSR_BSE	Power management CSR		48h
Reserved				4C–ECh
PCI miscellaneous configuration				F0h
Link_enhancements				F4h
Subsystem ID alias		Subsystem vendor ID alias		F8h
GPIO3	GPIO2	GPIO1	GPIO0	FCh

8.2 Vendor ID Register

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

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

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

8.3 Device ID Register

This 16-bit read-only register contains a value assigned to the PCI4450 by Texas Instruments. The device identification for the PCI4450 OHCI controller function is 8011h.

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	0	0	1

Register: **Device ID register**
Type: Read-only
Offset: 02h
Default: 8011h

8.4 PCI Command Register

The command register provides control over the PCI4450 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.

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	FIELD NAME	TYPE	DESCRIPTION
15–10	RSVD	R	Reserved. These bits return 0s when read.
9	FBB_ENB	R	Fast back-to-back enable. The PCI4450 will not generate fast back to back transactions, thus this bit returns 0 when read.
8	SERR_ENB	R/W	$\overline{\text{SERR}}$ enable. When set, the PCI4450 $\overline{\text{SERR}}$ 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 PCI4450 does not support address/data stepping, and this bit is hardwired to 0.
6	PERR_ENB	R/W	Parity error enable. When set, the PCI4450 is enabled to drive $\overline{\text{PERR}}$ response to parity errors through the $\overline{\text{PERR}}$ signal.
5	VGA_ENB	R	VGA palette snoop enable. The PCI4450 does not feature VGA palette snooping. This bit returns 0 when read.
4	MWI_ENB	R/W	Memory write and invalidate enable. When set, the PCI4450 is enabled to generate MWI PCI bus commands. If reset, the PCI4450 will generate memory write commands instead.
3	SPECIAL	R	Special cycle enable. The PCI4450 function does not respond to special cycle transactions. This bit returns 0 when read.
2	MASTER_ENB	R/W	Bus master enable. When set, the PCI4450 is enabled to initiate cycles on the PCI bus.
1	MEMORY_ENB	R/W	Memory response enable. Setting this bit enables the PCI4450 to respond to memory cycles on the PCI bus. This bit must be set to access OHCI registers.
0	IO_ENB	R	I/O space enable. The PCI4450 link does not implement any I/O mapped functionality; thus, this bit returns 0 when read.

8.5 PCI Status Register

PCI Bus Specification, as seen in the bit descriptions.

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	FIELD NAME	TYPE	DESCRIPTION
15	PAR_ERR	RCU	Detected parity error. This bit is set when a parity error is detected, either address or data parity errors.
14	SYS_ERR	RCU	Signaled system error. This bit is set when $\overline{\text{SERR}}$ is enabled and the PCI4450 signaled a system error to the host.
13	MABORT	RCU	Received master abort. This bit is set when a cycle initiated by the PCI4450 on the PCI bus has been terminated by a master abort.
12	TABORT_REC	RCU	Received target abort. This bit is set when a cycle initiated by the PCI4450 on the PCI bus was terminated by a target abort.
11	TABORT_SIG	RCU	Signaled target abort. This bit is set by the PCI4450 when it terminates a transaction on the PCI bus with a target abort.
10–9	PCI_SPEED	R	DEVSEL timing. These bits encode the timing of $\overline{\text{DEVSEL}}$ and are hardwired 01b indicating that the PCI4450 asserts this signal at a medium speed on non-configuration cycle accesses.
8	DATAPAR	RCU	Data parity error detected. This bit is set when the following conditions have been met: a. $\overline{\text{PERR}}$ was asserted by any PCI device including the PCI4450 b. The PCI4450 was the bus master during the data parity error c. The parity error response bit is set in the command register
7	FBB_CAP	R	Fast back-to-back capable. The PCI4450 cannot accept fast back-to-back transactions; thus, this bit is hardwired to 0.
6	UDF	R	UDF supported. The PCI4450 does not support the user definable features; thus, this bit is hardwired to 0.
5	66MHZ	R	66 MHz capable. The PCI4450 operates at a maximum PCLK frequency of 33 MHz; therefore, this bit is hardwired to 0.
4	CAPLIST	R	Capabilities list. This bit returns 1 when read, and indicates that capabilities additional to standard PCI are implemented. The linked list of PCI power management capabilities is implemented in this function.
3–0	RSVD	R	Reserved. These bits return 0s when read.

8.6 Class Code And Revision ID Register

This read-only register categorizes the PCI4450 as a serial bus controller (0Ch), controlling an IEEE1394 bus (00h), with an OHCI programming model (10h). Furthermore, the TI chip revision is indicated in the lower byte.

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

Register: **Class code and revision ID**
Type: Read-only
Offset: 08h
Default: 0C00 100Xh

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

BIT	FIELD NAME	TYPE	DESCRIPTION
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 a IEEE1394 serial bus.
15–8	PGMIF	R	Programming interface. This field returns 10h when read, which indicates that the programming model is compliant with the 1394 OHCI specification.
7–0	CHIPREV	R	Silicon revision. This field returns the silicon revision of the PCI4450.

8.7 Latency Timer And Class Cache Line Size Register

This register is programmed by host BIOS to indicate system cache line size and the latency timer associated with the PCI4450.

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	FIELD NAME	TYPE	DESCRIPTION
15–8	LATENCY_TIMER	R/W	PCI latency timer. The value in this register specifies the latency timer for the PCI4450, in units of PCI clock cycles. When the PCI4450 is a PCI bus initiator and asserts $\overline{\text{FRAME}}$, the latency timer will begin counting from zero. If the latency timer expires before the PCI4450 transaction has terminated, then the PCI4450 will terminate the transaction when its $\overline{\text{GNT}}$ is deasserted.
7–0	CACHELINE_SZ	R/W	Cache line size. This value is used by the PCI4450 during memory write and invalidate, memory read line, and memory read multiple transactions.

8.8 Header Type and BIST Register

This register indicates that this function is part of a multifunction device and has a standard PCI header type and indicates no built-in self-test.

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	FIELD NAME	TYPE	DESCRIPTION
15–8	BIST	R	Built-in self-test. The PCI4450 does not include a built-in self-test, and this field returns 00h when read.
7–0	HEADER_TYPE	R	PCI header type. The PCI4450 includes the standard PCI header, and this is communicated by returning 00h when this field is read.

8.9 Open HCI Registers Base Address Register

This 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.

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

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

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

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

8.10 TI Extension Base Address Register

This register is programmed with a base address referencing the memory mapped TI extension registers.

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

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

Table 8–9. TI Extension Base Address Register Description

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

8.11 PCI Subsystem Identification Register

This register is used for subsystem and option card identification purposes. This register can be initialized from the serial EEPROM or can be written using the subsystem access register.

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	FIELD NAME	TYPE	DESCRIPTION
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

This register provides a pointer into the PCI configuration header where the PCI power management register block resides. PCI4450 configuration header double-words 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 Pin Registers

This register is used to communicate interrupt line routing information.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Interrupt line and 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	0	0	0	0	0	0	0	0	0	0

Register: **Interrupt line and pin**
Type: Read-only, Read/Write
Offset: 3Ch
Default: 0000h

Table 8–11. Interrupt Line and Pin Registers Description

BIT	FIELD NAME	TYPE	DESCRIPTION
15–8	INTR_PIN	R	Interrupt pin register. This register returns 01h, 02h, or 03h when read, indicating that the PCI4450 link function signals interrupts on $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, or $\overline{\text{INTC}}$ pin.
7–0	INTR_LINE	R/W	Interrupt line register. This register is programmed by the system and indicates to the software which interrupt line the PCI4450 $\overline{\text{INTA}}$ is connected.

8.14 MIN_GNT and MAX_LAT registers

This register is used to communicate to the system the desired setting of the latency timer register. If a serial ROM is detected, then the contents of this register are loaded through the serial ROM interface after a PCI reset. If no serial ROM is detected, then this register returns a default value that corresponds to the MIN_GNT = 2, MAX_LAT = 4.

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	0	1	0	0	0	0	0	0	0	1	0

Register: **MIN_GNT and MAX_LAT**
Type: Read/Update
Offset: 3Eh
Default: 0202h

Table 8–12. MIN_GNT and MAX_LAT Registers Description

BIT	FIELD NAME	TYPE	DESCRIPTION
15–8	MAX_LAT	RU	Maximum latency. The contents of this register may be used by host BIOS to assign an arbitration priority-level to the PCI4450. The default for this register indicates that the PCI4450 may need to access the PCI bus as often as every 1/4 μs; thus, an extremely high priority level is requested. The contents of this field may also be loaded through the serial ROM.
7–0	MIN_GNT	RU	Minimum grant. The contents of this register may be used by host BIOS to assign a latency timer register value to the PCI4450. The default for this register indicates that the PCI4450 may need to sustain burst transfers for nearly 64 μs; thus, requesting a large value be programmed in the PCI4450 latency timer register.

8.15 PCI OHCI Control Register

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

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

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

8.16 Capability ID And Next Item Pointer Registers

This register identifies the linked list capability item, and provides a pointer to the next capability item.

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	FIELD NAME	TYPE	DESCRIPTION
15–8	NEXT_ITEM	R	Next item pointer. The PCI4450 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

This register indicates the capabilities of the PCI4450 related to PCI power management. In summary, the D0, D2, and $D3_{hot}$ device states are supported.

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	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Register: **Power management capabilities**
Type: Read/Update
Offset: 46h
Default: 4001h

Table 8–14. Power Management Capabilities Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
15	PME_D3COLD	RU	$\overline{PME}$ support from $D3_{cold}$. When set, PCI4450 generates a $\overline{PME}$ wake event from $D3_{cold}$. This bit state is dependent upon PCI4450 V_{aux} implementation and may be configured by host software using the PCI miscellaneous configuration register.
14–11	PME_SUPPORT	RU	$\overline{PME}$ support. This four-bit field indicates the power states from which the PCI4450 may assert $\overline{PME}$. These four bits return a value of 4'b1100 by default, indicating that $\overline{PME}$ may be asserted from the $D3_{hot}$ and D2 power states. Bit 13 may be modified by host software using the PCI miscellaneous configuration register.
10	D2_SUPPORT	R	D2 support. This bit returns a 1 when read, indicating that the PCI4450 supports the D2 power state.
9	D1_SUPPORT	R	D1 support. This bit returns a 0 when read, indicating that the PCI4450 does not support the D1 power state.
8	DYN_DATA	R	Dynamic data support. This bit returns a 0 when read, indicating that the PCI4450 does not report dynamic power consumption data.
7–6	RSVD	R	Reserved. These bits return 0s when read.
5	DSI	R	Device specific initialization. This bit returns 0 when read, indicating that the PCI4450 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 PCI4450 supports $\overline{PME}$ generation in the $D3_{cold}$ device state and requires V_{aux}, this bit returns 1 when read.
3	PME_CLK	R	$\overline{PME}$ clock. This bit returns 0 when read indicating that no host bus clock is required for the PCI4450 to generate $\overline{PME}$.
2–0	PM_VERSION	R	Power management version. This field returns 001b when read, indicating that the PCI4450 is compatible with the registers described in the revision 1.0 *PCI Bus Power Management Specification*.

8.18 Power Management Control And Status Register

This 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.

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	FIELD NAME	TYPE	DESCRIPTION
15	PME_STS	RC	This bit is set when the PCI4450 would normally be asserting the $\overline{PME}$ signal, independent of the state of the PME_ENB bit. This bit is cleared by a write back of 0, and this also clears the $\overline{PME}$ signal driven by the PCI4450. Writing a 0 to this bit has no effect.
14–9	DYN_CTRL	R	Dynamic data control. This bit field returns 0s when read since the PCI4450 does not report dynamic data.
8	PME_ENB	R/W	$\overline{PME}$ enable. This bit enables the function to assert $\overline{PME}$. If the bit is cleared assertion of $\overline{PME}$ is disabled.
7–5	RSVD	R	Reserved. These bits return 0s when read.
4	DYN_DATA	R	Dynamic data. This bit returns 0 when read since the PCI4450 does not report dynamic data.
3–2	RSVD	R	Reserved. These bits return 0s when read.
1–0	PWR_STATE	R/W	Power state. This two-bit field is used to set the PCI4450 device power state, and is encoded as follows: 00 = Current power state is D0 01 = Current power state is D1 10 = Current power state is D2 11 = Current power state is $D3_{hot}$

8.19 Power Management Extension Register

This register provides extended power management features not applicable to the PCI4450, thus it is read-only and returns 0 when read.

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	FIELD NAME	TYPE	DESCRIPTION
15–8	PM_DATA	R	Power management data. This bit field returns 0s when read since the PCI4450 does not report dynamic data.
7–0	PMCSR_BSE	R	Power management CSR – bridge support extensions. This field returns 0s since the PCI4450 does not provide P2P bridging.

8.20 PCI Miscellaneous Configuration Register

This register provides miscellaneous PCI-related configuration.

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/W	R/W	R/W	R/W
Default	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0

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

Table 8–17. PCI Miscellaneous Configuration Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31–16	RSVD	R	Reserved. Bits 31–16 return 0s when read.
15	PME_D3COLD	R/W	$\overline{\text{PME}}$ support from $D3_{cold}$. This bit is used to program the corresponding read-only value read from power management capabilities. This bit retains state through PCI reset and D3 – D0 transitions.
14	RSVD	R	Reserved. Bit 14 returns 0 when read.
13	PME_SUPPORT_D2	R/W	$\overline{\text{PME}}$ support. This bit is used to program the corresponding read-only value read from power management capabilities. If wake from the D2 power state implemented in PCI4450 is not desired, then this bit may be cleared to indicate to power management software that wake–up from D2 is not supported. This bit retains state through PCI reset and D3 – D0 transitions.
12–11	RSVD	R	Reserved. Bits 12 and 11 return 0s when read.
10	D2_SUPPORT	R/W	D2 support. This bit is used to program the corresponding read-only value read from power management capabilities. If the D2 power state implemented in PCI4450 is not desired, then this bit may be cleared to indicate to power management software that D2 is not supported. This bit retains state through PCI reset and D3 – D0 transitions.
9–4	RSVD	R	Reserved. Bits 9–4 return 0s when read.
3	RSVD	R/W	Reserved. Bit 3 defaults to 0.
2	DISABLE_SCLKGATE	R/W	When set, the internal SCLK runs identically with the chip input.
1	DISABLE_PCIGATE	R/W	When set, the internal PCI clock runs identically with the chip input.
0	KEEP_PCLK	R/W	When set, the PCI clock is always kept running through the $\overline{\text{CLKRUN}}$ protocol. When cleared, the PCI clock may be stopped using $\overline{\text{CLKRUN}}$.

8.21 Link Enhancement Control Register

This register implements TI proprietary bits that are initialized by software or by a serial EEPROM if present. After these bits are set, their functionality is enabled only if the aPhyEnhanceEnable bit in the HCControl register is set.

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

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

Table 8–18. Link Enhancement Control Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31–14	RSVD	R	Reserved. These bits return 0s 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 PCI4450 retries the packet, it uses a 2K-byte threshold resulting in store-and-forward operation. 00 = Threshold ~ 2 Kbytes resulting in store-and-forward operation 01 = Threshold ~ 1.7 Kbytes (default) 10 = Threshold ~ 1 K 11 = Threshold ~ 512 bytes
11–10	RSVD	R	Reserved. This bit returns 0 when read.
9	enab_audio_ts	R/W	Enable audio/music CIP timestamp enhancement. When this bit is set, the enhancement is enabled for audio/music CIP transmit streams (FMT = 6'h10).
8	enab_dv_ts	R/W	Enable DV CIP timestamp enhancement. When this bit is set, the enhancement is enabled for DV CIP transmit streams (FMT = 6'h00).
7	enab_unfair	R/W	Enable asynchronous priority requests. OHCILynx (TSB12LV22) compatible.
6	RSVD	R	This reserved field will not be assigned in PCI4450 follow–on products since this bit location loaded by the serial ROM from the *enhancements* field corresponds to HCControl.programPhyEnable in Open HCI register space.
5–3	RSVD	R	Reserved. Bits 5–3 return 0s when read.
2	enab_insert_idle	R/W	Enable insert idle. OHCILynx (TSB12LV22) compatible.
1	enab_accel	R/W	Enable acceleration enhancements. OHCILynx (TSB12LV22) compatible.
0	RSVD	R	Reserved. This bit returns 0 when read.

8.22 Subsystem Access Register

This register is used for system and option card identification purposes. The contents of this register are aliased to subsystem identification register at address 2Ch.

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	FIELD NAME	TYPE	DESCRIPTION
31–16	SUBDEV_ID	R/W	Subsystem device ID. This field indicates the subsystem device ID.
15–0	SUBVEN_ID	R/W	Subsystem vendor ID. This field indicates the subsystem vendor ID.

8.23 GPIO Control Register

This 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 the disable_BMC bit to 1. The LPS terminal can be configured as GPIO1 by setting the disable_LPS bit to 1.

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. General-Purpose Input/Output Control Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31–30	RSVD	R	Reserved. These bits return 0s when read.
29	GPIO_INV3	R/W	GPIO3 polarity invert. This bit controls the input/output polarity control of GPIO3. 0 = noninverted (default) 1 = inverted
28	GPIO_ENB3	R/W	GPIO3 enable control. This bit controls the output enable for GPIO3 0 = high-impedance output (default) 1 = output enabled
27–25	RSVD	R	Reserved. These bits return 0s when read.
24	GPIO_DATA3	R/W	GPIO3 data. When GPIO3 output is enabled, the value written to this bit represents the logical data driven to the GPIO3 terminal.
23–22	RSVD	R	Reserved. These bits return 0s when read.
21	GPIO_INV2	R/W	GPIO2 polarity invert. This bit controls the input/output polarity control of GPIO2. 0 = noninverted (default) 1 = inverted
20	GPIO_ENB2	R/W	GPIO2 enable control. This bit controls the output enable for GPIO2. 0 = high-impedance output (default) 1 = output enabled
19–17	RSVD	R	Reserved. These bits return 0s when read.
16	GPIO_DATA2	R/W	GPIO2 data. When GPIO2 output is enabled, the value written to this bit represents the logical data driven to the GPIO2 terminal.
15	DISABLE_LPS	R/W	Disable link power status (LPS). This bit configures this terminal as 0 = LPS (default) 1 = GPIO1
14	RSVD	R	Reserved. This bit returns 0 when read.
13	GPIO_INV1	R/W	GPIO1 polarity invert. When DISABLE_LPS bit is set to 1, this bit controls the input output polarity control of GPIO1 0 = noninverted (default) 1 = inverted
12	GPIO_ENB1	R/W	GPIO1 enable control. When DISABLE_LPS bit is set to 1, this bit controls the output enable for GPIO1 0 = high-impedance output 1 = output enabled (default)

Table 8–20. General-Purpose Input/Output Control Register Description (Continued)

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

8.24 Link Timer Adjustment Register

This register is used to control the link rollover value, and should be programmed with a nonzero value only when PCI4450 is the 1394 cycle master.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Link timer adjustment															
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 timer adjustment															
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: **Link timer adjustment**
Type: Read-only, Read/Write
Offset: C10h
Default: 0000 0000h

Table 8–21. Link Timer Adjustment Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31–8	RSVD	R	Reserved. These bits return 0s when read.
7	TIMER_ADJ_REQ	R	This bit indicates that the timer adjust request is active, but not completed. This bit is set whenever the timer adjust field is written, and is cleared when the actual adjustment is made.
6–4	RSVD	R	Reserved. These bits return 0s when read.
3–0	TIMER_ADJ_VAL	R/W	Cycle timer adjustment value. This four bit signed value is used to adjust the PCI4450 cycleTimer. The cycle timer offset field may be adjusted from +7 to –8 using this register.

9 Open HCI Registers

The open HCI registers defined by the *IEEE1394 Open HCI Specification* are memory mapped into a 2-Kbyte region of memory pointed to by the OHCI base address register at offset 10h in PCI configuration space. These registers are the primary interface for controlling the PCI4450 IEEE1394 link function.

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

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

Table 9–1. Open HCI Register Map

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

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

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

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

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

9.1 OHCI Version Register

This register indicates the OHCI version support, and whether or not the serial ROM is present.

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	0

Register: **OHCI version**
Type: Read-only
Offset: 00h
Default: 0X01 0000h

Table 9–2. OHCI Version Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31–25	RSVD	R	Reserved. These bits return 0s when read.
24	GUID_ROM	R	The PCI4450 sets this bit if the serial ROM is detected. If the serial ROM is present, then the Bus_Info_Block will be automatically loaded on hardware reset.
23–16	version	R	Major version of the Open HCI. The PCI4450 is compliant with the OHCI specification version 1.00; thus, this field reads 8'h01.
15–8	RSVD	R	Reserved. These bits return 0s when read.
7–0	revision	R	Minor version of the Open HCI. The PCI4450 is compliant with the OHCI specification version 1.00; thus, this field reads 8'h00.

9.2 GUID ROM Register

This register is used to access the serial ROM, and is only applicable if the Version.GUID_ROM bit is set.

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	FIELD NAME	TYPE	DESCRIPTION
31	addrReset	RSU	Software sets this bit to reset the GUID ROM address to 0. When the PCI4450 completes the reset, it clears this bit. The PCI4450 does not automatically fill rdData with the 0^{th} byte.
30–26	RSVD	R	Reserved. These bits return 0s when read.
25	rdStart	RSU	A read of the currently addressed byte is started when this bit is set. This bit is automatically cleared when the PCI4450 completes the read of the currently addressed GUID ROM byte.
24	RSVD	R	Reserved. ThIs bit returns 0 when read.
23–16	rdData	RU	This field represents the data read from the GUID ROM.
15–0	RSVD	R	Reserved. These bits return 0s when read.

9.3 Asynchronous Transmit Retries Register

This register indicates the number of times the PCI4450 will attempt a retry for asynchronous DMA request transmit and for asynchronous physical and DMA response transmit.

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	FIELD NAME	TYPE	DESCRIPTION
31–29	secondLimit	R	The second limit field returns 0s when read, since outbound dual-phase retry is not implemented.
28–16	cycleLimit	R	The cycle limit field returns 0s when read, since outbound dual-phase retry is not implemeted.
15–12	RSVD	R	Reserved. These bits 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

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

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

Register: **CSR data**
Type: Read-only
Offset: 0Ch
Default: XXXX XXXXh

9.5 CSR Compare Register

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

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

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

9.6 CSR Control Register

This register is used to access the bus management CSR registers from the host through compare-swap operations. This register is used to control the compare-swap operation and select the CSR resource.

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	X	X

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

Table 9–5. CSR Control Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31	csrDone	RU	This bit is set by the PCI4450 when a compare-swap operation is complete. It is reset whenever this register is written.
30–2	RSVD	R	Reserved. These bits return 0s when read.
1–0	csrSel	R/W	This field selects the CSR resource as follows: 00 = BUS_MANAGER_ID 01 = BANDWIDTH_AVAILABLE 10 = CHANNELS_AVAILABLE_HI 11 = CHANNELS_AVAILABLE_LO

9.7 Configuration ROM Header Register

This register externally maps to the first quadlet of the 1394 configuration ROM, offset 48'hFFFF_F000_0400.

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	FIELD NAME	TYPE	DESCRIPTION
31–24	info_length	R/W	IEEE1394 bus mgmt field. Must be valid when HCControl.linkEnable bit is set.
23–16	crc_length	R/W	IEEE1394 bus mgmt field. Must be valid when HCControl.linkEnable bit is set.
15–0	rom_crc_value	R/W	IEEE1394 bus management field. Must be valid at any time the HCControl.linkEnable bit is set. The reset value is undefined if no serial ROM is present. If a serial ROM is present, then this field is loaded from the serial ROM.

9.8 Bus Identification Register

This register externally maps to the first quadlet in the Bus_Info_Block, and contains the constant 32'h31333934, which is the ASCII value of 1394.

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

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

9.9 Bus Options Register

This register externally maps to the second quadlet of the Bus_Info_Block.

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	FIELD NAME	TYPE	DESCRIPTION
31	irmc	R/W	Isochronous resource manager capable. IEEE1394 bus management field. Must be valid when HCControl.linkEnable bit is set.
30	cmc	R/W	Cycle master capable. IEEE1394 bus management field. Must be valid when HCControl.linkEnable bit is set.
29	isc	R/W	Isochronous support capable. IEEE1394 bus management field. Must be valid when HCControl.linkEnable bit is set.
28	bmc	R/W	Bus manager capable. IEEE1394 bus management field. Must be valid when HCControl.linkEnable bit is set.
27	pmc	R/W	IEEE1394 bus management field. Must be valid when HCControl.linkEnable bit is set.
26–24	RSVD	R	Reserved. These bits return 0s when read.
23–16	cyc_clk_acc	R/W	Cycle master clock accuracy. (accuracy in parts per million) IEEE1394 bus management field. Must be valid when HCControl.linkEnable bit is set.
15–12	max_rec	R/W	IEEE 1394 bus management field. Hardware shall initialize max_rec 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 greater, and is calculated by 2^(max_rec + 1). Software may change max_rec, however this field must be valid at any time the HCControl.linkEnable bit is set. 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 value indicating 2048 bytes on hard reset.
11–8	RSVD	R	Reserved. These bits return 0s when read.
7–6	g	R/W	Generation counter. This field shall be incremented is any portion the configuration ROM has incremented since the prior bus reset.
5–3	RSVD	R	Reserved. These bits return 0s when read.
2–0	Lnk_spd	R	Link speed. This field returns 010, indicating that the link speeds of 100, 200 and 400 Mbits/s are supported.

9.10 GUID High Register

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

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

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

9.11 GUID Low Register

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

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

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

9.12 Configuration ROM Mapping Register

This register contains the start address within system memory that will map to the start address of 1394 configuration ROM for this node.

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	FIELD NAME	TYPE	DESCRIPTION
31–10	configROMaddr	R/W	If a quadlet read request to 1394 offset 48'hFFFF_F000_0400 through offset 48'hFFFF_F000_07FF is received, then the low order 10 bits of the offset are added to this register to determine the host memory address of the read request.
9–0	RSVD	R	Reserved. These bits return 0s when read.

9.13 Posted Write Address Low Register

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

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	FIELD NAME	TYPE	DESCRIPTION
31–0	offsetLo	RU	The lower 32 bits of the 1394 destination offset of the write request that failed.

9.14 Posted Write Address High Register

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

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	FIELD NAME	TYPE	DESCRIPTION
31–16	sourceID	RU	This bus and node number of the node that issued the write request that failed.
15–0	offsetHi	RU	The upper 16 bits of the 1394 destination offset of the write request that failed.

9.15 Vendor ID Register

The vendor ID register holds the company ID of an organization that specifies any vendor-unique registers. The PCI4450 does not implement Texas Instruments unique behavior with regards to Open HCI. Thus this register is read-only and returns 0s when read.

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

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

9.16 Host Controller Control Register

This set/clear register pair provides flags for controlling the PCI4450 link function.

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

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

Table 9–11. Host Controller Control Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31	RSVD	R	Reserved. These bits return 0s when read.
30	noByteSwapData	RSC	This bit is used to control whether physical accesses to locations outside the PCI4450 itself as well as any other DMA data accesses should be swapped.
29–24	RSVD	R	Reserved. These bits return 0s when read.
23	programPhyEnable	RC	This bit informs upper level software that lower level software has consistently configured the p1394a enhancements in the Link and PHY. When 1 generic software such as the OHCI driver is responsible for configuring p1394a enhancements in the PHY and the aPhyEnhanceEnable bit in the PCI4450. When 0, the generic software may not modify the p1394a enhancements in the PCI4450 or PHY and cannot interpret the setting of aPhyEnhanceEnable. This bit can be initialized from serial EEPROM.
22	aPhyEnhanceEnable	RSC	When the programPhyenable is 1 and link enable is 1, the OHCI driver can set this bit to use all p1394a enhancements. When programPhyEnable is set 0, the software shall not change PHY enhancements or the aPhyEnhanceEnable bit.
21–20	RSVD	R	Reserved. These bits return 0s when read.
19	LPS	RSC	This bit is used to control the link power status. Software must set LPS 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 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 clear the PCI4450 is logically and immediately disconnected from the 1394 bus, no packets will be received or processed nor will packets be transmitted.
16	SoftReset	RSCU	When set to 1, all PCI4450 state is reset, all FIFO's are flushed, and all OHCI registers are set to their hardware reset values unless otherwise specified. PCI registers are not affected by this bit. This bit remains set to 1 while the softReset is in progress and reverts back to 0 when the reset has completed.
15–0	RSVD	R	Reserved. These bits return 0s when read.

9.17 Self ID Buffer Pointer Register

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

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

Register: **Self ID buffer pointer**
Type: Read-only, Read/Write
Offset: 64h
Default: XXXX XX00h

9.18 Self ID Count Register

This register keeps a count of the number of times the bus self ID process has occurred, flags self ID packet errors, and keeps a count of the amount of self ID data in the self ID buffer.

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	FIELD NAME	TYPE	DESCRIPTION
31	selfIDError	RU	When this bit is 1, an error was detected during the most recent self ID packet reception. The contents of the self ID buffer are undefined. This bit is cleared after a self ID reception in which no errors are detected. Note that an error can be a hardware error or a host bus write error.
30–24	RSVD	R	Reserved. These bits 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. These bits return 0s when read.
10–2	selfIDSize	RU	This field indicates the number of quadlets that have been written into the self ID buffer for the current selfIDGeneration. This includes the header quadlet and the self ID data. This field is cleared to 0 when the self ID reception begins.
1–0	RSVD	R	Reserved. These bits return 0s when read.

9.19 ISO Receive Channel Mask High Register

This set/clear register is used to enable packet receives from the upper 32 isochronous data channels. A read from either the set register or clear register returns the value of the IRChannelMaskHi register.

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

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

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

BIT	FIELD NAME	TYPE	DESCRIPTION
31	isoChannel63	RSC	When set, the PCI4450 is enabled to receive from iso channel number 63.
30	isoChannel62	RSC	When set, the PCI4450 is enabled to receive from iso channel number 62.
29	isoChannel61	RSC	When set, the PCI4450 is enabled to receive from iso channel number 61.
28	isoChannel60	RSC	When set, the PCI4450 is enabled to receive from iso channel number 60.
27	isoChannel59	RSC	When set, the PCI4450 is enabled to receive from iso channel number 59.
26	isoChannel58	RSC	When set, the PCI4450 is enabled to receive from iso channel number 58.
25	isoChannel57	RSC	When set, the PCI4450 is enabled to receive from iso channel number 57.
24	isoChannel56	RSC	When set, the PCI4450 is enabled to receive from iso channel number 56.
23	isoChannel55	RSC	When set, the PCI4450 is enabled to receive from iso channel number 55.
22	isoChannel54	RSC	When set, the PCI4450 is enabled to receive from iso channel number 54.
21	isoChannel53	RSC	When set, the PCI4450 is enabled to receive from iso channel number 53.
20	isoChannel52	RSC	When set, the PCI4450 is enabled to receive from iso channel number 52.
19	isoChannel51	RSC	When set, the PCI4450 is enabled to receive from iso channel number 51.
18	isoChannel50	RSC	When set, the PCI4450 is enabled to receive from iso channel number 50.
17	isoChannel49	RSC	When set, the PCI4450 is enabled to receive from iso channel number 49.
16	isoChannel48	RSC	When set, the PCI4450 is enabled to receive from iso channel number 48.
15	isoChannel47	RSC	When set, the PCI4450 is enabled to receive from iso channel number 47.
14	isoChannel46	RSC	When set, the PCI4450 is enabled to receive from iso channel number 46.
13	isoChannel45	RSC	When set, the PCI4450 is enabled to receive from iso channel number 45.
12	isoChannel44	RSC	When set, the PCI4450 is enabled to receive from iso channel number 44.
11	isoChannel43	RSC	When set, the PCI4450 is enabled to receive from iso channel number 43.
10	isoChannel42	RSC	When set, the PCI4450 is enabled to receive from iso channel number 42.
9	isoChannel41	RSC	When set, the PCI4450 is enabled to receive from iso channel number 41.
8	isoChannel40	RSC	When set, the PCI4450 is enabled to receive from iso channel number 40.

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

BIT	FIELD NAME	TYPE	DESCRIPTION
7	isoChannel39	RSC	When set, the PCI4450 is enabled to receive from iso channel number 39.
6	isoChannel38	RSC	When set, the PCI4450 is enabled to receive from iso channel number 38.
5	isoChannel37	RSC	When set, the PCI4450 is enabled to receive from iso channel number 37.
4	isoChannel36	RSC	When set, the PCI4450 is enabled to receive from iso channel number 36.
3	isoChannel35	RSC	When set, the PCI4450 is enabled to receive from iso channel number 35.
2	isoChannel34	RSC	When set, the PCI4450 is enabled to receive from iso channel number 34.
1	isoChannel33	RSC	When set, the PCI4450 is enabled to receive from iso channel number 33.
0	isoChannel32	RSC	When set, the PCI4450 is enabled to receive from iso channel number 32.

9.20 ISO Receive Channel Mask Low Register

This set/clear register is used to enable packet receives from the lower 32 isochronous data channels.

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

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

Table 9–14. ISO Receive Channel Mask Low Register Descriptions

BIT	FIELD NAME	TYPE	DESCRIPTION
31	isoChannel31	RSC	When set, the PCI4450 is enabled to receive from iso channel number 31.
30	isoChannel30	RSC	When set, the PCI4450 is enabled to receive from iso channel number 30.
⋮	⋮	⋮	Bits 29 through 2 follow the same pattern
1	isoChannel1	RSC	When set, the PCI4450 is enabled to receive from iso channel number 1.
0	isoChannel0	RSC	When set, the PCI4450 is enabled to receive from iso channel number 0.

9.21 Interrupt Event Register

This set/clear register reflects the state of the various PCI4450 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 the bits in this register is to write a 1 to the corresponding bit in the clear register.

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

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

Table 9–15. Interrupt Event Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31	RSVD	R	Reserved. This bit returns 0 when read.
30	vendorSpecific	R	Vendor defined.
29–27	RSVD	R	Reserved. These bits return 0s when read.
26	phyRegRcvd	RSCU	The PCI4450 has received a PHY register data byte which can be read from the PHY control register.
25	cycleTooLong	RSCU	If LinkControl.cycleMaster is set, this indicated that over 125 μs elapsed between the start of sending a cycle start packet and the end of a subaction gap. LinkControl.cycleMaster is cleared by this event.
24	unrecoverableError	RSCU	This event occurs when the PCI4450 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. While unrecoverableError is set, all normal interrupts for the context(s) that caused this interrupt will be blocked from being set.
23	cycleInconsistent	RSCU	A cycle start was received that had an isochronous cycleTimer.seconds and isochronous cycleTimer.count different from the value in the CycleTimer register.
22	cycleLost	RSCU	A lost cycle is indicated when no cycle_start packet is sent/received between two successive cycleSynch events. A lost cycle can be predicted when a cycle_start packet does not immediately follow the first subaction gap after the cycleSynch event or if an arbitration reset gap is detected after a cycleSynch event without an intervening cycle start. CycleLost may be set either when it occurs or when logic predicts that it will occur.
21	cycle64Seconds	RSCU	Indicates that the 7th bit of the cycle second counter has changed.
20	cycleSynch	RSCU	Indicates that a new isochronous cycle has started and is set 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. These bits return 0s when read.
17	busReset	RSCU	Indicates that the PHY chip has entered bus reset mode.

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

BIT	FIELD NAME	TYPE	DESCRIPTION
16	selfDcomplete	RSCU	A selfID packet stream has been received. It will be generated at the end of the bus initialization process. This bit is turned off simultaneously when IntEvent.busReset is turned on.
15–10	RSVD	R	Reserved. These bits return 0s when read.
9	lockRespErr	RSCU	Indicates that the PCI4450 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 PCI4450 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 and interrupt. This is not a latched event, it is the OR'ing of all bits in (isoRecvIntEvent and isoRecvIntMask). The isoRecvIntEvent register indicates which contexts have interrupted.
6	iscohTx	RU	Isochronous transmit DMA interrupt. Indicates that one or more isochronous transmit contexts have generated and interrupt. This is not a latched event, it is the OR'ing of all bits in (isoXmitIntEvent and isoXmitIntMask). The isoXmitIntEvent register indicates which contexts have interrupted.
5	RSPkt	RSCU	Indicates that a packet was sent to an asynchronous receive response context buffer and the descriptor's xferStatus and resCount fields have been updated.
4	RQPkt	RSCU	Indicates that a packet was sent to an asynchronous receive request context buffer and the descriptor's xferStatus and resCount fields have been updated.
3	ARRS	RSCU	Async receive response DMA interrupt. This bit is conditionally set upon completion of an ARRS context command descriptor.
2	ARRQ	RSCU	Async receive request DMA interrupt. This bit is conditionally set upon completion of an ARRQ DMA context command descriptor.
1	respTxComplete	RSCU	Asynchronous response transmit DMA interrupt. This bit is conditionally set upon completion of an ATRS DMA command.
0	reqTxComplete	RSCU	Asynchronous request transmit DMA interrupt. This bit is conditionally set upon completion of an ATRQ DMA command.

9.22 Interrupt Mask Register

This set/clear register is used to enable the various PCI4450 interrupt sources. Reads from either the set register or the clear register always return IntMask. In all cases except masterIntEnable (bit 31), the enables for each interrupt event align with the event register bits detailed in Table 9–15.

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
Offset: 88h set register
8Ch clear register
Default: XXXX 0XXXh

Table 9–16. Interrupt Mask Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31	masterIntEnable	RSC	When set, external interrupts will be generated in accordance with the IntMask register. If clear, no external interrupts will be generated.
30–0			See Table 9–15

9.23 Isochronous Transmit Interrupt Event Register

This set/clear register reflects the interrupt state of the isochronous transmit contexts. An interrupt is generated on behalf of an isochronous transmit context if an OUTPUT_LAST command completes and its interrupt bits are set. Upon determining that the IntEvent.isochTx interrupt has occurred, software can check this register to determine which context(s) caused the interrupt. The interrupt bits are set 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 the bits in this register is to write a 1 to the corresponding bit in the clear register.

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
Offset: 90h set register
84h clear register [returns IsoXmitEvent and IsoXmitMask when read]
Default: 0000 00XXh

Table 9–17. Isochronous Transmit Interrupt Event Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31–8	RSVD	R	Reserved. These bits return 0s when read.
7	isoXmit7	RSC	Isochronous transmit channel 7 caused the isochTx interrupt.
6	isoXmit6	RSC	Isochronous transmit channel 6 caused the isochTx interrupt.
5	isoXmit5	RSC	Isochronous transmit channel 5 caused the isochTx interrupt.
4	isoXmit4	RSC	Isochronous transmit channel 4 caused the isochTx interrupt.
3	isoXmit3	RSC	Isochronous transmit channel 3 caused the isochTx interrupt.
2	isoXmit2	RSC	Isochronous transmit channel 2 caused the isochTx interrupt.
1	isoXmit1	RSC	Isochronous transmit channel 1 caused the isochTx interrupt.
0	isoXmit0	RSC	Isochronous transmit channel 0 caused the isochTx interrupt.

9.24 Isochronous Transmit Interrupt Mask Register

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

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous transmit interrupt mask															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous transmit interrupt mask															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
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
Offset: 98h set register
9Ch clear register
Default: 0000 00XXh

9.25 Isochronous Receive Interrupt Event Register

This set/clear register reflects the interrupt state of the isochronous receive contexts. An interrupt is generated on behalf of an isochronous receive context if an INPUT_* command completes and its interrupt bits are set. Upon determining that the IntEvent.isochRx interrupt has occurred, software can check this register to determine which context(s) caused the interrupt. The interrupt bits are set 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 the bits in this register is to write a 1 to the corresponding bit in the clear register.

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
Offset: A0h set register
A4h clear register [returns IsoRecvEvent and IsoRecvMask when read]
Default: 0000 000Xh

Table 9–18. Isochronous Receive Interrupt Event Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31–4	RSVD	R	Reserved. These bits return 0s when read.
3	isoRecv3	RSC	Isochronous receive channel 3 caused the isochRx interrupt.
2	isoRecv2	RSC	Isochronous receive channel 2 caused the isochRx interrupt.
1	isoRecv1	RSC	Isochronous receive channel 1 caused the isochRx interrupt.
0	isoRecv0	RSC	Isochronous receive channel 0 caused the isochRx interrupt.

9.26 Isochronous Receive Interrupt Mask Register

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

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous receive interrupt mask															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Isochronous receive interrupt mask															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
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
Offset: A8h set register
ACh clear register
Default: 0000 000Xh

9.27 Fairness Control Register (Optional Register)

This register provides a mechanism by which software can direct the host controller to transmit multiple asynchronous requests during a fairness interval.

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

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

Table 9–19. Link Control Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31–8	RSVD	R	Reserved.
7–0	pri_req	R	This field specifies the maximum number of priority arbitration requests for asynchronous request packets that the link is permitted to make of the PHY during fairness interval.

9.28 Link Control Register

This set/clear register provides the control flags that enable and configure the link core protocol portions of the PCI4450. It contains controls for the receiver and cycle timer.

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
Offset: E0h set register
E4h clear register
Default: 00X0 0X00h

Table 9–20. Link Control Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31–23	RSVD	R	Reserved. These bits return 0s when read.
22	cycleSource	RSC	When 1, the cycle timer will use an external source (CYCLEIN) to determine when to roll over the cycle timer. When 0, the cycle timer rolls over when the timer reaches 3072 cycles of the 24.576 MHz clock (125 μs).
21	cycleMaster	RSCU	When set, and the PHY has notified the PCI4450 that it is root, the PCI4450 will generate a cycle start packet every time the cycle timer rolls over, based on the setting of the cycleSource bit. When 0, the OHICLynx will accept received cycle start packets to maintain synchronization with the node which is sending them. This bit is automatically reset when the cycleTooLong event occurs and cannot be set until the IntEvent.cycleTooLong bit is cleared.
20	CycleTimerEnable	RSC	When 1, the cycle timer offset will count cycles of the 24.576 MHz clock and roll over at the appropriate time based on the settings of the above bits. When 0, the cycle timer offset will not count.
19–11	RSVD	R	Reserved. These bits return 0s when read.
10	RcvPhyPkt	RSC	When 1, the receiver will accept 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 1, the receiver will accept 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. These bits return 0s when read.

9.29 Node Identification Register

This register contains the address of the node on which the OHICLynx chip resides, and indicates the valid node number status. The 16-bit combination of busNumber and NodeNumber is referred to as the node ID.

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
Offset: E8h
Default: 0000 11XXh

Table 9–21. Node Identification Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31	iDValid	RU	This bit indicates whether or not the PCI4450 has a valid node number. It is cleared when a 1394 bus reset is detected and set when the PCI4450 receives a new node number from the PHY.
30	root	RU	This bit is set during the bus reset process if the attached PHY is root.
29–28	RSVD	R	Reserved. These bits return 0s when read.
27	CPS	RU	Set if the PHY is reporting that cable power status is OK (VP 8V).
26–16	RSVD	R	Reserved. These bits return 0s when read.
15–6	busNumber	RWU	This number is used to identify the specific 1394 bus the PCI4450 belongs to when multiple 1394-compatiable buses are connected via a bridge.
5–0	NodeNumber	RU	This number is the physical node number established by the PHY during self-identification. It is automatically set to the value received from the PHY after the self-identification phase. If the PHY sets the nodeNumber to 63, software should not set ContextControl.run for either of the AT DMA contexts.

9.30 PHY Control Register

This register is used to read or write a PHY register.

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
Offset: ECh
Default: XXXX 0XXXh

Table 9–22. PHY Control Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31	rdDone	RU	This bit is cleared to 0 by the PCI4450 when either rdReg or 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. These bits 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 by software to initiate a read request to a PHY register, and is cleared by hardware when the request has been sent. The wrReg and rdReg bits must be used exclusively.
14	wrReg	RWU	This bit is set by software to initiate a write request to a PHY register, and is cleared by hardware when the request has been sent. The wrReg and rdReg bits must be used exclusively.
13–12	RSVD	R	Reserved. These bits 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

This read/write register indicates the current cycle number and offset. When the PCI4450 is cycle master, this register is transmitted with the cycle start message. When the PCI4450 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.

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	FIELD NAME	TYPE	DESCRIPTION
31–25	cycleSeconds	RWU	This field counts seconds (cycleCount rollovers) modulo 128.
24–12	cycleCount	RWU	This field counts cycles (cycleOffset rollovers) modulo 8000.
11–0	cycleOffset	RWU	This field counts 24.576 MHz clocks modulo 3072, i.e., 125 μs. If an external 8 kHz clock configuration is being used, then cycleOffset must be set to 0 at each tick of the external clock.

9.32 Asynchronous Request Filter High Register

This set/clear register is used to enable asynchronous receive requests on a per node basis, and handles the upper node IDs. When a packet is destined for either the physical request context or the ARRQ context, the source node ID is examined. If the bit corresponding to the node ID is not set in this register, then the packet is not acknowledged and the request is not queued. The node ID comparison is done if the source node is on the same bus as the PCI4450. All nonlocal bus sourced packets are not acknowledged unless bit 31 in this register is set.

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	FIELD NAME	TYPE	DESCRIPTION
31	asynReqAllBuses	RSC	If set to 1, all asynchronous requests received by the PCI4450 from nonlocal bus nodes will be accepted.
30	asynReqResource62	RSC	If set to 1 for local bus node number 62, asynchronous requests received by the PCI4450 from that node will be accepted.
29	asynReqResource61	RSC	If set to 1 for local bus node number 61, asynchronous requests received by the PCI4450 from that node will be accepted.
28	asynReqResource60	RSC	If set to 1 for local bus node number 60, asynchronous requests received by the PCI4450 from that node will be accepted.
27	asynReqResource59	RSC	If set to 1 for local bus node number 59, asynchronous requests received by the PCI4450 from that node will be accepted.
26	asynReqResource58	RSC	If set to 1 for local bus node number 58, asynchronous requests received by the PCI4450 from that node will be accepted.
25	asynReqResource57	RSC	If set to 1 for local bus node number 57, asynchronous requests received by the PCI4450 from that node will be accepted.
24	asynReqResource56	RSC	If set to 1 for local bus node number 56, asynchronous requests received by the PCI4450 from that node will be accepted.
23	asynReqResource55	RSC	If set to 1 for local bus node number 55, asynchronous requests received by the PCI4450 from that node will be accepted.
22	asynReqResource54	RSC	If set to 1 for local bus node number 54, asynchronous requests received by the PCI4450 from that node will be accepted.
21	asynReqResource53	RSC	If set to 1 for local bus node number 53, asynchronous requests received by the PCI4450 from that node will be accepted.
20	asynReqResource52	RSC	If set to 1 for local bus node number 52, asynchronous requests received by the PCI4450 from that node will be accepted.

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

BIT	FIELD NAME	TYPE	DESCRIPTION
19	asynReqResource51	RSC	If set to 1 for local bus node number 51, asynchronous requests received by the PCI4450 from that node will be accepted.
18	asynReqResource50	RSC	If set to 1 for local bus node number 50, asynchronous requests received by the PCI4450 from that node will be accepted.
17	asynReqResource49	RSC	If set to 1 for local bus node number 49, asynchronous requests received by the PCI4450 from that node will be accepted.
16	asynReqResource48	RSC	If set to 1 for local bus node number 48, asynchronous requests received by the PCI4450 from that node will be accepted.
15	asynReqResource47	RSC	If set to 1 for local bus node number 47, asynchronous requests received by the PCI4450 from that node will be accepted.
14	asynReqResource46	RSC	If set to 1 for local bus node number 46, asynchronous requests received by the PCI4450 from that node will be accepted.
13	asynReqResource45	RSC	If set to 1 for local bus node number 45, asynchronous requests received by the PCI4450 from that node will be accepted.
12	asynReqResource44	RSC	If set to 1 for local bus node number 44, asynchronous requests received by the PCI4450 from that node will be accepted.
11	asynReqResource43	RSC	If set to 1 for local bus node number 43, asynchronous requests received by the PCI4450 from that node will be accepted.
10	asynReqResource42	RSC	If set to 1 for local bus node number 42, asynchronous requests received by the PCI4450 from that node will be accepted.
9	asynReqResource41	RSC	If set to 1 for local bus node number 41, asynchronous requests received by the PCI4450 from that node will be accepted.
8	asynReqResource40	RSC	If set to 1 for local bus node number 40, asynchronous requests received by the PCI4450 from that node will be accepted.
7	asynReqResource39	RSC	If set to 1 for local bus node number 39, asynchronous requests received by the PCI4450 from that node will be accepted.
6	asynReqResource38	RSC	If set to 1 for local bus node number 38, asynchronous requests received by the PCI4450 from that node will be accepted.
5	asynReqResource37	RSC	If set to 1 for local bus node number 37, asynchronous requests received by the PCI4450 from that node will be accepted.
4	asynReqResource36	RSC	If set to 1 for local bus node number 36, asynchronous requests received by the PCI4450 from that node will be accepted.
3	asynReqResource35	RSC	If set to 1 for local bus node number 35, asynchronous requests received by the PCI4450 from that node will be accepted.
2	asynReqResource34	RSC	If set to 1 for local bus node number 34, asynchronous requests received by the PCI4450 from that node will be accepted.
1	asynReqResource33	RSC	If set to 1 for local bus node number 33, asynchronous requests received by the PCI4450 from that node will be accepted.
0	asynReqResource32	RSC	If set to 1 for local bus node number 32, asynchronous requests received by the PCI4450 from that node will be accepted.

9.33 Asynchronous Request Filter Low Register

This set/clear register is used to enable asynchronous receive requests on a per node basis, and handles the lower node IDs. Other than filtering different node IDs, this register behaves identical to the asynchronous request filter high register.

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	FIELD NAME	TYPE	DESCRIPTION
31	asynReqResource31	RSC	If set to 1 for local bus node number 31, asynchronous requests received by the PCI4450 from that node will be accepted.
30	asynReqResource30	RSC	If set to 1 for local bus node number 30, asynchronous requests received by the PCI4450 from that node will be accepted.
⋮	⋮	⋮	Bits 29 through 2 follow the same pattern.
1	asynReqResource1	RSC	If set to 1 for local bus node number 1, asynchronous requests received by the PCI4450 from that node will be accepted.
0	asynReqResource0	RSC	If set to 1 for local bus node number 0, asynchronous requests received by the PCI4450 from that node will be accepted.

9.34 Physical Request Filter High Register

This set/clear register is used to enable physical receive requests on a per node basis, and handles the upper node IDs. When a packet is destined for the physical request context, and the node ID has been compared against the ARRQ registers, then the comparison is done again with this register. If the bit corresponding to the node ID is not set in this register, then the request will be handled by the ARRQ context instead of the physical request context.

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	FIELD NAME	TYPE	DESCRIPTION
31	physReqAllBusses	RSC	If set to 1, then all asynchronous requests received by the PCI4450 from nonlocal bus nodes will be accepted.
30	physReqResource62	RSC	If set to 1 for local bus node number 62, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
29	physReqResource61	RSC	If set to 1 for local bus node number 61, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
28	physReqResource60	RSC	If set to 1 for local bus node number 60, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
27	physReqResource59	RSC	If set to 1 for local bus node number 59, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
26	physReqResource58	RSC	If set to 1 for local bus node number 58, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
25	physReqResource57	RSC	If set to 1 for local bus node number 57, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
24	physReqResource56	RSC	If set to 1 for local bus node number 56, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
23	physReqResource55	RSC	If set to 1 for local bus node number 55, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
22	physReqResource54	RSC	If set to 1 for local bus node number 54, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
21	physReqResource53	RSC	If set to 1 for local bus node number 53, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
20	physReqResource52	RSC	If set to 1 for local bus node number 52, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
19	physReqResource51	RSC	If set to 1 for local bus node number 51, then physical requests received by the PCI4450 from that node will be handled through the physical request context.

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

BIT	FIELD NAME	TYPE	DESCRIPTION
18	physReqResource50	RSC	If set to 1 for local bus node number 50, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
17	physReqResource49	RSC	If set to 1 for local bus node number 49, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
16	physReqResource48	RSC	If set to 1 for local bus node number 48, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
15	physReqResource47	RSC	If set to 1 for local bus node number 47, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
14	physReqResource46	RSC	If set to 1 for local bus node number 46, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
13	physReqResource45	RSC	If set to 1 for local bus node number 45, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
12	physReqResource44	RSC	If set to 1 for local bus node number 44, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
11	physReqResource43	RSC	If set to 1 for local bus node number 43, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
10	physReqResource42	RSC	If set to 1 for local bus node number 42, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
9	physReqResource41	RSC	If set to 1 for local bus node number 41, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
8	physReqResource40	RSC	If set to 1 for local bus node number 40, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
7	physReqResource39	RSC	If set to 1 for local bus node number 39, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
6	physReqResource38	RSC	If set to 1 for local bus node number 38, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
5	physReqResource37	RSC	If set to 1 for local bus node number 37, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
4	physReqResource36	RSC	If set to 1 for local bus node number 36, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
3	physReqResource35	RSC	If set to 1 for local bus node number 35, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
2	physReqResource34	RSC	If set to 1 for local bus node number 34, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
1	physReqResource33	RSC	If set to 1 for local bus node number 33, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
0	physReqResource32	RSC	If set to 1 for local bus node number 32, then physical requests received by the PCI4450 from that node will be handled through the physical request context.

9.35 Physical Request Filter Low Register

This set/clear register is used to enable physical receive requests on a per node basis, and handles the lower node IDs. When a packet is destined for the physical request context, and the node ID has been compared against the asynchronous request filter registers, then the node ID comparison is done again with this register. If the bit corresponding to the node ID is not set in this register, then the request will be handled by the asynchronous request context instead of the physical request context.

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	FIELD NAME	TYPE	DESCRIPTION
31	physReqResource31	RSC	If set to 1 for local bus node number 31, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
30	physReqResource30	RSC	If set to 1 for local bus node number 30, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
⋮	⋮	⋮	Bits 29 through 2 follow the same pattern.
1	physReqResource1	RSC	If set to 1 for local bus node number 1, then physical requests received by the PCI4450 from that node will be handled through the physical request context.
0	physReqResource0	RSC	If set to 1 for local bus node number 0, then physical requests received by the PCI4450 from that node will be handled through the physical request context.

9.36 Physical Upper Bound Register (Optional Register)

This register is an optional register and is not implemented. This register is read-only and returns all 0s.

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

Register: **Physical upper bound**
Type: Read-only
Offset: 120h
Default: 0000 0000h

9.37 Asynchronous Context Control Register

This set/clear register controls the state and indicates status of the DMA context.

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
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	FIELD NAME	TYPE	DESCRIPTION
31–16	RSVD	R	Reserved. These bits 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 PCI4450 will only change this bit on a hardware or software reset.
14–13	RSVD	R	Reserved. These bits return 0s when read.
12	wake	RSU	Software sets this bit to cause the PCI4450 to continue or resume descriptor processing. The PCI4450 will clear this bit on every descriptor fetch.
11	dead	RU	The PCI4450 sets this bit when it encounters a fatal error and clears the bit when software resets the run bit.
10	active	RU	The PCI4450 sets this bit to 1 when it is processing descriptors.
9–8	RSVD	R	Reserved. These bits return 0s when read.
7–5	spd	RU	This field indicates the speed at which a packet was received or transmitted, and only contains meaningful information for receive contexts. This field is encoded as: 000b = 100 Mbits/sec, 001b = 200 Mbits/sec, 010b = 400 Mbits/sec, and all other values are reserved.
4–0	eventcode	RU	This field holds the acknowledge sent by the link core for this packet or an internally generated error code if the packet was not transferred successfully.

9.38 Asynchronous Context Command Pointer Register

This register contains a pointer to the address of the first descriptor block that the PCI4450 will access when software enables the context by setting the ContextControl.run bit.

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	FIELD NAME	TYPE	DESCRIPTION
31–4	descriptorAddress	RWU	Contains the upper 28 bits of the address of a 16-byte aligned descriptor block.
3–0	Z	RWU	Indicates the number of contiguous descriptors at the address pointed to by the descriptor address. If Z is 0, it indicates that the descriptorAddress is not valid.

9.39 Isochronous Transmit Context Control Register

This set/clear register controls options, state, and status for the isochronous transmit DMA contexts. The n value in the following register addresses indicates the context number (n = 0, 1, 2, 3, …).

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
Offset: 200h + (16 * n) set register
204h + (16 * n) clear register
Default: XXXX X0XXh

Table 9–30. Isochronous Transmit Context Control Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31	cycleMatchEnable	RSCU	When set to 1, processing will occur such that the packet described by the context's first descriptor block will be transmitted in the cycle whose number is specified in the cycleMatch field of this register. The 13-bit cycleMatch field must match the 13-bit cycleCount field in the cycle start packet that is sent or received immediately before isochronous transmission begins.
30–16	cycleMatch	RSC	Contains a 15-bit value, corresponding to the lower order 2 bits of cycleSeconds and 13-bit cycleCount field. If cycleMatchEnable is set, then this IT DMA context will become enabled for transmits when the bus cycleCount value equals the cycleMatch value.
15	run	RSC	This bit is set by software to enable descriptor processing for the context and cleared by software to stop descriptor processing. The PCI4450 will only change this bit on a hardware or software reset.
14–13	RSVD	R	Reserved. These bits return 0s when read.
12	wake	RSU	Software sets this bit to cause the PCI4450 to continue or resume descriptor processing. The PCI4450 will clear this bit on every descriptor fetch.
11	dead	RU	The PCI4450 sets this bit when it encounters a fatal error, and clears the bit when software resets the run bit.
10	active	RU	The PCI4450 sets this bit to 1 when it is processing descriptors.
9–8	RSVD	R	Reserved. These bits return 0s when read.
7–5	spd	RU	This field in not meaningful for isochronous transmit contexts.
4–0	event code	RU	Following an OUTPUT_LAST* command, the error code is indicated in this field. Possible values are: ack_complete, evt_descriptor_read, evt_data_read, and evt_unknown.

9.40 Isochronous Transmit Context Command Pointer Register

This register contains a pointer to the address of the first descriptor block that the PCI4450 will access when software enables an ISO transmit context by setting the ContextControl.run bit. The n value in the following register addresses indicates the context number (n = 0, 1, 2, 3, …).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous 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 XXXXh

9.41 Isochronous Receive Context Control Register

This set/clear register controls options, state, and status for the isochronous receive DMA contexts. The n value in the following register addresses indicates the context number (n = 0, 1, 2, 3, …).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Ischronous 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	Ischronous 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: **Ischronous receive context control**
Type: Read/Set/Clear/Update
Offset: 400h + (32 * n) set register
404h + (32 * n) clear register
Default: X000 X0XXh

Table 9–31. Isochronous Receive Context Control

BIT	FIELD NAME	TYPE	DESCRIPTION
31	bufferFill	RSC	When set, received packets are placed back-to-back to completely fill each receive buffer. When clear, each received packet is placed in a single buffer. If the multiChanMode bit is set to 1, this bit must also be set to 1. The value of bufferFill must not be changed while active or run are set.
30	isochHeader	RSC	When set to 1, received isochronous packets will include the complete 4-byte isochronous packet header seen by the link layer. The end of the packet will be 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 clear, the packet header is stripped off of received isochronous packets. The packet header, if received, immediately precedes the packet payload. The value of isochHeader must not be changed while active or run are set.
29	cycleMatchEnable	RSCU	When set, the context will begin running only when the 13-bit cycleMatch field in the contextMatch register matches the 13-bit cycleCount in the cycleStart packet. The effects of this bit, however, are impacted by the values of other bits in this register. Once the context has become active, hardware clears the cycleMatchEnable bit. The value of cycleMatchEnable must not be changed while active or run are set.
28	multiChanMode	RSC	When set, the corresponding isochronous receive DMA context will receive packets for all isochronous channels enabled in the IRChannelMaskHi and IRChannelMaskLo registers. The isochronous channel number specified in the IRDMA context match register is ignored. When 0, the IRDMA context will receive packets for that single channel. Only one IRDMA context may use the IRChannelMask registers. If more that one IRDMA context control register has the multiChanMode bit set, results are undefined. The value of multiChanMode must not be changed while active or run are set.
27–16	RSVD	R	Reserved. These bits 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 PCI4450 will only change this bit on a hardware or software reset.
14–13	RSVD	R	Reserved. These bits return 0s when read.
12	wake	RSU	Software sets this bit to cause the PCI4450 to continue or resume descriptor processing. The PCI4450 will clear this bit on every descriptor fetch.
11	dead	RU	The PCI4450 sets this bit when it encounters a fatal error, and clears the bit when software resets the run bit.
10	active	RU	The PCI4450 sets this bit to 1 when it is processing descriptors.

Table 9–31. Isochronous Receive Context Control (Continued)

BIT	FIELD NAME	TYPE	DESCRIPTION
9–8	RSVD	R	Reserved. These bits return 0s when read.
7–5	spd	RU	This field indicates the speed at which the packet was received. 3'b000 = 100 Mbits/sec, 3'b001 = 200 Mbits/sec, and 3'b010 = 400 Mbits/sec. All other values are reserved.
4–0	event code	RU	Following and INPUT* command, the error code is indicated in this field.

9.42 Isochronous Receive Context Command Pointer Register

This register contains a pointer to the address of the first descriptor block that the PCI4450 will access when software enables an ISO receive context by setting the ContextControl.run bit. The n value in the following register addresses indicates the context number (n = 0, 1, 2, 3, …).

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

Register: **Isochronous receive context command pointer**
Type: Read-only
Offset: 40Ch + (32 * n)
Default: XXXX XXXXh

9.43 Isochronous Receive Context Match Register

This register is used to start an isochronous receive context running on a specified cycle number, to filter incoming isochronous packets based on tag values, and to wait for packets with a specified sync value. The n value in the following register addresses indicates the context number (n = 0, 1, 2, 3, …).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Isochronous receive context match															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
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	Isochronous receive context match															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	X	X	X	X	X	X	X	X	0	X	X	X	X	X	X	X

Register: **Isochronous receive context match**
Type: Read/Write
Offset: 410Ch + (32 * n)
Default: XXXX XXXXh

Table 9–32. Isochronous Receive Context Match Register Description

BIT	FIELD NAME	TYPE	DESCRIPTION
31	tag3	R/W	If set, this context will match on iso receive packets with a tag field of 2'b11.
30	tag2	R/W	If set, this context will match on iso receive packets with a tag field of 2'b10.
29	tag1	R/W	If set, this context will match on iso receive packets with a tag field of 2'b01.
28	tag0	R/W	If set, this context will match on iso receive packets with a tag field of 2'b00.
27–25	RSVD	R	Reserved. These bits return 0s when read.
24–12	cycleMatch	R/W	Contains a 13-bit value, corresponding to the 13-bit cycleCount field in the cycleStart packet. If cycleMatchEnable is set, then this context is enabled for receives when the bus cycleCount value equals the cycleMatch value.
11–8	sync	R/W	This field contains the 4-bit field which is compared to the sync field of each iso packet for this channel when the command descriptor's w field is set to 2'b11.
7	RSVD	R	Reserved. These bits return 0s when read.
6	tag1SyncFilter	R/W	If set and the tag1 bit is set , then packets with tag2b01 shall be accepted into the context if the two most significant bits of the packets sync field are 2'b00. Packets with tag values other than 2'b01 are filtered according to tag0, tag2 and tag3 without any additional restrictions. If clear, this context will match on isochronous receive packets as specified in the tag0–3 bits with no additional restrictions.
5–0	channelNumber	R/W	This 6-bit field indicates the isochronous channel number for which this IR DMA context will accept packets.

10GPIO Interface

The GPIO interface consists of four general-purpose input output ports. On power reset, GPIO0 and GPIO1 are enabled and are configured as bus manager contender (BMC) and link power status (LPS) , respectively. BMC and LPS outputs can be configured via the GPIO control register in PCI Configuration space, as GPIO0 and GPIO1. Figure 10–1 shows the schematic for GPIO0 implementation. Figure 10–2 shows the schematic for GPIO1 implementation.

GPIO2 and GPIO3 power up as general-purpose inputs and are programmable via the GPIO control register. Figure 10–3 shows the schematic for GPIO2 and GPIO3 implementation.

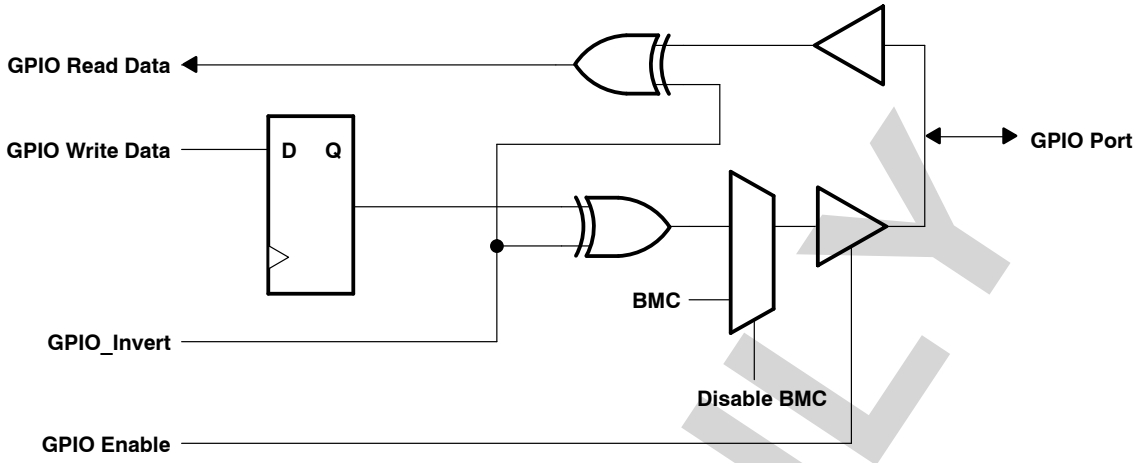


Figure 10–1. BMC/LINKON/GPIO0

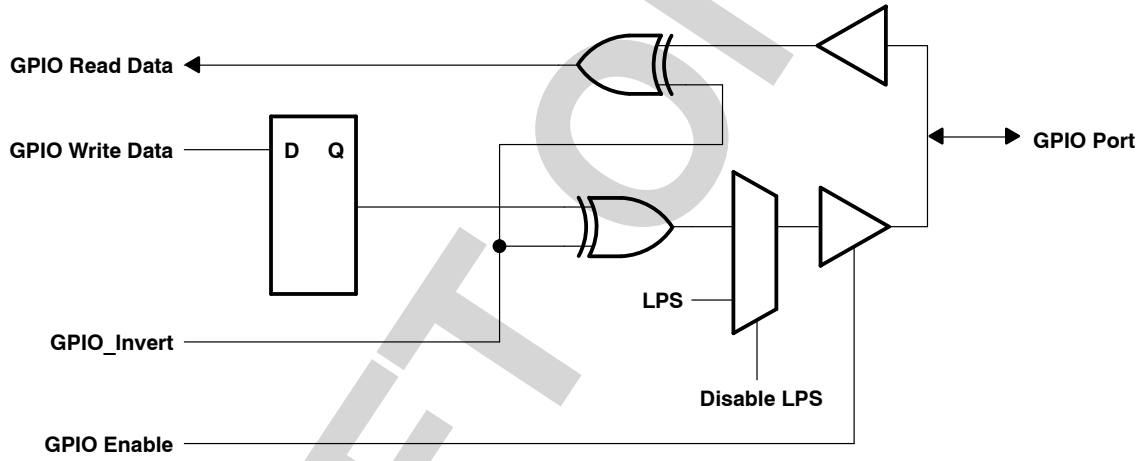


Figure 10–2. LPS/GPIO1

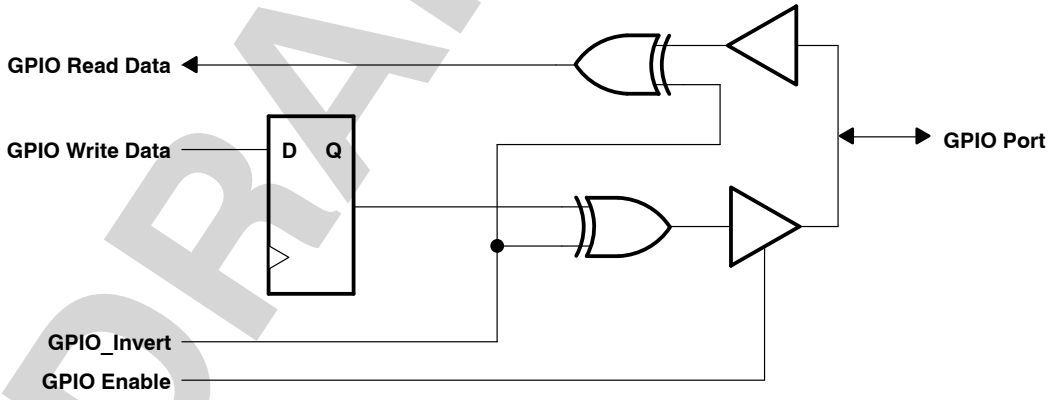


Figure 10–3. GPIO2 and GPIO3

11 Serial EEPROM

11.1 Serial Bus Interface

The PCI4450 provides a serial bus interface to initiallize the 1394 Global Unique ID register and a few PCI configuration registers through a serial EEPROM. The PCI4450 communicates with the serial EEPROM via the 2-wire serial interface.

After power-up the serial interface initializes the locations listed in Table 11–1. While the PCI4450 is accessing the serial ROM , all incoming PCI slave accesses are terminated with retry status. Table 11–2 shows the serial ROM memory map required for initializing the PCI4450 registers.

Table 11–1. Registers and Bits Loadable through Serial EEPROM

OFFSET	REGISTER	BITS LOADED FROM EEPROM
OHCI Register (24h)	1394 GlobalUniqueIDHi	31–0
OHCI Register(28h)	1394 GlobalUniqueIDLo	31–0
OHCI Register (50h)	Host Control Register	23
PCI Register (2Ch)	PCI Subsytem ID	15–0
PCI Register (2Dh)	PCI Vendor ID	15–0
PCI Register (3Eh)	PCI Maximum Latency, PCI Minimum Grant	15–0
PCI Register (F4h)	Link Enhancements Control Register	7, 2, 1

Table 11–2. Serial EEPROM Map

BYTE ADDRESS	BYTE DESCRIPTION							
00	PCI maximum latency (4'h0)				PCI_Minimum grant (4'h0)			
01	PCI vendor ID							
02	PCI vendor ID (ms byte)							
03	PCI subsytem ID (ls byte)							
04	PCI subsytem ID							
05	[7] Link_enhancement Control.enab_unfair	[6] HCControl. ProgramPhy Enable	[5] RSVD	[4] RSVD	[3] RSVD	[2] Link_enhancement Control.enab_ insert_idle	[1] Link_enhancement Control.enab_accel	[0] RSVD
06	Reserved							
07	1394 GlobalUniqueIDHi (ls byte 0)							
08	1394 GlobalUniqueIDHi (byte 1)							
09	1394 GlobalUniqueIDHi (byte 2)							
0A	1394 GlobalUniqueIDHi (ms byte 3)							
0B	1394 GlobalUniqueIDLo (ls byte 0)							
0C	1394 GlobalUniqueIDLo (byte 1)							
0D	1394 GlobalUniqueIDLo (byte 2)							
0E	1394 GlobalUniqueIDLo (ms byte 3)							

12Electrical Characteristics

12.1 Absolute Maximum Ratings Over Operating Temperature Ranges †

Supply voltage range, V_{CC} . . . . . –0.5 V to 4.6 V
Clamping voltage range, V_{CCP}, V_{CCA}, V_{CCB} . . . . . –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
Card B . . . . . –0.5 V to V_{CCB} + 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 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
Card B . . . . . –0.5 V to V_{CCB} + 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 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_{CCA} or V_{CCB}. ZV terminals and TTL signals are measured with respect to V_{CC}. 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_{CCA} or V_{CCB}. ZV terminals and TTL signals are measured with respect to V_{CC}. The limit specified applies for a dc condition.

12.2 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 voltage, ZV Port I/O voltage	Commercial	3.3 V	3	3.3	3.6	V
			5 V	4.5	5	5.5	
$V_{CCA/B}$	PC Card I/O 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 $V_{CCA/B}$		$V_{CCA/B}$	V
			5 V	2.4		$V_{CCA/B}$	
		PHY I/F		2		V_{CC}	V
		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 $V_{CCA/B}$	V
			5 V	0		0.8	
		PHY I/F		0		0.8	V
		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}	
		Fail safe§		0		V_{CC}	
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}	
		Fail safe§		0		V_{CC}	V
t_t	Input transition times (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

‡ TTL 4 mA pins are A_CVS1//A_$\overline{VS1}$, A_CVS2//A_$\overline{VS2}$, B_CVS1//B_$\overline{VS1}$, B_CVS2//B_$\overline{VS2}$, CLOCK, DATA, LATCH, SPKROUT, MFUNC4–MFUNC0, SCL, SDA, ZV_UVx, ZV_PCLK, ZV_SDATA, ZV_LRCLK, ZV_MCLK, ZV_VSYNC, ZV_HREF, ZV_SCLK, ZV_Yx, $\overline{SUSPEND}$, LPS, PHY_LREQ.
TTL 8 mA pins are MFUNC6, MFUNC5, PHY_CTL0, PHY_CTL1, PHY_DATAx, and LINKON.

§ Fail-safe pin is $\overline{RI\_OUT}$ (open drain).

¶ Applies to external output buffers

These junction temperatures reflect simulation conditions. The customer is responsible for verifying junction temperature.

NOTE 3: Unused pins (input or I/O) must be held high or low to prevent them from floating.

12.3 Electrical Characteristics Over Recommended Operating Conditions (unless otherwise noted)

PARAMETER		PINS	OPERATION	TEST CONDITIONS	MIN	MAX	UNIT
V_{OH}	High-level output voltage (see Note 4)	PCI	3.3 V	I_{OH} = –0.5 mA	0.9 V_{CC}		V
			5 V	I_{OH} = –2 mA	2.4		
		PC Card	3.3 V	I_{OH} = –0.15 mA	0.9 V_{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_{OL} = –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		0.1 V_{CC}	V
			5 V	I_{OL} = 6 mA		0.55	
		PC Card	3.3 V	I_{OL} = 0.7 mA		0.1 V_{CC}	
			5 V	I_{OL} = 0.7 mA		0.55	
		PHY I/F	3.3 V	I_{OL} = 4 mA		0.5	V
			3.3 V	I_{OL} = 8 mA		0.5	
		TTL		I_{OL} = 4 mA		0.5	V
				I_{OL} = 8 mA		0.5	
		$\overline{SERR}$		I_{OL} = 8 mA		0.5	V
I_{OZL}	3-state output, high-impedance state current (see Note 4)	Output pins	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 current	Output pins	3.6 V	V_I = V_{CC}†		10	µA
			5.25 V	V_I = V_{CC}†		25	
I_{IL}	Low-level input current	Input pins		V_I = GND		–1	µA
		I/O pins		V_I = GND		–10	
		Latch		V_I = GND		–2	
I_{IH}	High-level input current (see Note 5)	Input pins	3.6 V	V_I = V_{CC}‡		10	µA
			5.25 V	V_I = V_{CC}‡		20	
		I/O pins	3.6 V	V_I = V_{CC}‡		10	
			5.25 V	V_I = V_{CC}‡		25	
		Fail-safe pins	3.6 V	V_I = V_{CC}		10	

† For PCI pins, V_I = V_{CCP}. For PC Card pins, V_I = $V_{CC(A/B)}$. For ZV pins, V_I = V_{CC}. For miscellaneous pins, V_I = V_{CC}.

‡ For I/O pins, input leakage (I_{IL} and I_{IH}) includes I_{OZ} leakage of the disabled output.

NOTES: 4. V_{OH} and I_{OL} are not tested on $\overline{SERR}$ (GFN pin Y20, GJG pin W18)and $\overline{RI\_OUT}$ (GFN pin Y13, GJG pin R11) because they are open-drain outputs.

5. I_{IH} is not tested on LATCH (GFN pin W12, GJG pin W11) because it is pulled up with an internal resistor.

12.4 PCI Clock/Reset Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-air Temperature

(see Figure 12–2 and Figure 12–3)

PARAMETER		ALTERNATE SYMBOL	TEST CONDITIONS	MIN	MAX	UNIT
t_c	Cycle time, PCLK	t_{cyc}		30		ns
t_{wH}	Pulse duration, PCLK high	t_{high}		11		ns
t_{wL}	Pulse duration, 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, RSTIN	t_{rst}		1		ms
t_{su}	Setup time, PCLK active at end of $\overline{\text{RSTIN}}$	$t_{rst\text{-}clk}$		100		µs

12.5 PCI Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-air Temperature

(see Note 7, Figure 12–1, and Figure 12–4)

PARAMETER			ALTERNATE SYMBOL	TEST CONDITIONS	MIN	MAX	UNIT
t_{pd}	Propagation delay time, See Note 6	PCLK-to-shared signal valid delay time	t_{val}	C_L = 50 pF, See Note 7		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

NOTES: 6. 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.

7. This data sheet uses the following conventions to describe time (t) intervals. The format is t_A, where subscript *A* indicates the type of dynamic parameter being represented. One of the following is used: t_{pd} = propagation delay time, t_d = delay time, t_{su} = setup time, and t_h = hold time.

12.6 Switching Characteristics For PHY-Link Interface

PARAMETER		MEASURED	MIN	TYP	MAX	UNIT
t_{su}	Setup time, Dn, CTLn, LREQ to PHY_CLK	–50% to 50%	6			ns
t_h	Hold time, Dn, CTLn, LREQ before PHY_CLK	–50% to 50%	1			
t_d	Delay time, PHY_CLK to Dn, CTLn	–50% to 50%	2		11	

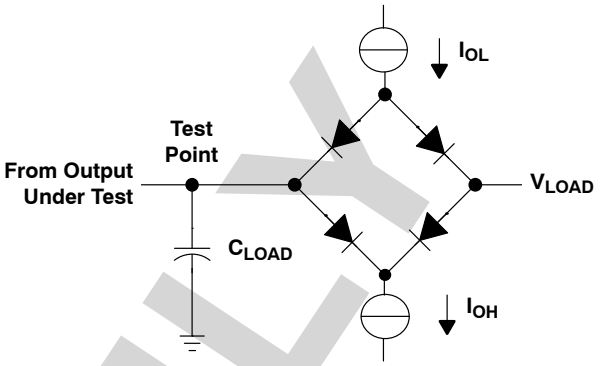
12.7 Parameter Measurement Information

LOAD CIRCUIT PARAMETERS

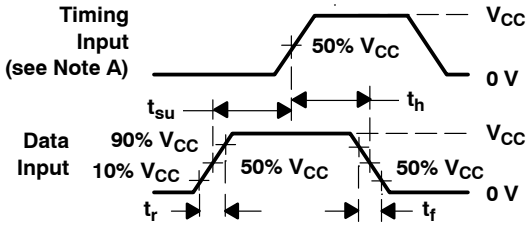
TIMING PARAMETER		C_{LOAD}† (pF)	I_{OL} (mA)	I_{OH} (mA)	V_{LOAD}‡ (V)
t_{en}	t_{PZH}	50	8	–8	0
	t_{PZL}				3
t_{dis}	t_{PHZ}	50	8	–8	1.5
	t_{PLZ}				
t_{pd}		50	8	–8	‡

† C_{LOAD} includes the typical load-circuit distributed capacitance.

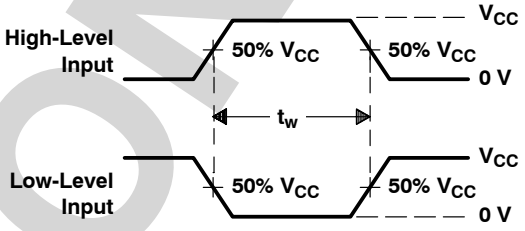
‡ $\frac{V_{LOAD} - V_{OL}}{I_{OL}}$ = 50 Ω, where V_{OL} = 0.6 V, I_{OL} = 8 mA



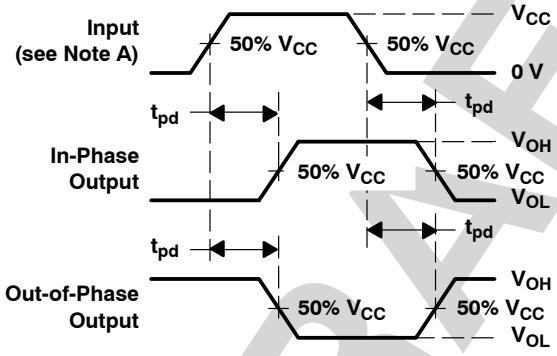
LOAD CIRCUIT



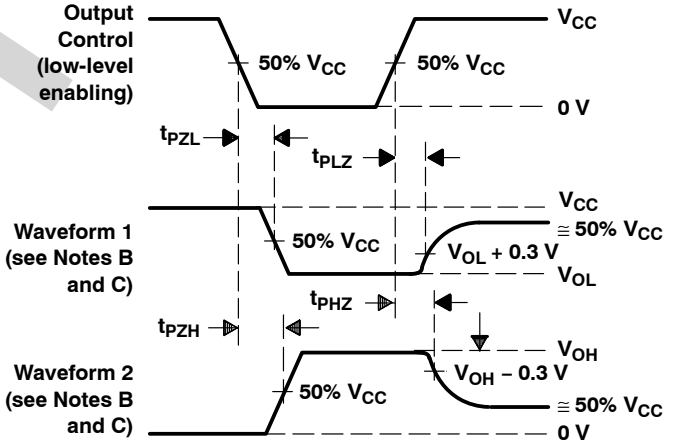
VOLTAGE WAVEFORMS
SETUP AND HOLD TIMES
INPUT RISE AND FALL TIMES



VOLTAGE WAVEFORMS
PULSE DURATION



VOLTAGE WAVEFORMS
PROPAGATION DELAY TIMES



VOLTAGE WAVEFORMS
ENABLE AND DISABLE TIMES, 3-STATE OUTPUTS

NOTES: A. Phase relationships between waveforms were chosen arbitrarily. All input pulses are supplied by pulse generators having the following characteristics: PRR = 1 MHz, Z_O = 50 Ω, t_r = 6 ns.
B. Waveform 1 is for an output with internal conditions such that the output is low except when disabled by the output control. Waveform 2 is for an output with internal conditions such that the output is high except when disabled by the output control.
C. For t_{PLZ} and t_{PHZ}, V_{OL} and V_{OH} are measured values.

Figure 12–1. Load Circuit and Voltage Waveforms

12.8 PCI Bus Parameter Measurement Information

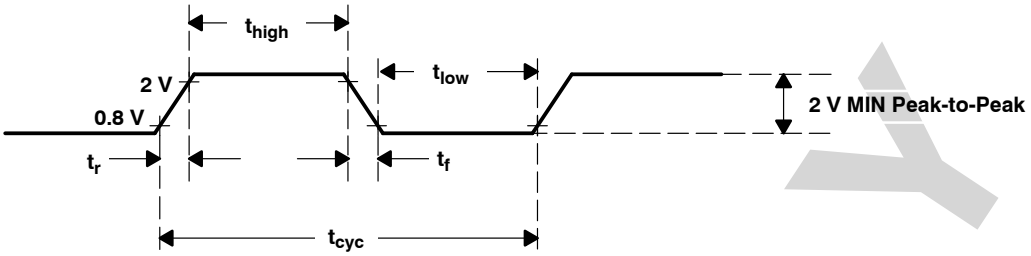


Figure 12–2. PCLK Timing Waveform

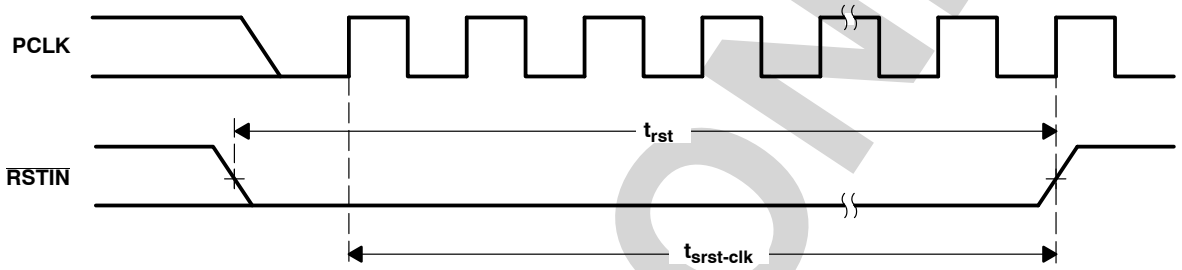


Figure 12–3. RSTIN Timing Waveforms

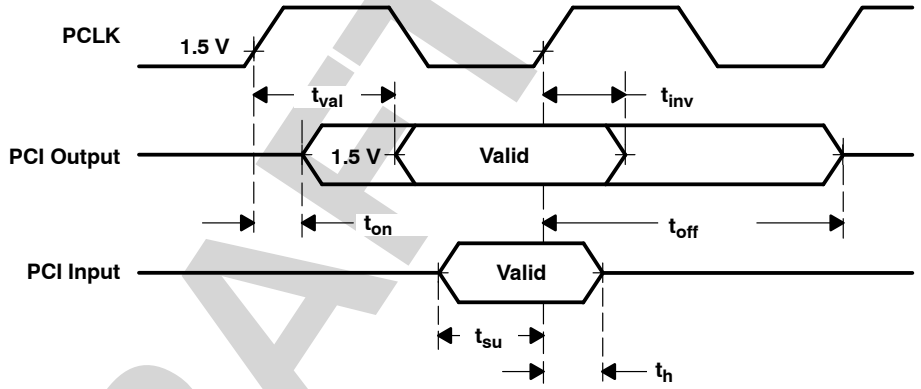


Figure 12–4. Shared Signals Timing Waveforms

12.9 PC Card Cycle Timing

The PC Card cycle timing is controlled by the wait-state bits in the Intel 82365SL-DF compatible memory and I/O window registers. The PC Card cycle generator uses the PCI clock to generate the correct card address setup and hold times and the PC Card command active (low) interval. This allows the cycle generator to output PC Card cycles that are as close to the Intel 82365SL-DF timing as possible, while always slightly exceeding the Intel 82365SL-DF values. This ensures compatibility with existing software and maximizes throughput.

The PC Card address setup and hold times are a function of the wait-state bits. Table 12–1 shows address setup time in PCLK cycles and nanoseconds for I/O and memory cycles. Table 12–2 and Table 12–3 show command active time in PCLK cycles and nanoseconds for I/O and memory cycles. Table 12–4 shows address hold time in PCLK cycles and nanoseconds for I/O and memory cycles.

Table 12–1. PC Card Address Setup Time, $t_{su(A)}$, 8-Bit and 16-Bit PCI Cycles

WAIT-STATE BITS			TS1 – 0 = 01 (PCLK/ns)
I/O			3/90
Memory	WS1	0	2/60
Memory	WS1	1	4/120

Table 12–2. PC Card Command Active Time, $t_{c(A)}$, 8-Bit PCI Cycles

WAIT-STATE BITS			TS1 – 0 = 01 (PCLK/ns)
	WS	ZWS	
I/O	0	0	19/570
	1	X	23/690
	0	1	7/210
Memory	00	0	19/570
	01	X	23/690
	10	X	23/690
	11	X	23/690
	00	1	7/210

Table 12–3. PC Card Command Active Time, $t_{c(A)}$, 16-Bit PCI Cycles

WAIT-STATE BITS			TS1 – 0 = 01 (PCLK/ns)
	WS	ZWS	
I/O	0	0	7/210
	1	X	11/330
	0	1	N/A
Memory	00	0	9/270
	01	X	13/390
	10	X	17/510
	11	X	23/630
	00	1	5/150

Table 12–4. PC Card Address Hold Time, $t_{h(A)}$, 8-Bit and 16-Bit PCI Cycles

WAIT-STATE BITS			TS1 – 0 = 01 (PCLK/ns)
I/O			2/60
Memory	WS1	0	2/60
Memory	WS1	1	3/90

12.10 Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-air Temperature, Memory Cycles (for 100-ns Common Memory)

(see Note 8 and Figure 12–5)

		ALTERNATE SYMBOL	MIN	MAX	UNIT
t_{su}	Setup time, $\overline{CE1}$ and $\overline{CE2}$ before $\overline{WE}$/$\overline{OE}$ low	T1	60		ns
t_{su}	Setup time, CA25–CA0 before $\overline{WE}$/$\overline{OE}$ low	T2	$t_{su(A)}$+2PCLK		ns
t_{su}	Setup time, $\overline{REG}$ before $\overline{WE}$/$\overline{OE}$ low	T3	90		ns
t_{pd}	Propagation delay time, $\overline{WE}$/$\overline{OE}$ low to $\overline{WAIT}$ low	T4			ns
t_w	Pulse duration, $\overline{WE}$/$\overline{OE}$ low	T5	200		ns
t_h	Hold time, $\overline{WE}$/$\overline{OE}$ low after $\overline{WAIT}$ high	T6			ns
t_h	Hold time, $\overline{CE1}$ and $\overline{CE2}$ after $\overline{WE}$/$\overline{OE}$ high	T7	120		ns
t_{su}	Setup time (read), CDATA15–CDATA0 valid before $\overline{OE}$ high	T8			ns
t_h	Hold time (read), CDATA15–CDATA0 valid after $\overline{OE}$ high	T9	0		ns
t_h	Hold time, CA25–CA0 and $\overline{REG}$ after $\overline{WE}$/$\overline{OE}$ high	T10	$t_{h(A)}$+1PCLK		ns
t_{su}	Setup time (write), CDATA15–CDATA0 valid before $\overline{WE}$ low	T11	60		ns
t_h	Hold time (write), CDATA15–CDATA0 valid after $\overline{WE}$ low	T12	240		ns

NOTE 8: These times are dependent on the register settings associated with ISA wait states and data size. They are also dependent on cycle type (read/write, memory/I/O) and $\overline{WAIT}$ from PC Card. The times listed here represent absolute minimums (the times that would be observed if programmed for zero wait state, 16-bit cycles) with a 33-MHz PCI clock.

12.11 Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-air Temperature, I/O Cycles

(see Figure 12–6)

		ALTERNATE SYMBOL	MIN	MAX	UNIT
t_{su}	Setup time, $\overline{REG}$ before $\overline{IORD}$/$\overline{IOWR}$ low	T13	60		ns
t_{su}	Setup time, $\overline{CE1}$ and $\overline{CE2}$ before $\overline{IORD}$/$\overline{IOWR}$ low	T14	60		ns
t_{su}	Setup time, CA25–CA0 valid before $\overline{IORD}$/$\overline{IOWR}$ low	T15	$t_{su(A)}$+2PCLK		ns
t_{pd}	Propagation delay time, $\overline{IOIS16}$ low after CA25–CA0 valid	T16		35	ns
t_{pd}	Propagation delay time, $\overline{IORD}$ low to $\overline{WAIT}$ low	T17	35		ns
t_w	Pulse duration, $\overline{IORD}$/$\overline{IOWR}$ low	T18	T_{cA}		ns
t_h	Hold time, $\overline{IORD}$ low after $\overline{WAIT}$ high	T19			ns
t_h	Hold time, $\overline{REG}$ low after $\overline{IORD}$ high	T20	0		ns
t_h	Hold time, $\overline{CE1}$ and $\overline{CE2}$ after $\overline{IORD}$/$\overline{IOWR}$ high	T21	120		ns
t_h	Hold time, CA25–CA0 after $\overline{IORD}$/$\overline{IOWR}$ high	T22	$t_{h(A)}$+1PCLK		ns
t_{su}	Setup time (read), CDATA15–CDATA0 valid before $\overline{IORD}$ high	T23	10		ns
t_h	Hold time (read), CDATA15–CDATA0 valid after $\overline{IORD}$ high	T24	0		ns
t_{su}	Setup time (write), CDATA15–CDATA0 valid before $\overline{IOWR}$ low	T25	90		ns
t_h	Hold time (write), CDATA15–CDATA0 valid after $\overline{IOWR}$ high	T26	90		ns

12.12 Switching Characteristics Over Recommended Ranges of Supply Voltage and Operating Free-air Temperature, Miscellaneous

(see Figure 12–7)

PARAMETER			ALTERNATE SYMBOL	MIN	MAX	UNIT
t_{pd}	Propagation delay time	BVD2 low to SPKROUT low	T27		30	ns
		BVD2 high to SPKROUT high			30	
		$\overline{IREQ}$ to IRQ15–IRQ3	T28		30	
		$\overline{STSCHG}$ to IRQ15–IRQ3			30	

12.13 PC Card Parameter Measurement Information

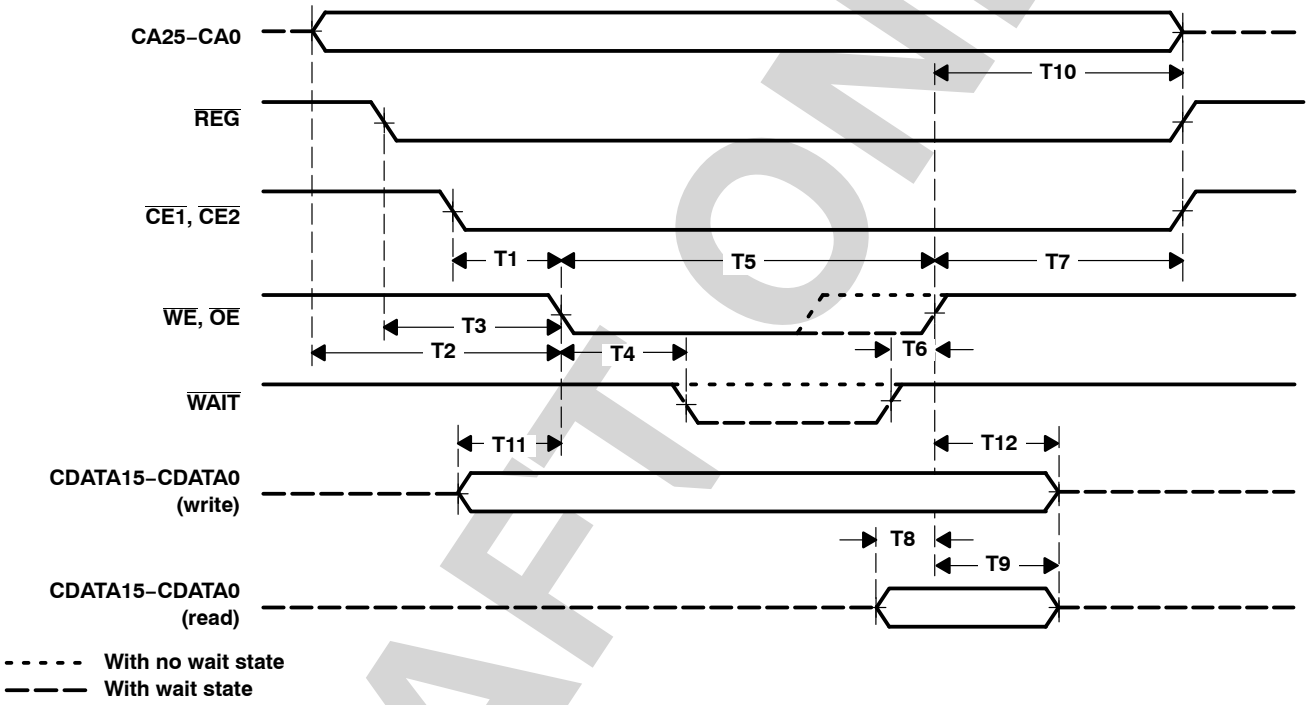


Figure 12–5. PC Card Memory Cycle

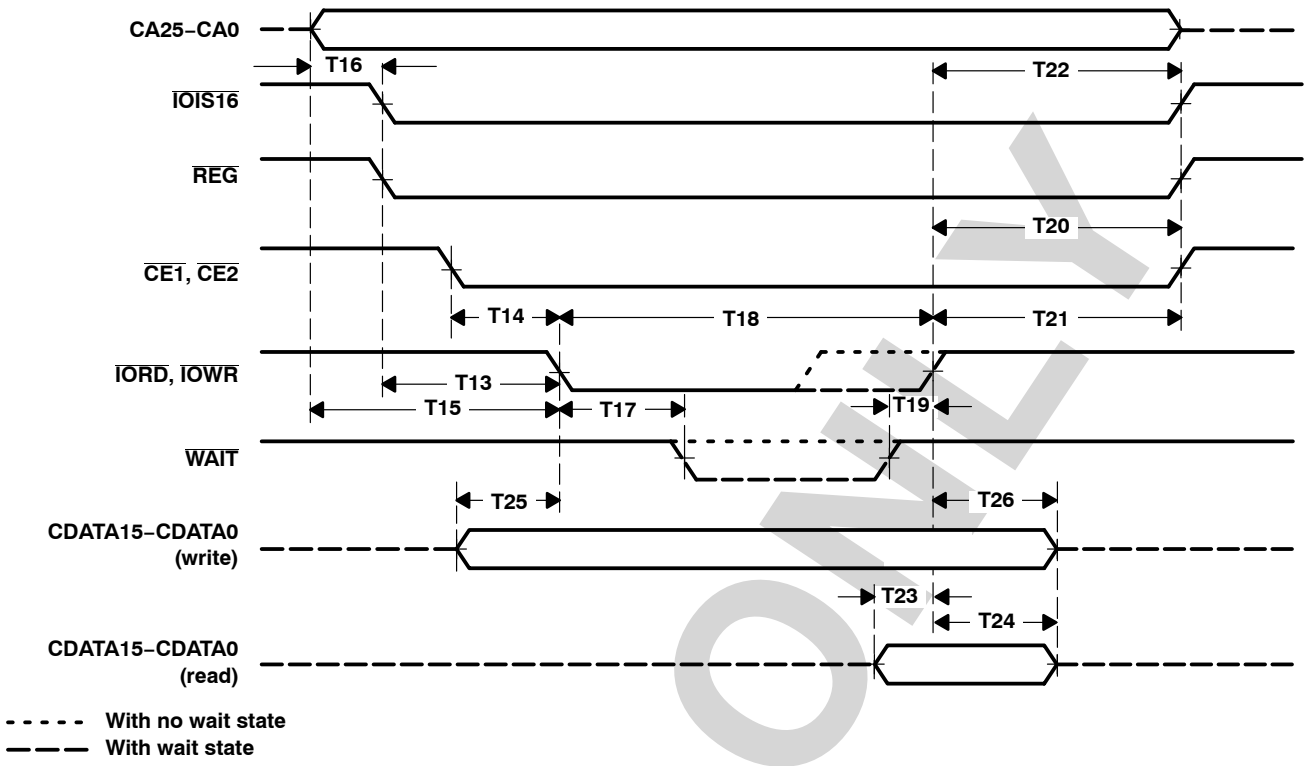


Figure 12–6. PC Card I/O Cycle

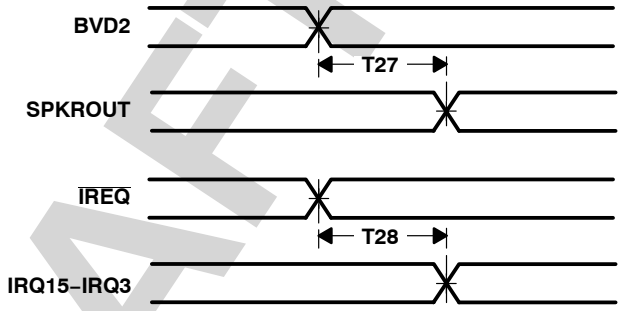
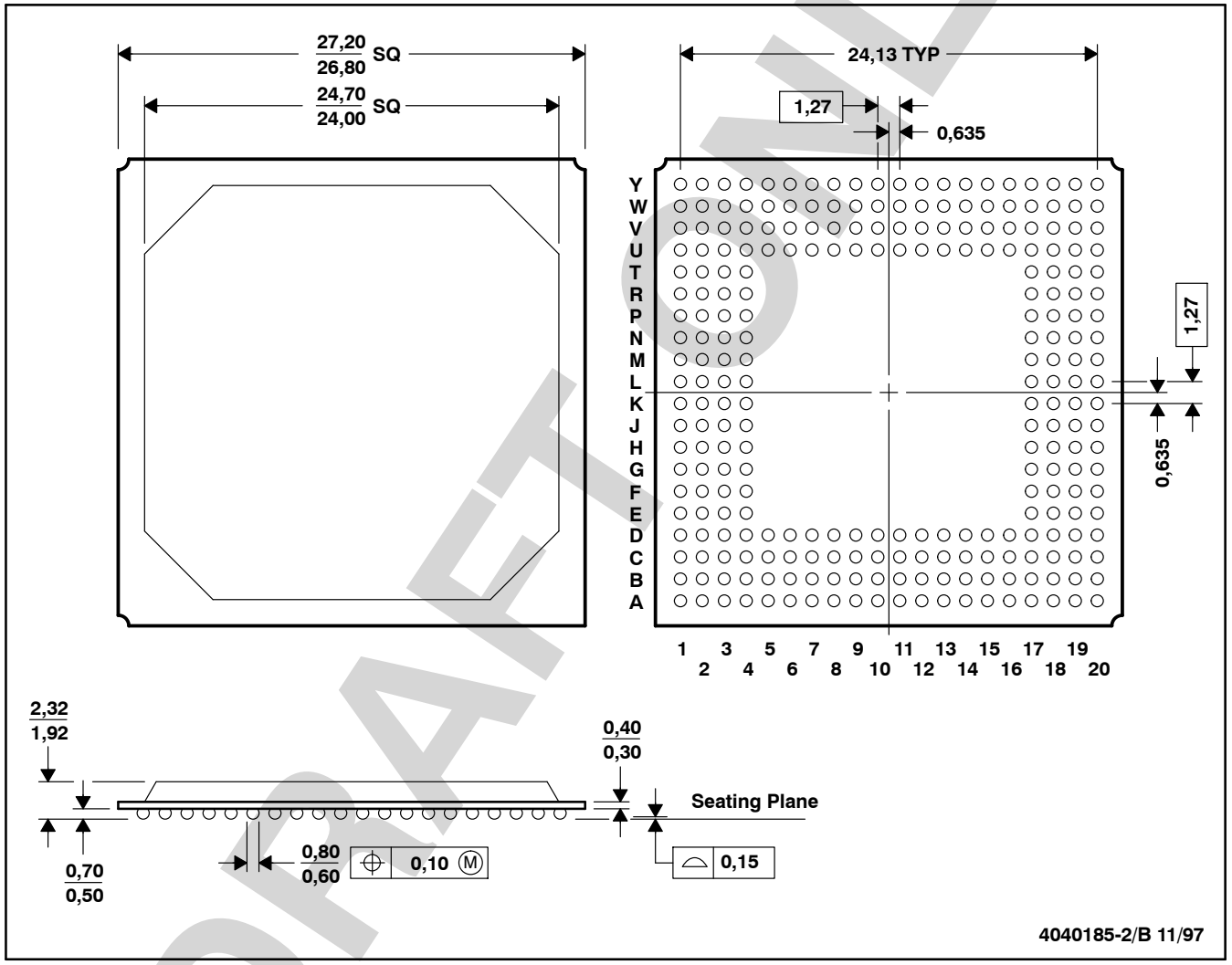


Figure 12–7. Miscellaneous PC Card Delay Times

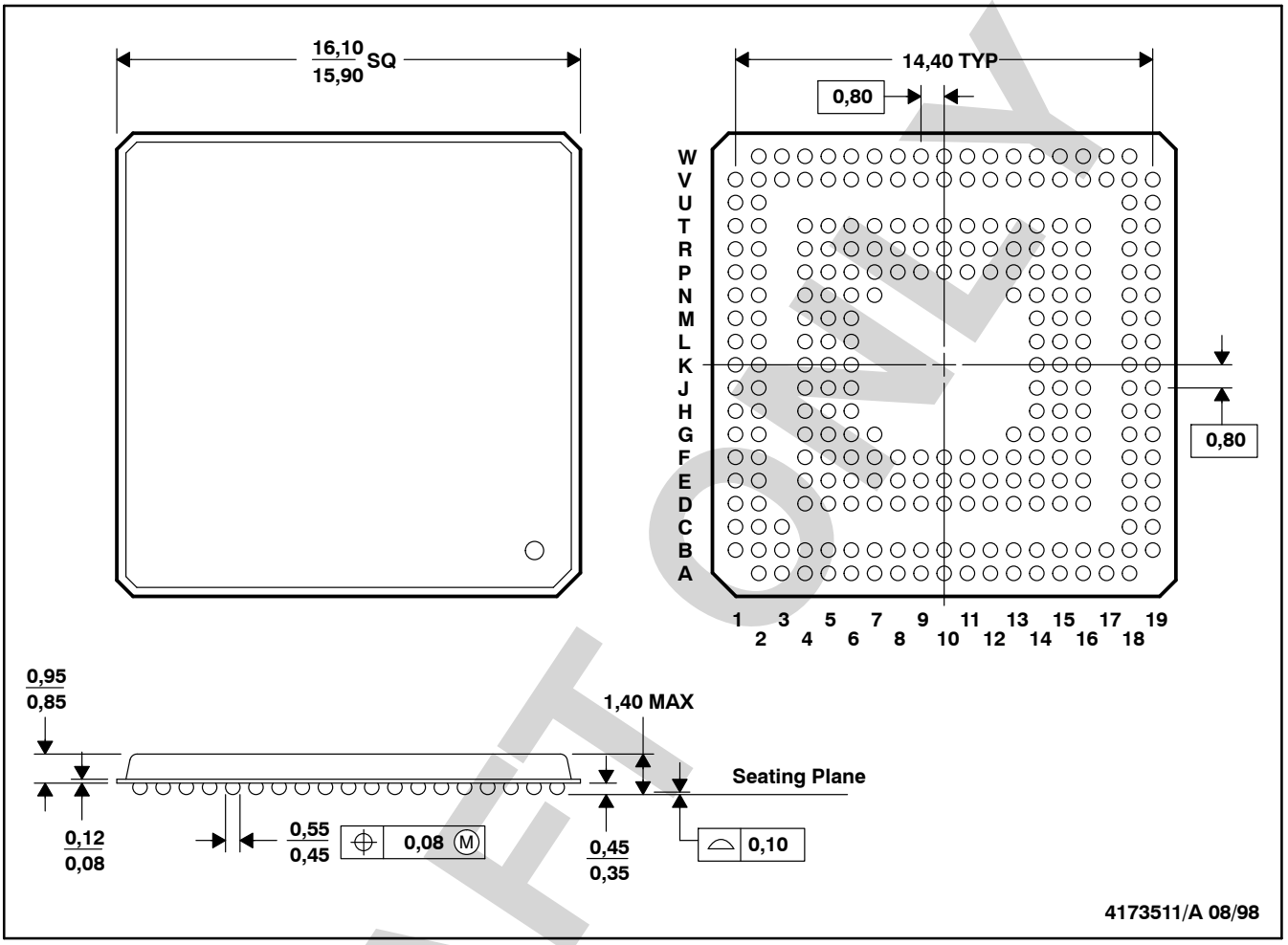
13Mechanical Data

GFN (S-PBGA-N256) PLASTIC BALL GRID ARRAY



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.

GJG (S-PBGA-N257) 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

Micro Star is a trademark of Texas Instruments Incorporated.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

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

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Mobile Processors	www.ti.com/omap		
Wireless Connectivity	www.ti.com/wirelessconnectivity		
	TI E2E Community Home Page		e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265