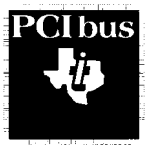
THE DATASHEET OF

PCI1450GJG

www.win-source.net 0086-755-83957316

- 1997 PC Standard Compliant
- PCI Bus Power Management Interface Specification 1.0
- ACPI 1.0 Compliant
- PCI Local Bus Specification Revision 2.1/2.2 Compliant
- PC 98/99 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 Specification
- Ultra Zoomed Video
- Zoomed Video Auto-Detect
- Advanced filtering on Card Detect Lines Provide 90 Microseconds of Noise Immunity.
- Programmable D3 Status Pin
- 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
- 8-Way Legacy IRQ Multiplexing
- Interrupt Modes Supported: Serial ISA/Serial PCI, Serial ISA/Parallel PCI, Parallel ISA/Parallel PCI, Parallel PCI Only.
- EEPROM Interface for Loading Subsystem ID and Subsystem Vendor ID
- Supports Zoomed Video with Internal Buffering
- Dedicated Pin for PCI $\overline{\text{CLKRUN}}$
- Four General Purpose I/O's
- 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 Pins
- Supports PCI Bus Lock ($\overline{\text{LOCK}}$)
- Packaged in a 256-pin BGA or 257-pin Micro-Star BGA

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.



Intel is a trademark of Intel Corporation.
PC Card is a trademark of Personal Computer Memory Card International Association (PCMCIA).
TI is a trademark of Texas Instruments Incorporated.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 1998, Texas Instruments Incorporated

Table of Contents

Description 3
Terminal Assignments 4
Signal Names and Terminal Assignments 5
PCI1450 System Block Diagram 9
Terminal Functions 10
I/O Characteristics 23
Clamping Rail Voltages 23
PCI Interface 23
PC Card Applications Overview 26
Programmable Interrupt Subsystem 34
Power Management Overview 38
PC Card Controller Programming Model 44
PCI Configuration Registers (Functions 0 and 1) 44
ExCA Compatibility Registers (Functions 0 and 1) 82
CardBus Socket Registers (Functions 0 and 1) 106
Distributed DMA (DDMA) Registers 114
Absolute Maximum Ratings 120
Recommended Operating Conditions 121
Electrical Characteristics 122
PCI Clock/Reset Timing Requirements 123
PCI Timing Requirements 123
Parameter Measurement Information 124
PCI Bus Parameter Measurement Information 125
PC Card Cycle Timing 126
Timing Requirements, (Memory Cycles) 127
Timing Requirements, (I/O Cycles) 127
Switching Characteristics (Miscellaneous) 128
PC Card Parameter Mesasurement Information 128
Mechanical Data 130

description

The Texas Instruments PCI1450 is a high-performance PC Card controller with a 32-bit PCI interface. The device supports two independent PC Card sockets compliant with the 1997 PC Card Standard and the *PCI Bus Interface Specification for PCI-to-CardBus Bridges*. The PCI1450 provides a rich feature set which makes it the best choice for bridging between PCI and PC Cards in both notebook and desktop computers. The 1995 and 1997 PC Card Standards retain the 16-bit PC Card specification defined in PCMCIA Release 2.1, and defines the new 32-bit PC Card, CardBus, capable of full 32-bit data transfers at 33 MHz. The PCI1450 supports any combination of 16-bit and CardBus PC Cards in the two sockets, powered at 5 Vdc or 3.3 Vdc as required.

The PCI1450 is compliant with the latest *PCI Bus Power Management Specification*. It is also compliant with the *PCI Local Bus Specification Revision 2.1*, and its PCI interface can act as either a PCI master device or a PCI slave device. The PCI bus mastering is initiated during 16-bit PC Card DMA transfers, or CardBus PC Card bridging transactions.

All card signals are internally buffered to allow hot insertion and removal. The PCI1450 is register compatible with the Intel 82365SL-DF ExCA controller. The PCI1450 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 PCI1450 can also be programmed to accept fast posted writes to improve system bus utilization.

The PCI1450 provides an internally buffered zoom video (ZV) path. This reduces the design effort of PC board manufacturers to add a ZV compatible solution and guarantees compliance with the CardBus loading specifications. Multiple system interrupt signaling options are provided: Serial ISA/Serial PCI, Serial ISA/Parallel PCI, Parallel ISA/Parallel PCI, and PCI Only interrupts. Furthermore, general-purpose inputs and outputs (GPIOs) are provided for the board designer to implement sideband functions. Many other features are designed into the PCI1450 such as socket activity LED outputs, and are discussed in detail throughout the design specification.

An advanced complementary metal-oxide semiconductor (CMOS) process achieves low system power consumption while operating at PCI clock rates up to 33 MHz. Several low-power modes allow the host power management system to further reduce power consumption.

Unused PCI1450 inputs must be pulled up using a 43 kΩ resistor.

use of symbols in this document

Throughout this data sheet the overbar symbol denotes an active-low signal. For example: $\overline{\text{FRAME}}$ denotes that this is an active-low signal.

1.

terminal assignments

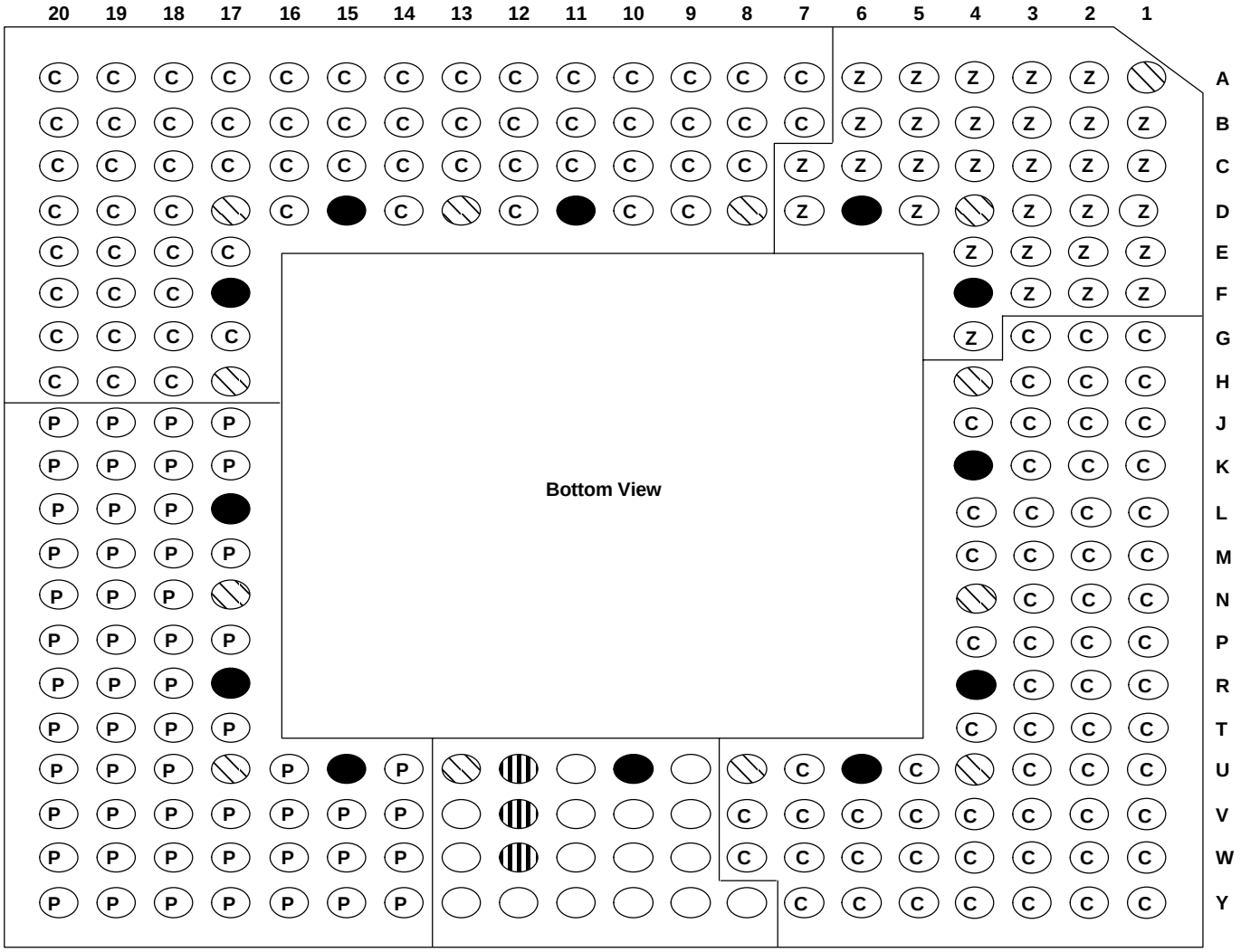


Figure 1. PCI1450 Pin Diagram

signal names and terminal assignments

Signal names and their terminal assignments are shown in Tables 1 and 2 and are sorted alphanumerically by the assigned terminal.

Table 1. GFN Terminals Sorted Alphanumerically for CardBus and 16-bit Signals

GFN	SIGNAL NAME	GFN	SIGNAL NAME	GFN	SIGNAL NAME
A1	GND	C9	B_CCD2//B_CD2	G3	A_CCD1//A_CD1
A2	ZV_UV3	C10	V_{CCB}	G17	B_CAD7//B_D7
A3	ZV_Y7	C11	B_CAD26//B_A0	G18	B_CAD6//B_D13
A4	V_{CCZ}	C12	B_CAD23//B_A3	G19	B_CAD4//B_D12
A5	ZV_Y1	C13	B_CRST//B_RESET	G20	B_CAD1//B_D4
A6	ZV_HREF	C14	B_CAD19//B_A25	H1	A_CAD3//A_D5
A7	B_RSVD//B_D2	C15	B_CFRAME//B_A23	H2	A_CAD4//A_D12
A8	B_CAD28//B_D8	C16	B_CTRDY//B_A22	H3	A_CAD1//A_D4
A9	B_CSTSCHG//B_BVD1(STSCHG/RI)	C17	B_CSTOP//B_A20	H4	GND
A10	B_CINT//B_READY(IREQ)	C18	B_CAD16//B_A17	H17	GND
A11	B_CVS1//B_VS1	C19	B_CAD15//B_IOWR	H18	B_CAD2//B_D11
A12	B_CAD24//B_A2	C20	B_CAD11//B_OE	H19	B_CAD0//B_D3
A13	B_CAD22//B_A4	D1	V_{CCZ}	H20	B_CCD1//B_CD1
A14	B_CAD20//B_A6	D2	ZV_UV7	J1	A_CAD7//A_D7
A15	B_CAD18//B_A7	D3	ZV_MCLK	J2	A_RSVD//A_D14
A16	B_CIRDY//B_A15	D4	GND	J3	A_CAD5//A_D6
A17	B_CCLK//B_A16	D5	ZV_UV0	J4	A_CAD6//A_D13
A18	B_CDEVSEL//B_A21	D6	V_{CC}	J17	PCLK
A19	B_CPAR//B_A13	D7	ZV_Y2	J18	CLKRUN
A20	B_RSVD//B_A18	D8	GND	J19	PRST
B1	ZV_UV5	D9	B_CAD27//B_D0	J20	GNT
B2	ZV_UV4	D10	B_CAUDIO//B_BVD2(SPKR)	K1	A_CC/BE0//A_CE1
B3	ZV_UV1	D11	V_{CC}	K2	V_{CCA}
B4	ZV_Y6	D12	B_CREQ//B_INPACK	K3	A_CAD8//A_D15
B5	ZV_Y4	D13	GND	K4	V_{CC}
B6	ZV_Y0	D14	B_CC/BE2//B_A12	K17	REQ
B7	B_CAD31//B_D10	D15	V_{CC}	K18	AD31
B8	B_CAD29//B_D1	D16	B_CGNT//B_WE	K19	AD30
B9	B_CCLKRUN//B_WP(IOIS16)	D17	GND	K20	V_{CCP}
B10	B_CSERR//B_WAIT	D18	B_CAD12//B_A11	L1	A_CAD9//A_A10
B11	B_CAD25//B_A1	D19	B_CAD10//B_CE2	L2	A_CAD10//A_CE2
B12	B_CC/BE3//B_REG	D20	B_CC/BE0//B_CE1	L3	A_CAD11//A_OE
B13	B_CAD21//B_A5	E1	ZV_PCLK	L4	A_CAD13//A_IORD
B14	B_CVS2//B_VS2	E2	ZV_SDATA	L17	V_{CC}
B15	B_CAD17//B_A24	E3	ZV_LRCLK	L18	AD28
B16	V_{CCB}	E4	ZV_RSVD0	L19	AD27
B17	B_CPERR//B_A14	E17	B_CAD13//B_IORD	L20	AD29
B18	B_CBLOCK//B_A19	E18	B_CAD9//B_A10	M1	A_CAD12//A_A11
B19	B_CC/BE1//B_A8	E19	B_CAD8//B_D15	M2	A_CAD15//A_IOWR
B20	B_CAD14//B_A9	E20	B_RSVD//B_D14	M3	A_CAD14//A_A9
C1	ZV_RSVD1	F1	G_RST	M4	A_CAD16//A_A17
C2	ZV_SCLK	F4	V_{CC}	M17	C/BE3
C3	ZV_UV6	F17	V_{CC}	M18	AD24
C4	ZV_UV2	F18	V_{CCB}	M19	AD25
C5	ZV_Y5	F19	B_CAD5//B_D6	M20	AD26
C6	ZV_Y3	F20	B_CAD3//B_D5	N1	A_CC/BE1//A_A8
C7	ZV_VSYNC	G1	A_CAD2//A_D11	N2	A_RSVD//A_A18
C8	B_CAD30//B_D9	G2	A_CAD0//A_D3	N3	A_CPAR//A_A13

signal names and terminal assignments (continued)

Table 1. GFN Terminals Sorted Alphanumerically for CardBus and 16-bit Signals (Continued)

GFN	SIGNAL NAME	GFN	SIGNAL NAME	GFN	SIGNAL NAME
N4	GND	U9	IRQMUX1	W6	A_CCD2//A_CD2
N17	GND	U10	V_{CC}	W7	A_CAD29//A_D1
N18	AD22	U11	PCGNT/IRQMUX6	W8	A_CAD31//A_D10
N19	AD23	U12	CLOCK	W9	IRQMUX3
N20	IDSEL	U13	GND	W10	IRQMUX5
P1	A_CBLOCK//A_A19	U14	AD6	W11	GPIO1/LEDA2
P2	A_CPERR//A_A14	U15	V_{CC}	W12	LATCH
P3	A_CGNT//A_WE	U16	AD11	W13	IRQSER/INTB
P4	A_CTRDY//A_A22	U17	GND	W14	AD1
P17	AD17	U18	PERR	W15	AD4
P18	V_{CCP}	U19	SERR	W16	AD7
P19	AD20	U20	TRDY	W17	AD9
P20	AD21	V1	A_CAD18//A_A7	W18	AD13
R1	A_CSTOP//A_A20	V2	A_CAD20//A_A6	W19	C/BE1
R2	A_CDEVSEL//A_A21	V3	A_CAD21//A_A5	W20	V_{CCP}
R3	V_{CCA}	V4	A_CAD25//A_A1	Y1	A_CREQ//A_INPACK
R4	V_{CC}	V5	A_CSERR//A_WAIT	Y2	A_CC/BE3//A_REG
R17	V_{CC}	V6	A_CSTSCHG//A_BVD1(STSCHG/RI)	Y3	A_CVS1//A_VS1
R18	AD16	V7	A_CAD28//A_D8	Y4	A_CINT//A_READY(IREQ)
R19	AD18	V8	A_RSVD//A_D2	Y5	A_CAUDIO//A_BVD2(SPKR)
R20	AD19	V9	IRQMUX2	Y6	A_CAD27//A_D0
T1	A_CCLK//A_A16	V10	V_{CCI}	Y7	A_CAD30//A_D9
T2	A_CIRDY//A_A15	V11	GPIO0/LEDA1	Y8	IRQMUX0
T3	A_CC/BE2//A_A12	V12	DATA	Y9	IRQMUX4
T4	A_CAD19//A_A25	V13	GPIO3/INTA	Y10	SPKROUT
T17	STOP	V14	AD3	Y11	SUSPEND
T18	IRDY	V15	V_{CCP}	Y12	PCREQ/IRQMUX7
T19	FRAME	V16	AD8	Y13	RI_OUT/PME
T20	C/BE2	V17	AD12	Y14	AD0
U1	A_CFRAME//A_23	V18	AD15	Y15	AD2
U2	A_CAD17//A_A24	V19	GPIO2/LOCK	Y16	AD5
U3	A_CVS2//A_VS2	V20	DEVSEL	Y17	C/BE0
U4	GND	W1	A_CRST//A_RESET	Y18	AD10
U5	A_CAD26//A_A0	W2	A_CAD22//A_A4	Y19	AD14
U6	V_{CC}	W3	A_CAD23//A_A3	Y20	PAR
U7	A_CCLKRUN//A_WP(IOIS16)	W4	A_CAD24//A_A2		
U8	GND	W5	V_{CCA}		

signal names and terminal assignments (continued)

Table 2. GJG Terminals Sorted Alphanumerically for CardBus and 16-bit Signals

NO.	SIGNAL NAME	NO.	SIGNAL NAME	NO.	SIGNAL NAME
A2	ZV_UV6	D12	B_CREQ//B_INPACK	G16	GND
A3	ZV_UV4	D13	B_CRST//B_RESET	G18	B_CAD5//B_D6
A4	ZV_UV2	D14	B_CC/BE2//B_A12	G19	B_CAD6//B_D13
A5	ZV_Y6	D15	B_CCLK//B_A16	H1	A_CAD5//A_D6
A6	ZV_Y3	D16	B_CAD16//B_A17	H2	A_RSVD//A_D14
A7	ZV_VSYNC	D18	B_CAD15//B_IOWR	H4	A_CAD7//A_D7
A8	V_{CC}	D19	B_CAD12//B_A11	H5	GND
A9	B_CCLKRUN//B_WP(IOIS16)	E1	ZV_SDATA	H6	A_CAD6//A_D13
A10	B_CSERR//B_WAIT	E2	ZV_PCLK	H14	B_CCD1//B_CD1
A11	B_CAD24//B_A2	E4	V_{CCZ}	H15	B_CAD4//B_D12
A12	B_CAD21//B_A5	E5	ZV_LRCLK	H16	B_CAD1//B_D4
A13	B_CAD20//B_A6	E6	ZV_Y5	H18	B_CAD2//B_D11
A14	B_CAD17//B_A24	E7	ZV_Y1	H19	B_CAD0//B_D3
A15	V_{CCB}	E8	B_CAD31//B_D10	J1	A_CAD8//A_D15
A16	B_CGNT//B_WE	E9	B_CAD28//B_D8	J2	A_CC/BE0//A_CE1
A17	B_CPERR//B_A14	E10	B_CSTSCHG//B_BVD1(STSCHG/RI	J4	V_{CCA}
A18	B_CBLOCK//B_A19	E11	B_CAD26//B_A0	J5	A_CAD9//A_A10
B1	ZV_SCLK	E12	B_CAD23//B_A3	J6	V_{CC}
B2	ZV_UV5	E13	B_CAD19//B_A25	J14	GNT
B3	ZV_UV3	E14	B_CFRAME//B_A23	J15	PCLK
B4	ZV_UV1	E15	B_CTRDY//B_A22	J16	CLKRUN
B5	ZV_Y7	E16	B_CAD13//B_IORD	J18	PRST
B6	ZV_Y4	E18	B_CAD11//B_OE	J19	GND
B7	ZV_Y0	E19	B_CAD10//B_CE2	K1	A_CAD11//A_OE
B8	B_CAD30//B_D9	F1	A_CCD1//A_CD1	K2	A_CAD13//A_IORD
B9	B_CCD2//B_CD2	F2	A_CAD0//A_D3	K4	A_CAD12//A_A11
B10	V_{CCB}	F4	G_RST	K5	GND
B11	B_CAD25//B_A1	F5	GND	K6	A_CAD10//A_CE2
B12	B_CAD22//B_A4	F6	V_{CC}	K14	V_{CCP}
B13	B_CVS2//B_VS2	F7	ZV_Y2	K15	REQ
B14	GND	F8	B_CAD29//B_D1	K16	AD31
B15	B_CIRDY//B_A15	F9	GND	K18	AD30
B16	B_CDEVSEL//B_A21	F10	B_CINT//B_READY(IREQ)	K19	V_{CC}
B17	B_CSTOP//B_A20	F11	B_CVS1//B_VS1	L1	A_CAD14//A_A9
B18	B_CPAR//B_A13	F12	V_{CC}	L2	A_CAD16//A_A17
B19	B_RSVD//B_A18	F13	B_CAD18//B_A7	L4	A_CC/BE1//A_A8
C1	ZV_UV7	F14	VCC	L5	A_RSVD//A_A18
C2	ZV_MCLK	F15	B_CAD9//B_A10	L6	A_CAD15//A_IOWR
C18	B_CC/BE1//B_A8	F16	B_CC/BE0//B_CE1	L14	AD29
C19	B_CAD14//B_A9	F18	B_CAD8//B_D15	L15	AD28
D1	ZV_RSVD0	F19	V_{CCB}	L16	AD25
D2	ZV_RSVD1	G1	A_CAD4//A_D12	L18	AD27
D4	GND	G2	V_{CC}	L19	AD26
D5	ZV_UV0	G4	A_CAD3//A_D5	M1	A_CPAR//A_A13
D6	V_{CCZ}	G5	A_CAD1//A_D4	M2	A_CBLOCK//A_A19
D7	GND	G6	A_CAD2//A_D11	M4	A_CPERR//A_A14
D8	B_RSVD//B_D2	G7	ZV_HREF	M5	A_CSTOP//A_A20
D9	B_CAD27//B_D0	G13	B_CAD3//B_D5	M6	V_{CC}
D10	B_CAUDIO//B_BVD2(SPKR)	G14	B_CAD7//B_D7	M14	AD22
D11	B_CC/BE3//B_REG	G15	B_RSVD//B_D14	M15	AD24

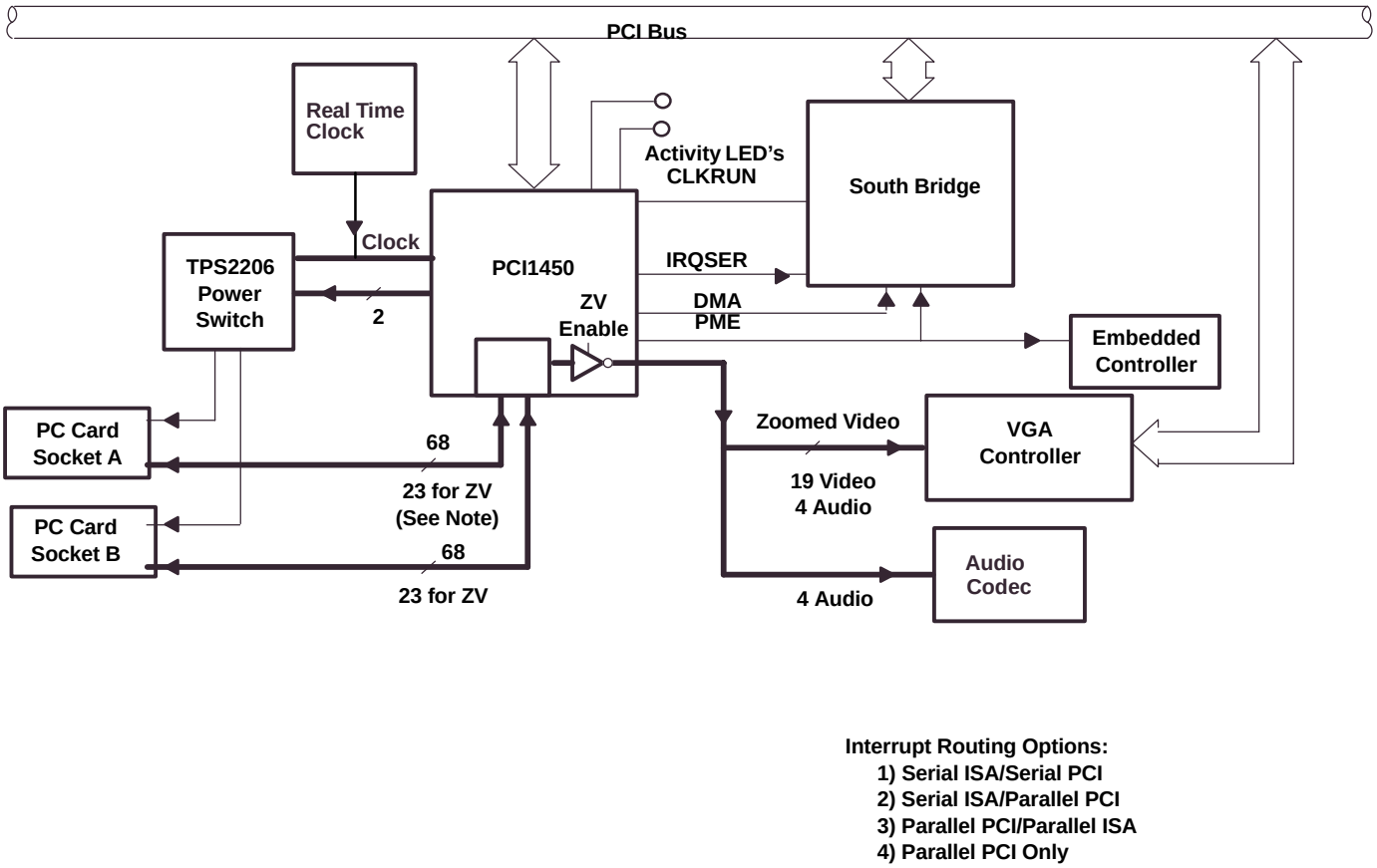
signal names and terminal assignments (continued)

Table 2. GJG Terminals Sorted Alphanumerically for CardBus and 16-bit Signals. (Continued)

NO.	SIGNAL NAME	NO.	SIGNAL NAME	NO.	SIGNAL NAME
M16	CBE3	R6	A_CSTSCHG//A_BVD1(STSCHG/RI)	V3	A_CAD25//A_A1
M18	IDSEL	R7	A_CAD28//A_D8	V4	A_CVS1//A_VS1
M19	AD23	R8	IRQMUX2	V5	A_CAUDIO//A_BVD2(SPKR)
N1	A_CDEVSEL//A_A21	R9	IRQMUX5	V6	GND
N2	GND	R10	PCGNT/IRQMUX6	V7	A_CAD29//A_D1
N4	A_CCLK//A_A16	R11	CLOCK	V8	IRQMUX0
N5	A_CTRDY//A_A22	R12	AD0	V9	GND
N6	V_{CCA}	R13	GND	V10	GPI01/LEDA2
N7	A_CGNT//A_WE	R14	C/BE0	V11	LATCH
N13	AD1	R15	V_{CC}	V12	V_{CC}
N14	GND	R16	TRDY	V13	AD3
N15	AD19	R18	FRAME	V14	V_{CCP}
N16	AD21	R19	IRDY	V15	AD10
N18	V_{CCP}	T1	A_CAD20//A_A6	V16	AD13
N19	AD20	T2	A_CRST//A_RESET	V17	C/BE1
P1	A_CIRDY//A_A15	T4	A_CAD21//A_A5	V18	V_{CCP}
P2	A_CFRAME//A_23	T5	A_CINT//A_READY(IREQ)	V19	GPI02/LOCK
P4	A_CC/BE2//A_A12	T6	A_CCLKRUN//A_WP(IOIS16)	W2	A_CC/BE3//A_REG
P5	V_{CC}	T7	A_RSVD//A_D2	W3	A_CAD24//A_A2
P6	A_CAD17//A_A24	T8	IRQMUX1	W4	A_CAD26//A_A0
P7	A_CAD27//A_D0	T9	IRQMUX3	W5	V_{CCA}
P8	V_{CC}	T10	GPIO0/LEDA1	W6	A_CCD2//A_CD2
P9	V_{CCI}	T11	DATA	W7	A_CAD30//A_D9
P10	SPKROUT	T12	GPI03/INTA	W8	A_CAD31//A_D10
P11	PCREQ/IRQMUX7	T13	AD4	W9	IRQMUX4
P12	RI_OUT/PME	T14	AD7	W10	SUSPEND
P13	AD5	T15	AD11	W11	GND
P14	AD8	T16	AD15	W12	IRQSER/INTB
P15	C/BE2	T18	DEVSEL	W13	AD2
P16	AD16	T19	STOP	W14	AD6
P18	AD18	U1	GND	W15	AD9
P19	AD17	U2	A_CAD22//A_A4	W16	AD12
R1	A_CAD18//A_A7	U18	PERR	W17	AD14
R2	A_CAD19//A_A25	U19	SERR	W18	PAR
R4	A_CVS2//A_VS2	V1	A_CREQ//A_INPACK		
R5	A_CSERR//A_WAIT	V2	A_CAD23//A_A3		

PCI1450 System Block Diagram

Figure 2 shows a simplified system implementation example using the PCI1450. The PCI interface includes all address/data and control signals for PCI protocol. Highlighted in this diagram is the functionality supported by the PCI1450. The PCI1450 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 IRQMUX, and 4 GPIO pins) that can be programmed for a wide variety of functions.



NOTE: 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 2. PCI1450 System Block Diagram

terminal functions

This section describes the PCI1450 terminal functions. The terminals are grouped in tables by functionality such as PCI system function, power supply function, etc., for quick reference. The terminal numbers are also listed for convenient reference.

Table 3. Power Supply

TERMINAL			FUNCTION
NAME	GFN NO.	GJG NO.	
GND	A1, D4, D8, D13, D17, H4, H17, N4, N17, U4, U8, U13, U17	B14, D4, D7, F5, F9, G16, H5, J19, K5, N2, N14, R13, U1, V6, V9, W11	Device ground terminals
V_{CC}	D6, D11, D15, F4, F17, K4, L17, R4, R17, U6, U10, U15	A8, F6, F12, F14, G2, J6, K19, M6, P5, P8, R15, V12	Power supply terminal for core logic (3.3 Vdc)
V_{CCA}	K2, R3, W5	J4, N6, W5	Clamp voltage for PC Card A interface. Indicates Card A signaling environment.
V_{CCB}	B16, C10, F18,	A15, B10, F19	Clamp voltage for PC Card B interface. Indicates Card B signaling environment.
V_{CCI}	V10	P9	Clamp voltage for interrupt subsystem interface and miscellaneous I/O. Indicates signaling level of the following inputs and shared outputs: IRQSER, PCGNT, PCREQ, SUSPEND, SPKROUT, GPIO1:0, IRQMUX7–IRQMUX0, INTA, INTB, CLOCK, DATA, LATCH, and RI_OUT.
V_{CCP}	K20, P18, V15, W20	K14, N18, V14, V18	Clamp voltage for PCI signaling (3.3 Vdc or 5 Vdc)
V_{CCZ}	A4, D1	D6, E4	Clamp voltage for zoom video interface (3.3 Vdc or 5 Vdc) and G_RST

Table 4. PC Card Power Switch

TERMINAL			I/O TYPE	FUNCTION
NAME	GFN NO.	GJG NO.		
CLOCK	U12	R11	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 of the *system control register* (PCI offset 80h) to a 1b. A 43kΩ pulldown resistor should be tied to this terminal.
DATA	V12	T11	O	3-line power switch data. DATA is used to serially communicate socket power-control information to the power switch.
LATCH	W12	V11	O	3-line power switch latch. LATCH is asserted by the PCI1450 to indicate to the PC Card power switch that the data on the DATA line is valid.

terminal functions (continued)

Table 5. PCI System

TERMINAL NAME	TERMINAL GFN NO.	TERMINAL GJG NO.	I/O TYPE	FUNCTION
$\overline{\text{CLKRUN}}$	J18	J16	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 PCI1450 responds accordingly. If $\overline{\text{CLKRUN}}$ is not implemented, then this pin should be tied low. $\overline{\text{CLKRUN}}$ is enabled by default by bit 1 (KEEPCLK) in the *system control register*.
PCLK	J17	J15	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.
$\overline{\text{PRST}}$	J19	J18	I	PCI reset. When the PCI bus reset is asserted, $\overline{\text{PRST}}$ causes the PCI1450 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 PCI1450 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.
$\overline{\text{G\_RST}}$	F1	F4	I	Global reset. When the global reset is asserted, the $\overline{\text{G\_RST}}$ signal causes the PCI1450 to 3-state all output buffers and reset all internal registers. When $\overline{\text{G\_RST}}$ is asserted, the device is completely in its default state. For systems that require wake-up from D3, $\overline{\text{G\_RST}}$ 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, $\overline{\text{G\_RST}}$ should be tied to $\overline{\text{PRST}}$.

terminal functions (continued)

Table 6. PCI Address and Data

TERMINAL NAME	TERMINAL GFN NO.	TERMINAL GJG NO.	I/O TYPE	FUNCTION
AD31	K18	K16	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	K19	K18		
AD29	L20	L14		
AD28	L18	L15		
AD27	L19	L18		
AD26	M20	L19		
AD25	M19	L16		
AD24	M18	M15		
AD23	N19	M19		
AD22	N18	M14		
AD21	P20	N16		
AD20	P19	N19		
AD19	R20	N15		
AD18	R19	P18		
AD17	P17	P19		
AD16	R18	P16		
AD15	V18	T16		
AD14	Y19	W17		
AD13	W18	V16		
AD12	V17	W16		
AD11	U16	T15		
AD10	Y18	V15		
AD9	W17	W15		
AD8	V16	P14		
AD7	W16	T14		
AD6	U14	W14		
AD5	Y16	P13		
AD4	W15	T13		
AD3	V14	V13		
AD2	Y15	W13		
AD1	W14	N13		
AD0	Y14	R12		
C/$\overline{BE3}$	M17	M16	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/$\overline{BE2}$	T20	P15		
C/$\overline{BE1}$	W19	V17		
C/$\overline{BE0}$	Y17	R14		
PAR	Y20	W18	I/O	PCI bus parity. In all PCI bus read and write cycles, the PCI1450 calculates even parity across the AD31–AD0 and C/$\overline{BE3}$–C/$\overline{BE0}$ buses. As an initiator during PCI cycles, the PCI1450 outputs this parity indicator with a one-PCLK delay. As a target during PCI cycles, the calculated parity is compared to the initiator's parity indicator. A compare error results in the assertion of a parity error ($\overline{PERR}$).

terminal functions (continued)

Table 7. PCI Interface Control

TERMINAL NAME	TERMINAL GFN NO.	TERMINAL GJG NO.	I/O TYPE	FUNCTION
$\overline{\text{DEVSEL}}$	V20	T18	I/O	PCI device select. The PCI1450 asserts $\overline{\text{DEVSEL}}$ to claim a PCI cycle as the target device. As a PCI initiator on the bus, the PCI1450 monitors $\overline{\text{DEVSEL}}$ until a target responds. If no target responds before timeout occurs, then the PCI1450 terminates the cycle with an initiator abort.
$\overline{\text{FRAME}}$	T19	R18	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.
$\overline{\text{GNT}}$	J20	J14	I	PCI bus grant. $\overline{\text{GNT}}$ is driven by the PCI bus arbiter to grant the PCI1450 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.
GPIO2/$\overline{\text{LOCK}}$	V19	V19	I/O	PCI bus general-purpose I/O pins or PCI bus lock. GPIO2/$\overline{\text{LOCK}}$ can be configured as PCI $\overline{\text{LOCK}}$ and used to gain exclusive access downstream. Since this functionality is not typically used, a general-purpose I/O may be accessed through this terminal. GPIO2/$\overline{\text{LOCK}}$ defaults to a general-purpose input and can be configured through the *GPIO2 control register*.
IDSEL	N20	M18	I	Initialization device select. IDSEL selects the PCI1450 during configuration space accesses. IDSEL can be connected to one of the upper 24 PCI address lines on the PCI bus.
$\overline{\text{IRDY}}$	T18	R19	I/O	PCI initiator ready. $\overline{\text{IRDY}}$ indicates the PCI bus initiator's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCLK where both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. Until $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are both sampled asserted, wait states are inserted.
$\overline{\text{PERR}}$	U18	U18	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*.
$\overline{\text{REQ}}$	K17	K15	O	PCI bus request. $\overline{\text{REQ}}$ is asserted by the PCI1450 to request access to the PCI bus as an initiator.
$\overline{\text{SERR}}$	U19	U19	O	PCI system error. $\overline{\text{SERR}}$ is an output that is pulsed from the PCI1450 when enabled through the *command register*, indicating a system error has occurred. The PCI1450 need not be the target of the PCI cycle to assert this signal. When $\overline{\text{SERR}}$ is enabled in the *bridge control register*, this signal also pulses, indicating that an address parity error has occurred on a CardBus interface.
$\overline{\text{STOP}}$	T17	T19	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.
$\overline{\text{TRDY}}$	U20	R16	I/O	PCI target ready. $\overline{\text{TRDY}}$ indicates the primary bus target's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCLK when both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. Until both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted, wait states are inserted.

terminal functions (continued)

Table 8. System Interrupt

TERMINAL NAME	TERMINAL GFN NO.	TERMINAL GJG NO.	I/O TYPE	FUNCTION
GPIO3/$\overline{\text{INTA}}$	V13	T12	I/O	Parallel PCI interrupt. $\overline{\text{INTA}}$ can be optionally mapped to GPI03 when parallel PCI interrupts are used. See *programmable interrupt subsystem* for details on interrupt signaling. GPIO3/$\overline{\text{INTA}}$ defaults to a general-purpose input.
IRQSER/$\overline{\text{INTB}}$	W13	W12	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 *programmable interrupt subsystem* for details on interrupt signaling. This terminal can be used to signal PCI $\overline{\text{INTB}}$ when one of the parallel interrupt modes is selected in the *device control register*.
IRQMUX7 IRQMUX6 IRQMUX5 IRQMUX4 IRQMUX3 IRQMUX2 IRQMUX1 IRQMUX0	Y12 U11 W10 Y9 W9 V9 U9 Y8	P11 R10 R9 W9 T9 R8 T8 V8	O	Interrupt request/secondary functions multiplexed. The primary function of these terminals is to provide the ISA-type IRQ signaling supported by the PCI1450. These interrupt multiplexer outputs can be mapped to any of 15 IRQs. The *device control register* must be programmed for the ISA IRQ interrupt mode and the *IRQMUX routing register* must have the IRQ routing programmed before these terminals are enabled. All of these terminals have secondary functions, such as PCI $\overline{\text{INTB}}$, PC/PCI DMA request/grant, ring indicate output, and zoom 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 IRQ routing. See the *IRQMUX routing register* for programming options.
$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	Y13	P12	O	Ring indicate out and power management event output. Terminal provides an output for ring-indicate or $\overline{\text{PME}}$ signals.

Table 9. PC/PCI DMA

TERMINAL NAME	TERMINAL GFN NO.	TERMINAL GJG NO.	I/O TYPE	FUNCTION
$\overline{\text{PCGNT}}$/ IRQMUX6	U11	R10	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. Interrupt request MUX 6. When configured for IRQMUX6, this terminal provides the IRQMUX6 interrupt output of the interrupt multiplexer, and can be mapped to any of 15 ISA-type IRQs. IRQMUX6 takes precedence over $\overline{\text{PCGNT}}$, and should not be enabled in a system using PC/PCI DMA. This terminal is also used for the serial EEPROM interface.
$\overline{\text{PCREQ}}$/ IRQMUX7	Y12	P11	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. Interrupt request MUX 7. When configured for IRQMUX7, this terminal provides the IRQMUX7 interrupt output of the interrupt multiplexer, and can be mapped to any of 15 ISA-type IRQs. IRQMUX7 takes precedence over $\overline{\text{PCREQ}}$, and should not be enabled in a system using PC/PCI DMA. This terminal is also used for the serial EEPROM interface.

terminal functions (continued)

Table 10. Zoom Video

TERMINAL NAME	TERMINAL GFN NO.	TERMINAL GJG NO.	I/O AND MEMORY INTERFACE SIGNAL	I/O TYPE	FUNCTION
ZV_HREF	A6	G7	A10	O	Horizontal sync to the zoom video port
ZV_VSYNC	C7	A7	A11	O	Vertical sync to the zoom video port
ZV_Y7 ZV_Y6 ZV_Y5 ZV_Y4 ZV_Y3 ZV_Y2 ZV_Y1 ZV_Y0	A3 B4 C5 B5 C6 D7 A5 B6	B5 A5 E6 B6 A6 F7 E7 B7	A20 A14 A19 A13 A18 A8 A17 A9	O	Video data to the zoom video port in YV:4:2:2 format
ZV_UV7 ZV_UV6 ZV_UV5 ZV_UV4 ZV_UV3 ZV_UV2 ZV_UV1 ZV_UV0	D2 C3 B1 B2 A2 C4 B3 D5	C1 A2 B2 A3 B3 A4 B4 D5	A25 A12 A24 A15 A23 A16 A22 A21	O	Video data to the zoom video port in YV:4:2:2 format
ZV_SCLK	C2	B1	A7	O	Audio SCLK PCM
ZV_MCLK	D3	C2	A6	O	Audio MCLK PCM
ZV_PCLK	E1	E2	$\overline{\text{IOIS16}}$	O	Pixel clock to the zoom video port
ZV_LRCLK	E3	E5	$\overline{\text{INPACK}}$	O	Audio LRCLK PCM
ZV_SDATA	E2	E1	$\overline{\text{SPKR}}$	O	Audio SDATA PCM
ZV_RSVD1 ZV_RSVD0	C1 E4	D2 D1	A5 A4	O	Reserved. No connection in the PC Card. ZV_RSVD1 and ZV_RSVD0 are put into the high-impedance state by host adapter.

terminal functions (continued)

Table 11. Miscellaneous

TERMINAL NAME	TERMINAL GFN NO.	TERMINAL GJG NO.	I/O TYPE	FUNCTION
GPIO0/LEDA1	V11	T10	I/O	GPIO0/socket activity LED indicator 1. When GPIO0/LEDA1 is configured as LEDA1, it provides an output indicating PC Card socket 0 activity. Otherwise, GPIO0/LEDA1 can be configured as a general-purpose input and output, GPIO0. The zoom video enable signal (ZV_STAT) can also be routed to this signal through the *GPIO0 control register*. GPIO0/LEDA1 defaults to a general-purpose input.
GPIO1/LEDA2	W11	V10	I/O	GPIO1/socket activity LED indicator 2. When GPIO1/LEDA2 is configured as LEDA2, it provides an output indicating PC Card socket 1 activity. Otherwise, GPIO1/LEDA2 can be configured as a general-purpose input and output, GPIO1. A CSC interrupt can be generated on a GPDATA change, and this input can be used for power switch overcurrent (OC) sensing. See *GPIO1 control register* for programming details. GPIO1/LEDA2 defaults to a general-purpose input.
SPKROUT	Y10	P10	O	Speaker output. SPKROUT is the output to the host system that can carry $\overline{\text{SPKR}}$ or CAUDIO through the PCI1450 from the PC Card interface. SPKROUT is driven as the XOR combination of card $\overline{\text{SPKR}}$//CAUDIO inputs.
$\overline{\text{SUSPEND}}$	Y11	W10	I	Suspend. $\overline{\text{SUSPEND}}$ is used to protect the internal registers from clearing when $\overline{\text{PRST}}$ is asserted. See *suspend mode* for details.

terminal functions (continued)

Table 12. 16-bit PC Card Address and Data (slots A and B)

TERMINAL					I/O TYPE	FUNCTION
	GFN NO.		GJG NO.			
NAME	SLOT A†	SLOT B‡	SLOT A†	SLOT B‡		
A25	T4	C14	R2	E13	O	PC Card address. 16-bit PC Card address lines. A25 is the most-significant bit.
A24	U2	B15	P6	A14		
A23	U1	C15	P2	E14		
A22	P4	C16	N5	E15		
A21	R2	A18	N1	B16		
A20	R1	C17	M5	B17		
A19	P1	B18	M2	A18		
A18	N2	A20	L5	B19		
A17	M4	C18	L2	D16		
A16	T1	A17	N4	D15		
A15	T2	A16	P1	B15		
A14	P2	B17	M4	A17		
A13	N3	A19	M1	B18		
A12	T3	D14	P4	D14		
A11	M1	D18	K4	D19		
A10	L1	E18	J5	F15		
A9	M3	B20	L1	C19		
A8	N1	B19	L4	C18		
A7	V1	A15	R1	F13		
A6	V2	A14	T1	A13		
A5	V3	B13	T4	A12		
A4	W2	A13	U2	B12		
A3	W3	C12	V2	E12		
A2	W4	A12	W3	A11		
A1	V4	B11	V3	B11		
A0	U5	C11	W4	E11		
D15	K3	E19	J1	F18	I/O	PC Card data. 16-bit PC Card data lines. D15 is the most-significant bit.
D14	J2	E20	H2	G15		
D13	J4	G18	H6	G19		
D12	H2	G19	G1	H15		
D11	G1	H18	G6	H18		
D10	W8	B7	W8	E8		
D9	Y7	C8	W7	B8		
D8	V7	A8	R7	E9		
D7	J1	G17	H4	G14		
D6	J3	F19	H1	G18		
D5	H1	F20	G4	G13		
D4	H3	G20	G5	H16		
D3	G2	H19	F2	H19		
D2	V8	A7	T7	D8		
D1	W7	B8	V7	F8		
D0	Y6	D9	P7	D9		

† Terminal name for slot A is preceded with A_. For example, the full name for terminal T4 is A_A25.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminal C14 is B_A25.

terminal functions (continued)

Table 13. 16-bit PC Card Interface Control (slots A and B)

NAME	TERMINAL GFN NO. SLOT A†	TERMINAL GFN NO. SLOT B‡	TERMINAL GJG NO. SLOT A†	TERMINAL GJG NO. SLOT B‡	I/O TYPE	FUNCTION
BVD1 (STSCHG/RI)	V6	A9	R6	E10	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 *ExCA card status-change interrupt configuration register* for the enable bits. See *ExCA card status-change register* and the *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)	Y5	D10	V5	D10	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 *ExCA card status-change interrupt configuration register* for enable bits. See *ExCA card status-change register* and the *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 PCI1450 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	G3 W6	H20 C9	F1 W6	H14 B9	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 *ExCA interface status register*.
CE1 CE2	K1 L2	D20 D19	J2 K6	F16 E19	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	Y1	D12	V1	D12	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, the PC Card asserts this signal to indicate a request for a DMA operation.
IORD	L4	E17	K2	E16	O	I/O read. IORD is asserted by the PCI1450 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 PCI1450 asserts IORD during DMA transfers from the PC Card to host memory.
IOWR	M2	C19	L6	D18	O	I/O write. IOWR is driven low by the PCI1450 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 PCI1450 asserts IOWR during transfers from host memory to the PC Card.

terminal functions (continued)

Table 13. 16-bit PC Card Interface Control (slots A and B) (continued)

NAME	TERMINAL GFN NO. SLOT A†	TERMINAL GFN NO. SLOT B‡	TERMINAL GJG NO. SLOT A†	TERMINAL GJG NO. SLOT B‡	I/O TYPE	FUNCTION
$\overline{OE}$	L3	C20	K1	E18	O	Output enable. $\overline{OE}$ is driven low by the PCI1450 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 PCI1450 asserts $\overline{OE}$ to indicate TC for a DMA write operation.
READY ($\overline{IREQ}$)	Y4	A10	T5	F10	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.
$\overline{REG}$	Y2	B12	W2	D11	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 PCI1450 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	W1	C13	T2	D13	O	PC Card reset. RESET forces a hard reset to a 16-bit PC Card.
$\overline{WAIT}$	V5	B10	R5	A10	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.
$\overline{WE}$	P3	D16	N7	A16	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 PCI1450 asserts $\overline{WE}$ to indicate TC for a DMA read operation.
$\overline{WP}$ ($\overline{IOIS16}$)	U7	B9	T6	A9	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.
$\overline{VS1}$ $\overline{VS2}$	Y3 U3	A11 B14	V4 R4	F11 B13	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.

† Terminal name for slot A is preceded with A_. For example, the full name for terminal P3 is A_$\overline{WE}$.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminal D16 is B_$\overline{WE}$.

terminal functions (continued)

Table 14. CardBus PC Card Interface System (slots A and B)

NAME	TERMINAL GFN NO. SLOT A†	TERMINAL GFN NO. SLOT B‡	TERMINAL GJG NO. SLOT A†	TERMINAL GJG NO. SLOT B‡	I/O TYPE	FUNCTION
CCLK	T1	A17	N4	D15	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, CCD2:1, and CVS2–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.
$\overline{\text{CCLKRUN}}$	U7	B9	T6	A9	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 PCI1450 to indicate that the CCLK frequency is decreased. CardBus clock run ($\overline{\text{CCLKRUN}}$) follows the PCI clock run ($\overline{\text{CLKRUN}}$).
$\overline{\text{CRST}}$	W1	C13	T2	D13	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 3-stated, and the PCI1450 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 terminal T1 is A_CCLK.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminal A17 is B_CCLK.

terminal functions (continued)

Table 15. CardBus PC Card Address and Data (slots A and B)

NAME	TERMINAL GFN NO. SLOT A†	TERMINAL GFN NO. SLOT B‡	TERMINAL GJG NO. SLOT A†	TERMINAL GJG NO. SLOT B‡	I/O TYPE	FUNCTION
CAD31	W8	B7	W8	E8	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	Y7	C8	W7	B8		
CAD29	W7	B8	V7	F8		
CAD28	V7	A8	R7	E9		
CAD27	Y6	D9	P7	D9		
CAD26	U5	C11	W4	E11		
CAD25	V4	B11	V3	B11		
CAD24	W4	A12	W3	A11		
CAD23	W3	C12	V2	E12		
CAD22	W2	A13	U2	B12		
CAD21	V3	B13	T4	A12		
CAD20	V2	A14	T1	A13		
CAD19	T4	C14	R2	E13		
CAD18	V1	A15	R1	F13		
CAD17	U2	B15	P6	A14		
CAD16	M4	C18	L2	D16		
CAD15	M2	C19	L6	D18		
CAD14	M3	B20	L1	C19		
CAD13	L4	E17	K2	E16		
CAD12	M1	D18	K4	D19		
CAD11	L3	C20	K1	E18		
CAD10	L2	D19	K6	E19		
CAD9	L1	E18	J5	F15		
CAD8	K3	E19	J1	F18		
CAD7	J1	G17	H4	G14		
CAD6	J4	G18	H6	G19		
CAD5	J3	F19	H1	G18		
CAD4	H2	G19	G1	H15		
CAD3	H1	F20	G4	G13		
CAD2	G1	H18	G6	H18		
CAD1	H3	G20	G5	H16		
CAD0	G2	H19	F2	H19		
CC/$\overline{\text{BE3}}$	Y2	B12	W2	D11	I/O	CardBus bus commands and byte enables. CC/$\overline{\text{BE3}}$–CC/$\overline{\text{BE0}}$ are multiplexed on the same CardBus terminals. During the address phase of a CardBus cycle, CC/$\overline{\text{BE3}}$–CC/$\overline{\text{BE0}}$ 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/$\overline{\text{BE0}}$ applies to byte 0 (CAD7–CAD0), CC/$\overline{\text{BE1}}$ applies to byte 1 (CAD15–CAD8), CC/$\overline{\text{BE2}}$ applies to byte 2 (CAD23–CAD8), and CC/$\overline{\text{BE3}}$ applies to byte 3 (CAD31–CAD24).
CC/$\overline{\text{BE2}}$	T3	D14	P4	D14		
CC/$\overline{\text{BE1}}$	N1	B19	L4	C18		
CC/$\overline{\text{BE0}}$	K1	D20	J2	F16		
CPAR	N3	A19	M1	B18	I/O	CardBus parity. In all CardBus read and write cycles, the PCI1450 calculates even parity across the CAD and CC/$\overline{\text{BE}}$ buses. As an initiator during CardBus cycles, the PCI1450 outputs CPAR with a one-CCLK delay. As a target during CardBus cycles, the calculated parity is compared to the initiator's parity indicator; a compare error results in a parity error assertion.

† Terminal name for slot A is preceded with A_. For example, the full name for terminal N3 is A_CPAR.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminal A19 is B_CPAR.

terminal functions (continued)

Table 16. CardBus PC Card Interface Control (slots A and B)

NAME	TERMINAL GFN NO. SLOT A†	TERMINAL GFN NO. SLOT B‡	TERMINAL GJG NO. SLOT A†	TERMINAL GJG NO. SLOT B‡	I/O TYPE	FUNCTION
CAUDIO	Y5	D10	V5	D10	I	CardBus audio. CAUDIO is a digital input signal from a PC Card to the system speaker. The PCI1450 supports the binary audio mode and outputs a binary signal from the card to SPKROUT.
$\overline{\text{CBLOCK}}$	P1	B18	M2	A18	I/O	CardBus lock. $\overline{\text{CBLOCK}}$ is used to gain exclusive access to a target.
$\overline{\text{CCD1}}$ $\overline{\text{CCD2}}$	G3 W6	H20 C9	F1 W6	H14 B9	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.
$\overline{\text{CDEVSEL}}$	R2	A18	N1	B16	I/O	CardBus device select. The PCI1450 asserts $\overline{\text{CDEVSEL}}$ to claim a CardBus cycle as the target device. As a CardBus initiator on the bus, the PCI1450 monitors $\overline{\text{CDEVSEL}}$ until a target responds. If no target responds before timeout occurs, then the PCI1450 terminates the cycle with an initiator abort.
$\overline{\text{CFRAME}}$	U1	C15	P2	E14	I/O	CardBus cycle frame. $\overline{\text{CFRAME}}$ is driven by the initiator of a CardBus bus cycle. $\overline{\text{CFRAME}}$ is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When $\overline{\text{CFRAME}}$ is deasserted, the CardBus bus transaction is in the final data phase.
$\overline{\text{CGNT}}$	P3	D16	N7	A16	I	CardBus bus grant. $\overline{\text{CGNT}}$ is driven by the PCI1450 to grant a CardBus PC Card access to the CardBus bus after the current data transaction has been completed.
$\overline{\text{CINT}}$	Y4	A10	T5	F10	I	CardBus interrupt. $\overline{\text{CINT}}$ is asserted low by a CardBus PC Card to request interrupt servicing from the host.
$\overline{\text{CIRDY}}$	T2	A16	P1	B15	I/O	CardBus initiator ready. $\overline{\text{CIRDY}}$ indicates the CardBus initiator's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of CCLK when both $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are asserted. Until $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are both sampled asserted, wait states are inserted.
$\overline{\text{CPERR}}$	P2	B17	M4	A17	I/O	CardBus parity error. $\overline{\text{CPERR}}$ is used to report 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.
$\overline{\text{CREQ}}$	Y1	D12	V1	D12	I	CardBus request. $\overline{\text{CREQ}}$ indicates to the arbiter that the CardBus PC Card desires use of the CardBus bus as an initiator.
$\overline{\text{CSERR}}$	V5	B10	R5	A10	I	CardBus system error. $\overline{\text{CSERR}}$ reports address parity errors and other system errors that could lead to catastrophic results. $\overline{\text{CSERR}}$ is driven by the card synchronous to CCLK, but deasserted by a weak pullup, and may take several CCLK periods. The PCI1450 can report $\overline{\text{CSERR}}$ to the system by assertion of $\overline{\text{SERR}}$ on the PCI interface.
$\overline{\text{CSTOP}}$	R1	C17	M5	B17	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	V6	A9	R6	E10	I	CardBus status change. CSTSCHG is used to alert the system to a change in the card's status and is used as a wake-up mechanism.
$\overline{\text{CTRDY}}$	P4	C16	N5	E15	I/O	CardBus target ready. $\overline{\text{CTRDY}}$ indicates the CardBus target's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of CCLK, when both $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are asserted; until this time, wait states are inserted.
CVS1 CVS2	Y3 U3	A11 B14	V4 R4	F11 B13	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 terminal Y5 is A_CAUDIO.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminal D10 is B_CAUDIO.

I/O characteristics

Figure 3 shows a 3-state bidirectional buffer illustration for reference. The table, *recommended operating conditions* provides the electrical characteristics of the inputs and outputs. The PCI1450 meets the ac specifications of the PC Card 95 Standard and the PCI Bus 2.1 specifications.

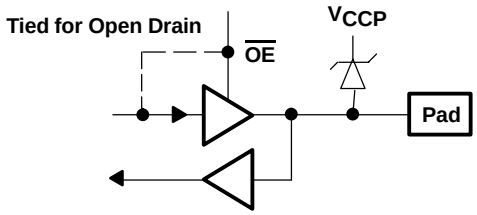


Figure 3. 3-State Bidirectional Buffer

clamping rail splits

The I/O sites can be pulled through a clamping diode to a power rail for protection. The core power supply is independent of the clamping rails. 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 PCI1450 must reliably accommodate both voltage levels. This is accomplished by using a 3.3-V buffer with tolerance (protection) at 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 rail for the sites as shown in Figure 3. After the terminal assignments are fixed, the fabrication facility will support a design by splitting the clamping rail for customization. The PCI1450 requires five separate clamping rails since it supports a wide range of features. The five rails are listed and defined in the table, *recommended operating conditions*.

PCI interface

This section describes the PCI interface of the PCI1450, and how the device responds and participates in PCI bus cycles. The PCI1450 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.

PCI bus lock ($\overline{\text{LOCK}}$)

The bus locking protocol defined in the PCI Specification is not highly recommended, but is provided on the PCI1450 as an additional compatibility feature. The PCI $\overline{\text{LOCK}}$ terminal is multiplexed with GPIO2, and the terminal function defaults to a general-purpose input (GPI). The use of $\overline{\text{LOCK}}$ is only supported by PCI-to-CardBus bridges in the downstream direction (away from the processor).

PCI $\overline{\text{LOCK}}$ indicates an atomic operation that may require multiple transactions to complete. When $\overline{\text{LOCK}}$ is asserted, nonexclusive transactions 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 $\overline{\text{LOCK}}$; control of $\overline{\text{LOCK}}$ is obtained under its own protocol. It is possible for different initiators to use the PCI bus while a single master retains ownership of $\overline{\text{LOCK}}$. To avoid confusion with the PCI bus clock, the CardBus signal for this protocol is $\overline{\text{CBLOCK}}$.

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

The PCI1450 supports all $\overline{\text{LOCK}}$ protocol associated with PCI-to-PCI bridges, as also defined for PCI-to-CardBus bridges. This includes disabling write posting while a locked operation is in progress, which can solve a potential deadlock when using devices such as PCI-to-PCI bridges. The potential deadlock can occur if a CardBus target supports delayed transactions, and blocks access 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 $\overline{\text{LOCK}}$.

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 PCI1450 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 PCI1450 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 PCI1450 core, including the serial EEPROM state machine. Refer to *suspend mode* for details on using $\overline{\text{SUSPEND}}$. The PCI1450 provides a two-line serial bus interface to the serial EEPROM.

The system designer must implement a pulldown resistor on the PCI1450 LATCH terminal to indicate the serial EEPROM mode. Only when this pulldown resistor is present will the PCI1450 attempt to load data through the serial EERPOM interface. The serial EEPROM interface is a two-pin interface with one data signal (SDA) and one clock signal (SCL). The SDA signal is mapped to the PCI1450 IRQMUX6 terminal and the SCL signal is mapped to the PCI1450 IRQMUX7 terminal. Figure 4 illustrates a typical PCI1450 application using the serial EEPROM interface.

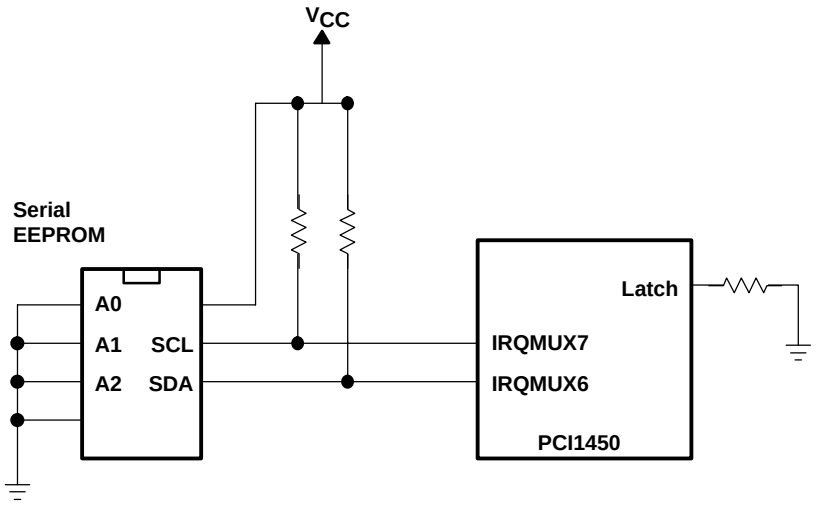


Figure 4. Serial EEPROM Application

As stated above, when the PCI1450 is reset by $\overline{\text{G\_RST}}$, the subsystem data is read automatically from the EEPROM. The PCI1450 masters the serial EEPROM bus and reads four bytes as described in Figure 5.

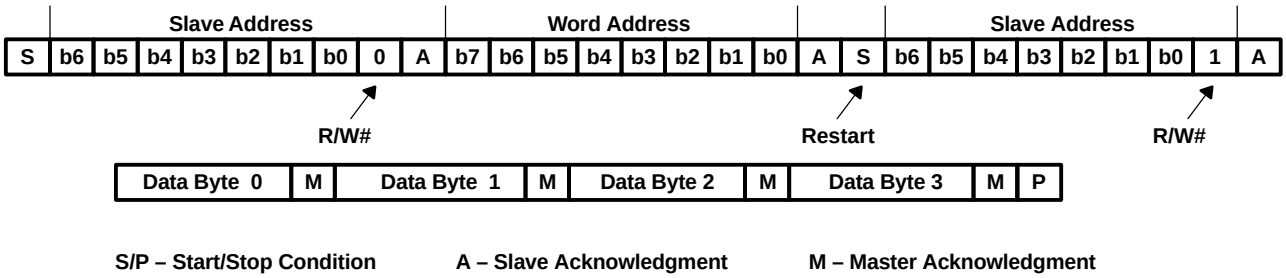


Figure 5. EEPROM Interface Subsystem Data Collection

The EEPROM is addressed at word address 00h, as indicated in Figure 5, and the address auto-increments after each byte transfers according to the protocol. 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 is addressed at slave address 1010000b by the PCI1450. All hardware address bits for the EEPROM should be tied to the appropriate level to achieve this address. The serial EEPROM chip in the sample application circuit, Figure 4, 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 3-stated and pulled high when the bus is not active. When the SDA line transitions to a logic low, this signals 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 6 illustrates this protocol.

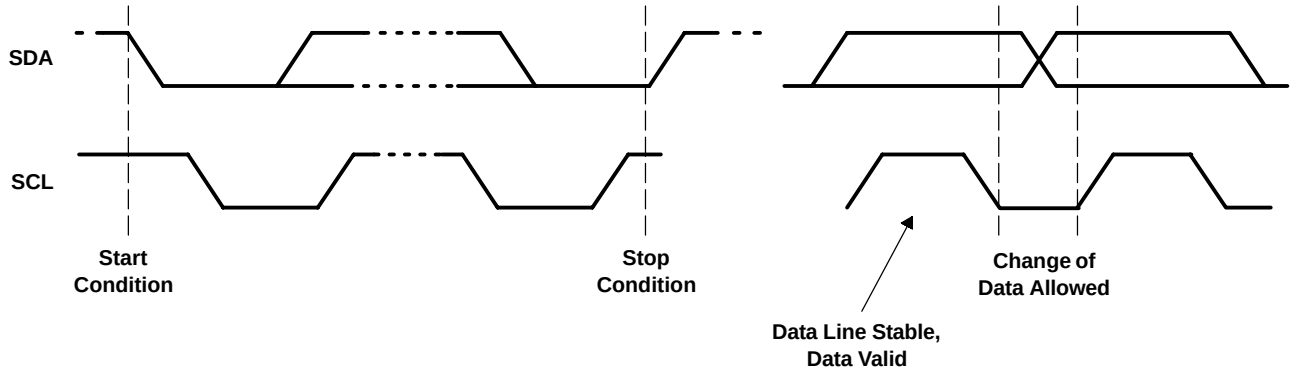


Figure 6. Serial EEPROM Start/Stop Conditions and BIt Transfers

Each address byte and data transfer is followed by an acknowledge bit, as indicated in Figure 5. When the PCI1450 transmits the addresses, it returns the SDA signal to the high state and 3-states the line. The PCI1450 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 PCI1450 transmitter and the EEPROM receiver. Figure 7 illustrates general acknowledges.

During the data byte transfers from the serial EEPROM to the PCI1450, the EEPROM clocks the SCL signal. After the EEPROM transmits the data to the PCI1450, it returns the SDA signal to the high state and 3-states the line. The EEPROM then generates an SCL clock cycle and expects the PCI1450 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 PCI1450 receiver. Figure 7 illustrates general acknowledges.

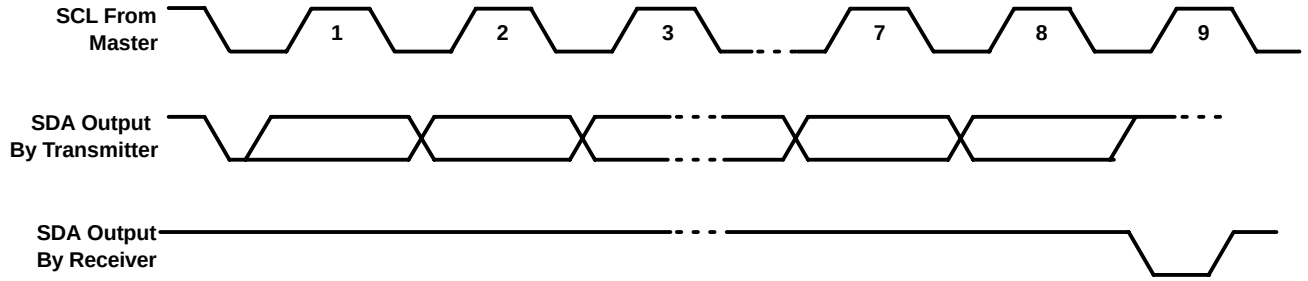


Figure 7. Serial EEPROM Protocol – Acknowledge

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 PCI1450 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).

PC Card applications overview

This section describes the PC Card interfaces of the PCI1450. A discussion is provided on PC Card recognition, which details the card interrogation procedure. The card powering procedure is discussed in this section including the protocol of the P2C power switch interface. The internal ZV buffering provided by the PCI1450 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.

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 17.

Table 17. 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		

P2C power switch interface (TPS2202A/2206)

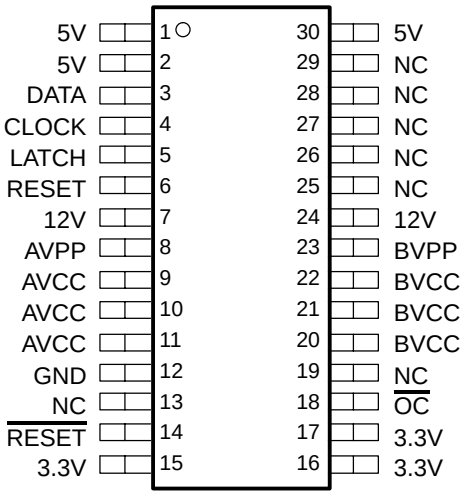
A power switch with a PCMCIA-to-peripheral control (P2C) interface is required for the PC Card powering interface. The TI TPS2206 (or TPS2202A) Dual-Slot PC Card Power-Interface Switch provides the P2C interface to the CLOCK, DATA, and LATCH terminals of the PCI1450. Figure 8 shows the terminal assignments of the TPS2206. Figure 9 illustrates a typical application where the PCI1450 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 PCI1450 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 8. TPS2206 Terminal Assignments

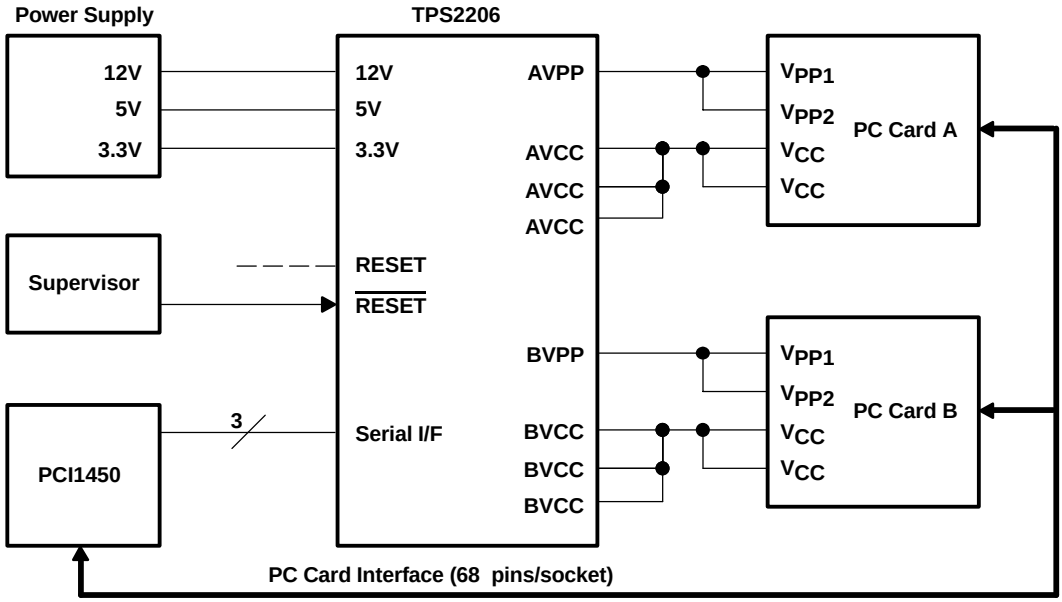
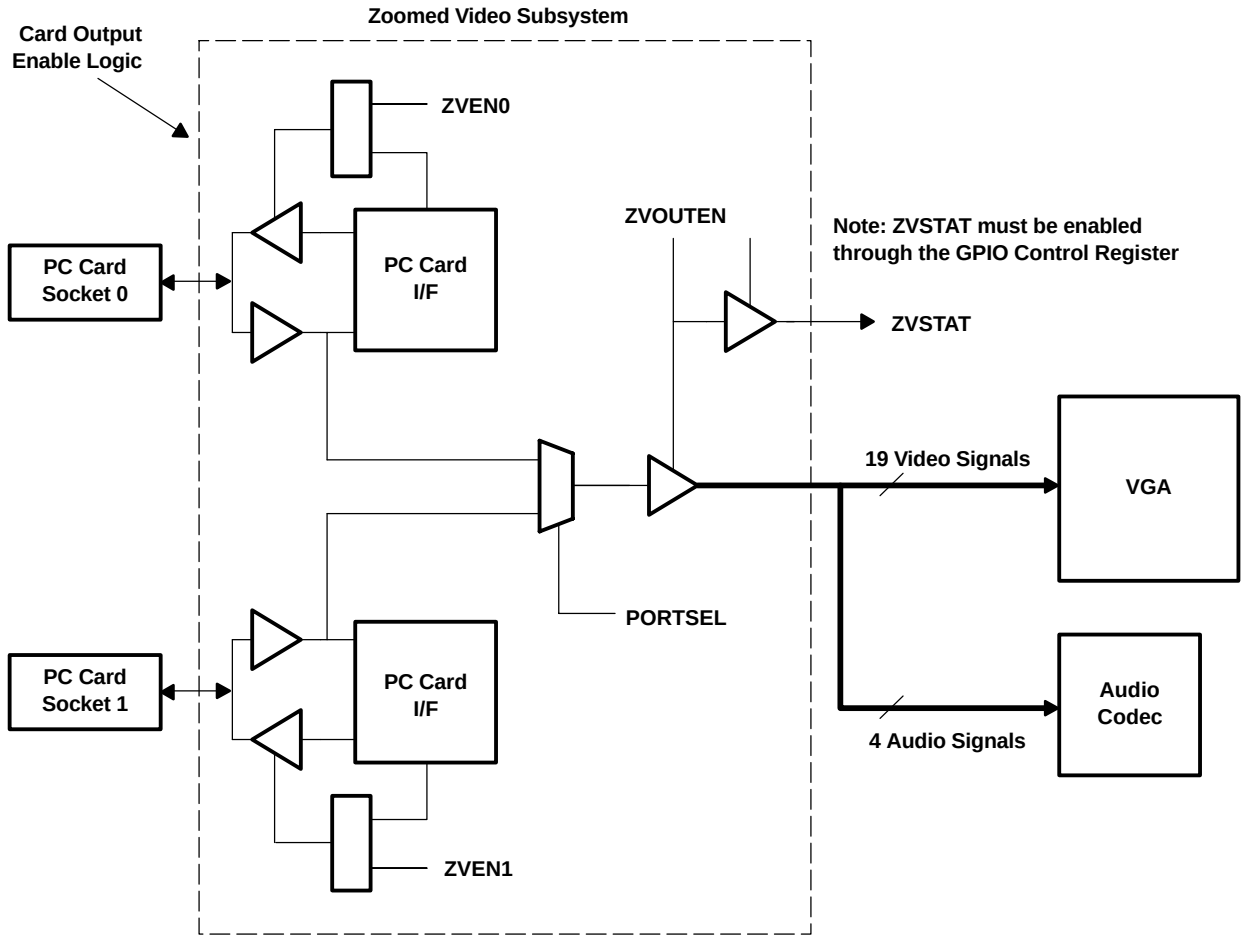


Figure 9. TPS2206 Typical Application

zoomed video support

The zoomed video (ZV) port on the PCI1450 provides an internally buffered 16-bit ZV PC Card data path. This internal routing is programmed through the *multimedia control register*. Figure 9 summarizes the zoomed video subsystem implemented in the PCI1450, and details the bit functions found in the *multimedia control register*.

An output port (PORTSEL) is always selected. The PCI1450 defaults to socket 0 (see the *multimedia control register*). When ZVOUTEN is enabled, the zoom video output terminals are enabled and allow the PCI1450 to route the zoom 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 zoom video port is driven low (i.e., no data is transmitted).



NOTES: A. ZVSTAT must be enabled through the GPIO control register.

Figure 10. Zoomed Video Subsystem

zoomed video auto detect

Zoomed video auto detect, when enabled, allows the PCI1450 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 PCI1450 automatically switches the internal zoomed video MUX to route the zoomed video stream to the PCI1450'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 PCI1450 can be programmed to switch a third zoomed video source by programming IRQMUX2 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 IRQMUX4 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.

The PCI1450 defaults with zoomed video auto-detect disabled so that it will function exactly like the PCI1250A. 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:

- *IRQMUX routing register* (PCI offset 8Ch), bits 11–8: Write 0010b to program IRQMUX2 as a pixel clock input pin.
- *IRQMUX routing register* (PCI ofset 8Ch), bits 19–16: Write 0001b to program IRQMUX4 as a ZVSTAT pin.

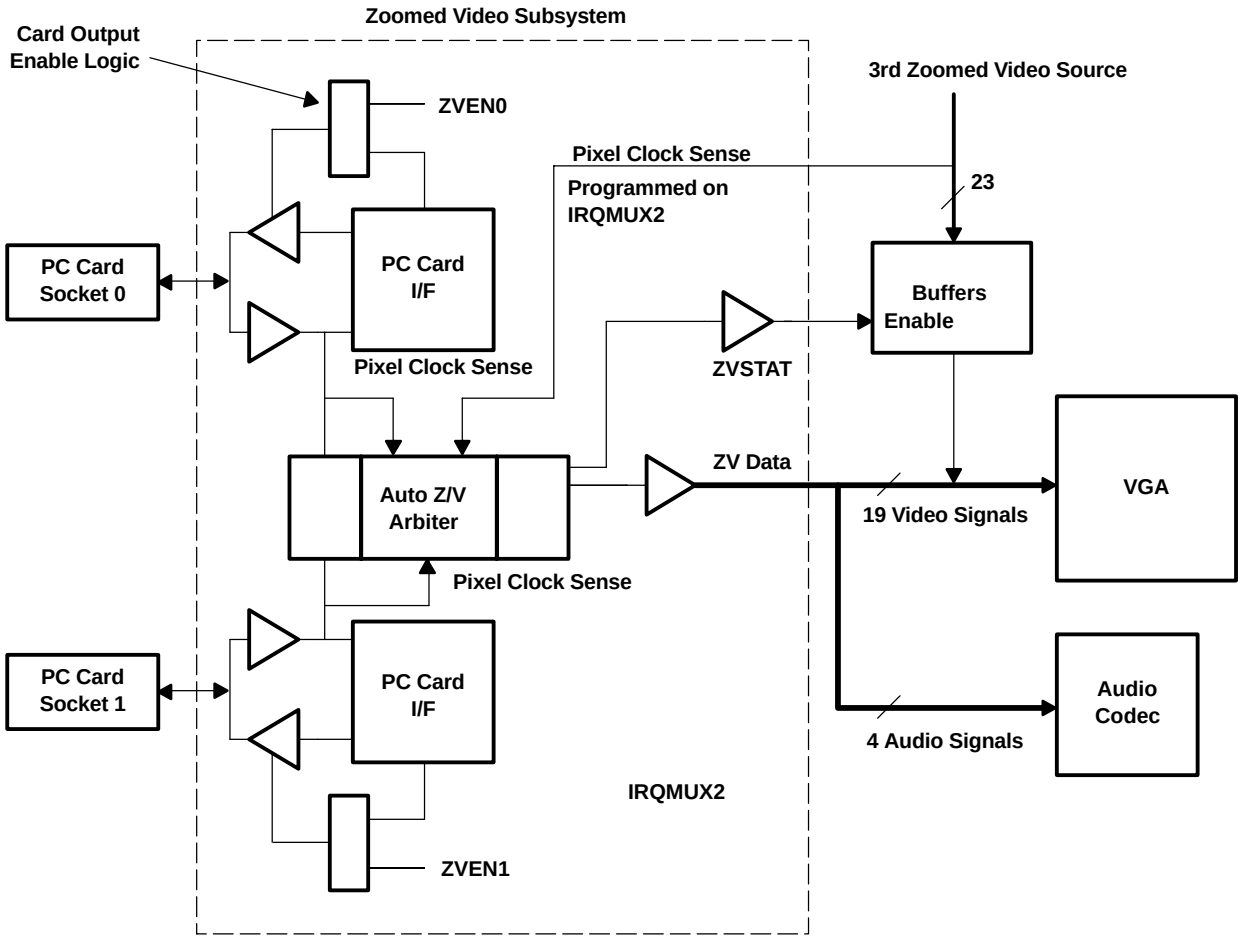


Figure 11. Zoomed Video with Auto Detect Enabled

ultra zoomed video

Ultra zoomed video is an enhancement to the PCI1450'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 1450 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 1450 prefetches an extra 16 bits (32 bits total) during each PCI read transaction. If the PCI Bus becomes busy, then the 1450 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.

$\overline{\text{D3\_STAT}}$ pin

Additional functionality added for the 1450 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

The intent of including this feature in the PCI1450 is to use this pin to switch an external V_{CC}/V_{AUX} switch. This feature can be programmed on GPIO1 pin (terminal W11) by writing 01b to bits 7–6 of the *GPIO1 control register* (PCI offset 89h).

internal ring oscillator

The internal ring oscillator provides an internal clock source for the PCI1450 so that neither the PCI clock nor an external clock is required in order for the PCI1450 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.

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 PCI1450. 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 PCI1450 to produce SPKROUT. Figure 12 illustrates the SPKROUT connection.

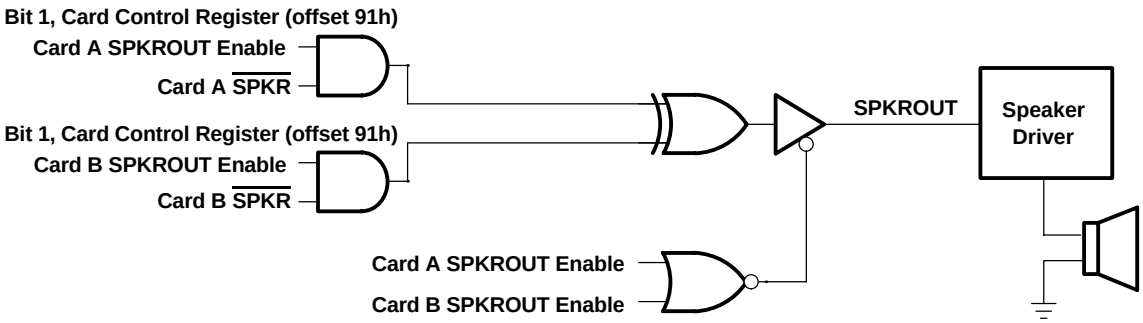


Figure 12. SPKROUT Connection to Speaker Driver

The SPKROUT signal is typically driven only by PC modem cards. To verify the SPKROUT on the PCI1450, 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 PCI1450 does not multiplex any other function on SPKROUT, this terminal does not require a pullup resistor.

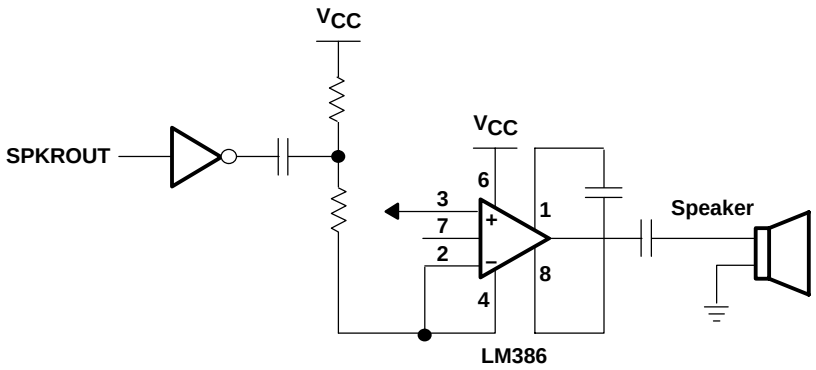


Figure 13. Simplified Test Schematic

LED socket activity indicators

The socket activity LEDs indicate when an access is occurring to a PC Card. The LED signals are multiplexed with general-purpose inputs and outputs (GPIOs); the default for these terminals is GPI. 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 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.

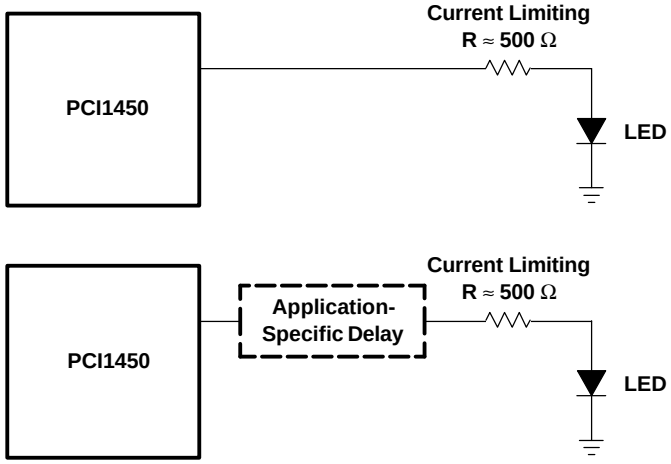


Figure 14. Two Sample LED Circuits

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.

PC Card 16 DMA support

The PCI1450 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 18 provides the DDMA register configuration.

Table 18. 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		

CardBus socket register

The PCI1450 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 19.

Table 19. 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

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 PCI1450. The PCI1450 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 PCI1450 is therefore backward compatible with existing interrupt control register definitions, and new registers have been defined where required.

The PCI1450 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 PCI1450, PC Card interrupts are classified as either card status change (CSC) or as functional interrupts.

The method by which any type of PCI1450 interrupt is communicated to the host interrupt controller varies from system to system. The PCI1450 offers system designers the choice of using parallel PCI interrupt signaling, parallel ISA type IRQ interrupt signaling, or the IRQSER serialized ISA and/or PCI interrupt protocol. Traditional ISA IRQ signaling is provided through eight IRQMUX terminals. It is possible to use the parallel PCI interrupts in combination with either parallel IRQs or serialized IRQs, as detailed in the sections that follow.

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 are defined as events at the PC Card interface which are detected by the PCI1450 and may warrant notification of host card and socket services software for service. CSC events include both card insertion and removal from PC Card sockets, as well as transitions of certain PC Card signals.

Table 20 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 20. 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 PCI1450 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 PCI1450 interrupt scheme may be used to notify the host system, as in indicated in Table 20, denoted by the power cycle complete event. This interrupt source is considered a PCI1450 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.

interrupt masks and flags

Host software may individually mask, or disable, most of the potential interrupt sources listed in Table 21 by setting the appropriate bits in the PCI1450. By individually masking the interrupt sources listed in these tables, software can control which events will cause a PCI1450 interrupt. Host software has some control over which system interrupt the PCI1450 will assert by programming the appropriate routing registers. The PCI1450 allows host software to route PC Card CSC and PC Card functional interrupts to separate system interrupts. A discussion of interrupt routing is somewhat specific to the interrupt signaling method used, and will be discussed in more detail in the next few sections.

When an interrupt is signaled by the PCI1450, the interrupt service routine must be able to discern which of the events in Table 21 caused the interrupt. Internal registers in the PCI1450 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 21 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 21. PCI1450 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

Notice that there is not a mask bit to stop the PCI1450 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 21. 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.

legacy interrupt multiplexer

The IRQ multiplexer implemented in the PCI1450 provides a mechanism to route the IRQMUX signals to any of the 15 legacy IRQ signals. The IRQMUX7–6 signals share the PC/PCI DMA terminals, and take precedence when routed. The other 6 IRQMUX signals (IRQMUX5–IRQMUX0) are available in all platforms. To use the IRQMUX interrupt signaling, software must program the *device control register*, located at PCI offset 92h, to select the legacy IRQ signaling scheme.

Figure 15 illustrates the IRQMUX functionality. This illustration describes only the PCREQ/IRQMUX7/SCL signal.

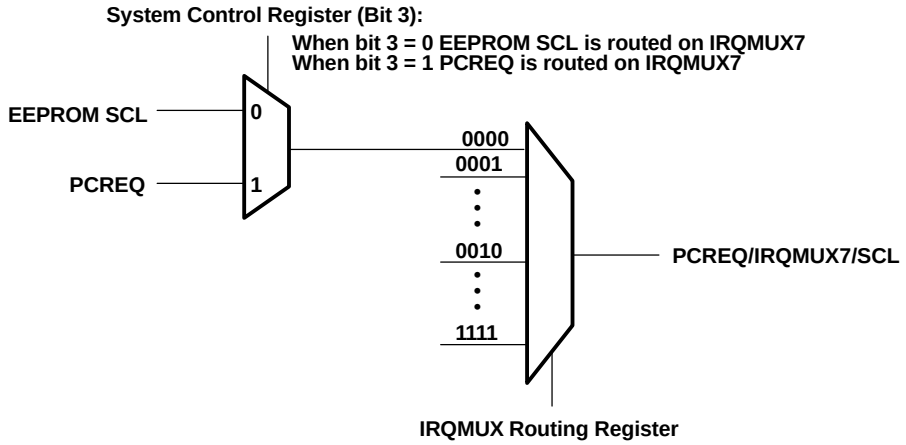


Figure 15. Interrupt Mux Functionality – Example of IRQMUX7 Routing

If parallel ISA IRQs are selected in the *device control register*, then the *IRQMUX routing register*, located at PCI offset 8Ch, must be programmed with the associated ISA IRQ connections. The PCI1450 supports up to eight parallel ISA IRQ signal connections, IRQMUX7–IRQMUX0. Figure 16 is an example PCI1450 IRQ implementation that provides eight ISA interrupts. The system in this example cannot support PC/PCI DMA since all eight ISA IRQs are used. In this example, the IRQMUX7 and IRQMUX6 terminals are used to signal ISA IRQs, and are not available for PC/PCI DMA. For systems not using all eight IRQs, PC/PCI DMA can be implemented and coexist with ISA IRQs by using IRQMUX6 and IRQMUX7 for PC/PCI DMA, i.e., legacy IRQs and PC/PCI DMA implementation are not mutually exclusive. However, if the IRQMUX registers are programmed to use IRQMUX7–IRQMUX6, then they will override PC/PCI DMA.

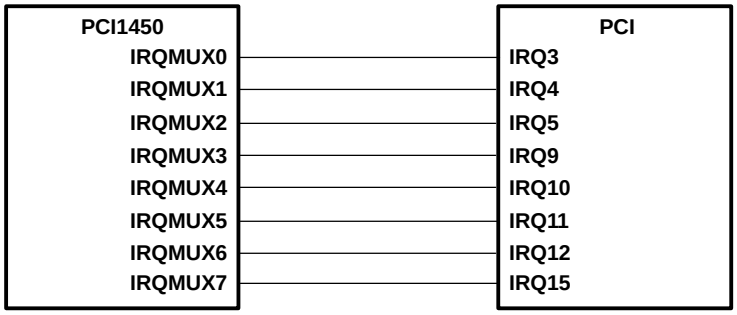


Figure 16. Example PCI1450 IRQ Implementation

Software is responsible for programming the *IRQMUX routing register* to reflect the IRQ configuration shown in Figure 16. In this example, this programming is accomplished by writing a double-word of data FCBA9543h to the PCI1450 *IRQMUX routing register*, PCI offset 8Ch. In this example (FCBA9543h), F corresponds to IRQ15, C to IRQ12, B to IRQ11, A to IRQ10, 9 to IRQ9, 5 to IRQ5, 4 to IRQ4, and 3 to IRQ3.

The *IRQMUX routing register* is shared between the two PCI1450 functions, and only one write to function 0 or function 1 is necessary to configure the IRQMUX signals.

using parallel PCI interrupts

Parallel PCI interrupts are available when in pure parallel PCI interrupt mode, IRQMUX signaling mode, and when only IRQs are serialized with the IRQSER protocol. The PCI interrupt signaling is dependent upon the interrupt mode and is summarized in Table 22. The interrupt mode is selected in the *device control register* (92h). The IRQSER/$\overline{\text{INTB}}$ signals INTB when one of the parallel interrupt modes is selected via bits 2–1 in the

device control register (92h). PCI $\overline{\text{INTB}}$ is also available on the IRQMUX0 terminal by programming bits 3–0 to 0001b in the *IRQMUX routing register* (8Ch). PCI $\overline{\text{INTA}}$ is available on the GPIO3 terminal by programming bits 7–6 in the *GPIO3 control register*.

Table 22. Interrupt Pin Register Cross Reference

Interrupt Signaling Mode	INTPIN Function 0	INTPIN Function 1
Parallel PCI Interrupts Only	0x01 ($\overline{\text{INTA}}$)	0x02 ($\overline{\text{INTB}}$)
Parallel IRQ & Parallel PCI Interrupts	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 ($\overline{\text{INTB}}$)

power management overview

In addition to the low-power CMOS technology process used for the PCI1450, various features are designed into the device to allow implementation of popular power saving techniques. These features and techniques are discussed in this section.

$\overline{\text{CLKRUN}}$ protocol

$\overline{\text{CLKRUN}}$ is the primary method of power management on the PCI bus side of the PCI1450. 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).

CardBus PC Card power management

The PCI1450 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.

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.

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 PCI1450 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 PCI1450, 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 PCI1450 V_{CC} pins. This switch should be a *make before break* type of switch, so that V_{CC} to the PCI1450 is not interrupted.

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.

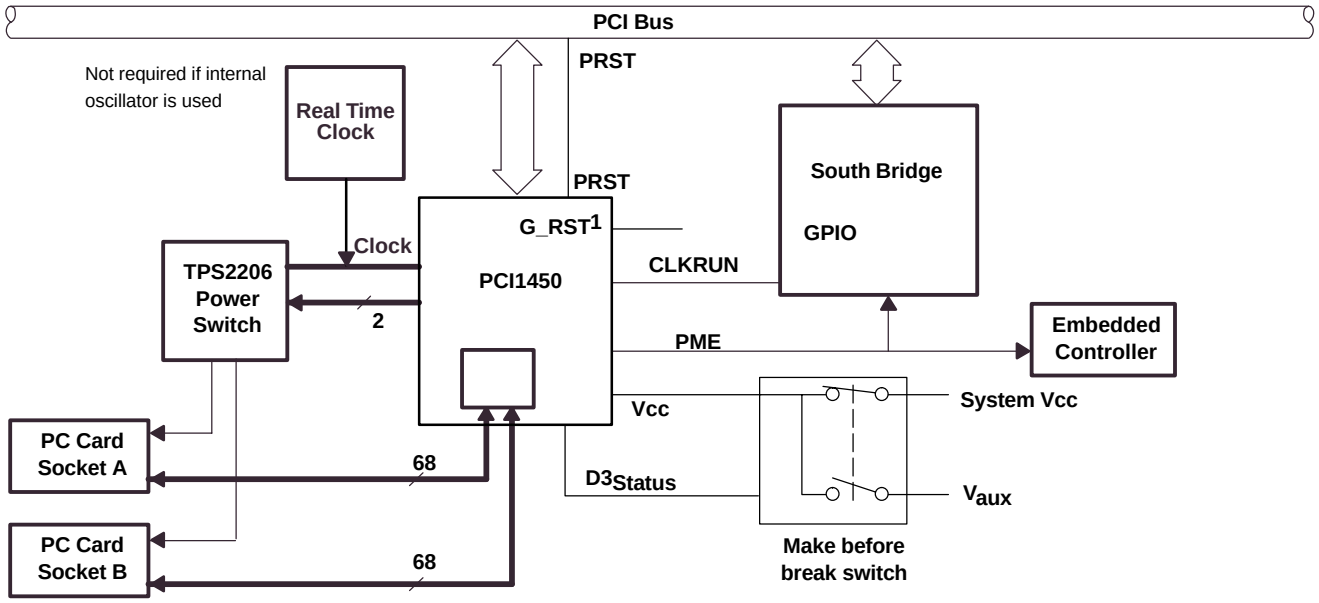
Global reset only bits:

- 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, 5, 4, 3, 1, 0
- Multimedia control register (PCI offset 84h): bits 7–0
- General status register (PCI offset 85h): bits 2–0
- GPIO0 control register (PCI offset 88h): bits 7, 6, 4, 3, 1, 0
- GPIO1 control register (PCI offset 89h): bits 7, 6, 3, 1, 0
- GPIO2 control register (PCI offset 8Ah): bits 7, 6, 4, 3, 1, 0
- GPIO3 control register (PCI offset 8Bh): bits 7, 6, 3, 1, 0
- IRQMUX 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 4, 3, 1, 0
- ExCA interrupt and general control (ExCA 803h/843h): bit 6
- ExCA card status change register (ExCA 804h/844h): bits 3, 2, 1, 0
- ExCA card status change interrupt register (ExCA 805h/845h): bits 3, 2, 1, 0
- CardBus socket event register (CardBus offset 00h): bits 3, 2, 1, 0
- CardBus socket mask register (CardBus offset 04h): bits 3, 2, 1, 0
- CardBus socket control register (CardBus offset 10h): bits 6, 5, 4, 2, 1, 0

system diagram implementing CardBus device class power management



NOTE: The system connection to $\overline{\text{G\_RST}}$ is implementation specific. $\overline{\text{G\_RST}}$ should be applied whenever V_{CC} is applied to the PCI1450. $\overline{\text{PRST}}$ should be applied for subsequent warm resets.

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 PCI1450. Besides gating $\overline{\text{PRST}}$ and $\overline{\text{G\_RST}}$, $\overline{\text{SUSPEND}}$ also gates PCLK inside the PCI1450 in order to minimize power consumption.

Gating PCLK does not create any issues with respect to the power switch interface in the PCI1450. This is because the PCI1450 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 PCI1450:

- Use an external clock to the PCI1450 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.

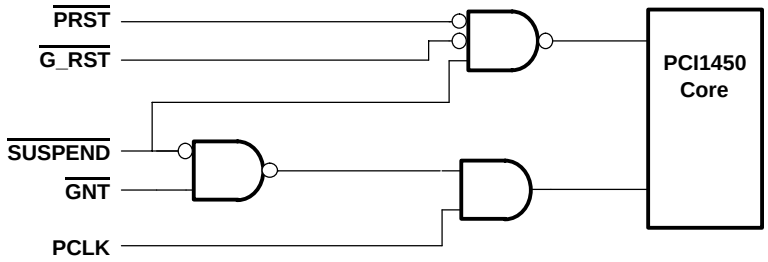


Figure 17. $\overline{\text{SUSPEND}}$ Functional Illustration

requirements for $\overline{\text{SUSPEND}}$

A requirement for implementing suspend mode is that the PCI bus must not be parked on the PCI1450 when $\overline{\text{SUSPEND}}$ is asserted. The PCI1450 responds to $\overline{\text{SUSPEND}}$ being asserted by 3-stating the $\overline{\text{REQ}}$ pin. The PCI1450 will also gate the internal clock and reset.

The GPIOs, IRQMUX signals, and $\overline{\text{RI\_OUT}}$ signals are all active during $\overline{\text{SUSPEND}}$, unless they are disabled in the appropriate PCI1450 registers.

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 PCI1450 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 PCI1450 $\overline{\text{RI\_OUT}}$ 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).

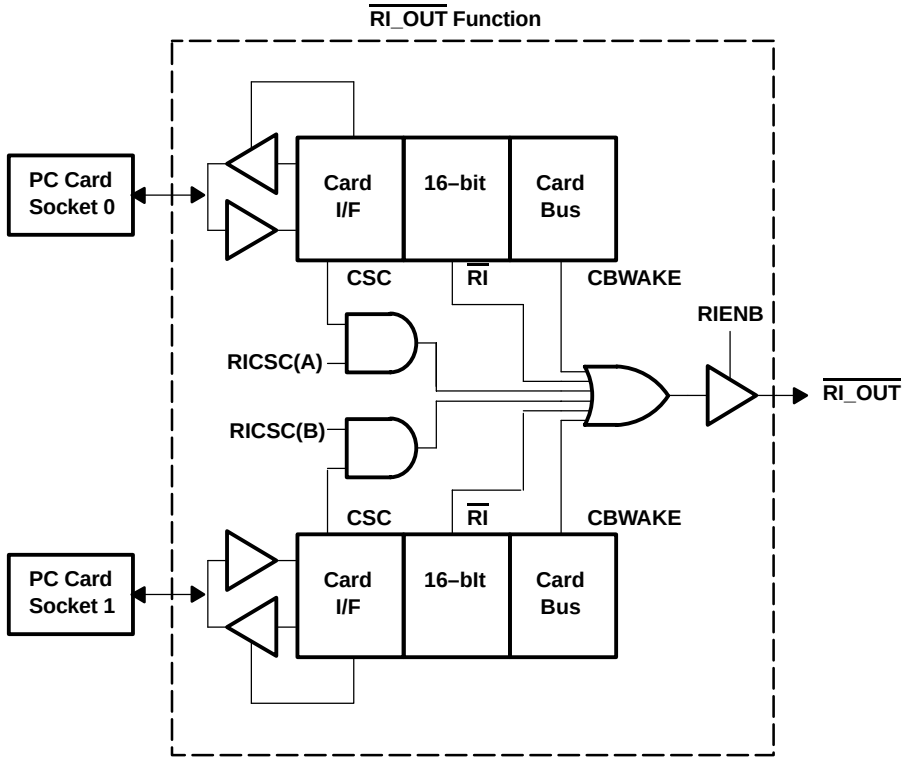


Figure 18. $\overline{\text{RI\_OUT}}$ Functional Illustration

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 18.

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.

PC CARD CONTROLLER PROGRAMMING MODEL

This section describes the PCI1450 PCI configuration registers that make up the 256-byte PCI configuration header for each PCI1450 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 $\overline{\text{G\_RST}}$. If $\overline{\text{PME}}$ is not enabled, then these bits are cleared by $\overline{\text{G\_RST}}$ 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 $\overline{\text{G\_RST}}$ signals should be tied together.

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

PCI configuration registers (functions 0 and 1)

The PCI1450 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 23 illustrates the PCI configuration header, which includes both the predefined portion of the configuration space and the user definable registers.

Table 23. 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

† One or more bits in the register are PME context bits and can only be cleared by the assertion of $\overline{\text{G\_RST}}$ 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 $\overline{\text{G\_RST}}$.
‡ One or more bits in this register are only cleared by the assertion $\overline{\text{G\_RST}}$.

Table 23. Functions 0 and 1 PCI Configuration Register Map (continued)

System control ‡				80h
Reserved	Reserved	General status †	Multimedia control ‡	84h
GPIO3 control ‡	GPIO2 control ‡	GPIO1 control ‡	GPIO0 control ‡	88h
IRQMUX 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 $\overline{\text{G\_RST}}$ 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 $\overline{\text{G\_RST}}$.
‡ One or more bits in this register are only cleared by the assertion $\overline{\text{G\_RST}}$.

vendor ID register

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
Description: This 16-bit 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.

device ID register

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	0	0	1	1	0	1	1

Register: **Device ID**
Type: Read-only
Offset: 02h (Functions 0, 1)
Default: AC1Bh
Description: This 16-bit register contains a value assigned to the PCI1450 by Texas Instruments. The device identification for the PCI1450 is AC1B.

command register

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
Description: The *command register* provides control over the PCI1450 interface to the PCI bus. All bit functions adhere to the definitions in the PCI Local Bus Specification, see Table 24. None of the bit functions in this register are shared between the two PCI1450 PCI functions. Two command registers exist in the PCI1450, one for each function. Software manipulates the two PCI1450 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.

Table 24. PCI Command Register Description

BIT	TYPE	FUNCTION
15–10	R	Reserved. These bits return 0s when read. Writes have no effect.
9	R	Fast back-to-back enable. The PCI1450 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 PCI1450 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 PCI1450 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 PCI1450'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 = PCI1450 ignores detected parity error (default). 1 = PCI1450 responds to detected parity errors.
5	R/W	VGA palette snoop. When set to 1, palette snooping is enabled (i.e., the PCI1450does not respond to palette register writes and snoops the data). When the bit is 0, the PCI1450 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 PCI1450 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 PCI1450 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 PCI1450 can act as a PCI bus initiator (master). The PCI1450 can take control of the PCI bus only when this bit is set. 0 = Disables the PCI1450's ability to generate PCI bus accesses (default). 1 = Enables the PCI1450's ability to generate PCI bus accesses.
1	R/W	Memory space enable. This bit controls whether or not the PCI1450 may claim cycles in PCI memory space. 0 = Disables the PCI1450's response to memory space accesses (default). 1 = Enables the PCI1450's response to memory space accesses.
0	R/W	I/O space control. This bit controls whether or not the PCI1450 may claim cycles in PCI I/O space. 0 = Disables the PCI1450 from responding to I/O space accesses (default). 1 = Enables the PCI1450 to respond to I/O space accesses.

status register

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
Description: 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.

Table 25. Status Register Description

BIT	TYPE	FUNCTION
15	R/W	PAR_ERR. 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	R/W	SYS_ERR. Signaled system error. This bit is set when $\overline{\text{SERR}}$ is enabled and the PCI1450 signaled a system error to the host. Write a 1 to clear this bit.
13	R/W	MABORT. Received master abort. This bit is set when a cycle initiated by the PCI1450 on the PCI bus has been terminated by a master abort. Write a 1 to clear this bit.
12	R/W	TABT_REC. Received target abort. This bit is set when a cycle initiated by the PCI1450 on the PCI bus was terminated by a target abort. Write a 1 to clear this bit.
11	R/W	TABT_SIG. Signaled target abort. This bit is set by the PCI1450 when it terminates a transaction on the PCI bus with a target abort. Write a 1 to clear this bit.
10–9	R	PCI_SPEED. DEVSEL timing. These bits encode the timing of $\overline{\text{DEVSEL}}$ and are hardwired 01b indicating that the PCI1450 asserts this signal at a medium speed on nonconfiguration cycle accesses.
8	R/W	DATAPAR. 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 PCI1450. b. The PCI1450 was the bus master during the data parity error. c. The parity error response bit is set in the *command register*.
7	R	FBB_CAP. Fast back-to-back capable. The PCI1450 cannot accept fast back-to-back transactions; thus, this bit is hardwired to 0.
6	R	UDF. UDF supported. The PCI1450 does not support the user definable features; therefore, this bit is hardwired to 0.
5	R	66 MHz capable. The PCI1450 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	R	Reserved. These bits return 0s when read.

revision ID register

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

Register: **Revision ID**
Type: Read-only
Offset: 08h (Functions 0, 1)
Default: 02h
Description: This register indicates the silicon revision of the PCI1450. This data sheet reflects the PCI1450 revision 02h silicon.

PCI class code register

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

Register: **PCI class code**
Type: Read-only
Offset: 09h (Functions 0, 1)
Default: 060700h
Description: This register recognizes the PCI1450 functions 0 and 1 as a bridge device (06h), and CardBus bridge device (07h), with a 00h programming interface.

cache line size register

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
Description: This register is programmed by host software to indicate the system cache line size.

latency timer register

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
Description: This register specifies the latency timer for the PCI1450, in units of PCI clock cycles. When the PCI1450 is a PCI bus initiator and asserts $\overline{\text{FRAME}}$, the latency timer begins counting from zero. If the latency timer expires before the PCI1450 transaction has terminated, then the PCI1450 terminates the transaction when its $\overline{\text{GNT}}$ is deasserted.

header type register

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
Description: This register returns 82h when read, indicating that the PCI1450 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.

BIST register

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
Description: Since the PCI1450 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 PCI1450 functions.

CardBus socket registers / ExCA registers base address register

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
Description: 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 4K-byte 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.

capability pointer

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
Description: This 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.

secondary status register

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
Description: This 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.

Table 26. Secondary Status Register Description

BIT	TYPE	FUNCTION
15	R/WC	CBPARITY. 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	R/WC	CBSERR. Signaled system error. This bit is set when $\overline{\text{CSERR}}$ is signaled by a CardBus card. The PCI1450 does not assert the $\overline{\text{CSERR}}$ signal. Write a 1 to clear this bit.
13	R/WC	CBMABORT. Received master abort. This bit is set when a cycle initiated by the PCI1450 on the CardBus bus has been terminated by a master abort. Write a 1 to clear this bit.
12	R/WC	REC_CBTA. Received target abort. This bit is set when a cycle initiated by the PCI1450 on the CardBus bus was terminated by a target abort. Write a 1 to clear this bit.
11	R/WC	SIG_CBTA. Signaled target abort. This bit is set by the PCI1450 when it terminates a transaction on the CardBus bus with a target abort. Write a 1 to clear this bit.
10–9	R	CB_SPEED. CDEVSEL timing. These bits encode the timing of $\overline{\text{CDEVSEL}}$ and are hardwired 01b indicating that the PCI1450 asserts this signal at a medium speed.
8	R/WC	CB_DPAR. 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 PCI1450 was the bus master during the data parity error. c. The parity error response bit is set in the *bridge control register*.
7	R	CBFBB_CAP. Fast back-to-back capable. The PCI1450 cannot accept fast back-to-back transactions; therefore, this bit is hardwired to 0.
6	R	CB_UDF. User definable feature support. The PCI1450 does not support the user definable features; therefore, this bit is hardwired to 0.
5	R	CB66MHZ. 66 MHz capable. The PCI1450 CardBus interface operates at a maximum CCLK frequency of 33 MHz; therefore, this bit is hardwired to 0.
4–0	R	Reserved. These bits return 0s when read.

PCI bus number register

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
Description: This register is programmed by the host system to indicate the bus number of the PCI bus to which the PCI1450 is connected. The PCI1450 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.

CardBus bus number register

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
Description: This register is programmed by the host system to indicate the bus number of the CardBus bus to which the PCI1450 is connected. The PCI1450 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 PCI1450 controller function.

subordinate bus number register

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
Description: This register is programmed by the host system to indicate the highest numbered bus below the CardBus bus. The PCI1450 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.

CardBus latency timer register

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
Description: This register is programmed by the host system to specify the latency timer for the PCI1450 CardBus interface, in units of CCLK cycles. When the PCI1450 is a CardBus initiator and asserts $\overline{\text{CFRAME}}$, the CardBus latency timer begins counting. If the latency timer expires before the PCI1450 transaction has terminated, then the PCI1450 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.

memory base registers 0, 1

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
Description: These registers indicate the lower address of a PCI memory address range. They are used by the PCI1450 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 4K-byte 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 PCI1450 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).

memory limit registers 0, 1

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
Description: These registers indicate the upper address of a PCI memory address range. They are used by the PCI1450 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 4K-byte 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 PCI1450 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).

I/O base registers 0, 1

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
Description: These registers indicate the lower address of a PCI I/O address range. They are used by the PCI1450 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 64K-byte page, and the upper 16 bits (31–16) are all 0s which locate this 64K-byte 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 64K-byte 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.

I/O limit registers 0, 1

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
Description: These registers indicate the upper address of a PCI I/O address range. They are used by the PCI1450 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 64K-byte page, and the upper 16 bits are a page register which locates this 64K-byte 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 64K-byte 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 PCI1450 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.

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
Description: This register communicates interrupt line routing information to the host system. This register is not used by the PCI1450, since there are many programmable interrupt signaling options. This register is considered reserved; however, host software may read and write to this register. Each PCI1450 function has an *interrupt line register*.

interrupt pin register

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.
Description: The value read from this register is function dependent. The value depends on the interrupt INTRTIE bit in the *system control register* and the signaling mode, selected through the *device control register*. When the INTRTIE bit is set, this register will read 0x01 ($\overline{\text{INTA}}$) for both functions. The PCI1450 defaults to signaling PCI & IRQ interrupts through the IRQSER serial interrupt terminal. Refer to Table 27 for a complete description of the register contents.

Table 27. Interrupt Pin Register Cross Reference

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

bridge control register

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
Description: This register provides control over various PCI1450 bridging functions. Some bits in this register are global in nature and should be accessed only through function 0.

Table 28. Bridge Control Register Description

BIT	TYPE	FUNCTION
15–11	R	Reserved. These bits return 0s when read.
10	R/W	POSTEN. 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	R/W	PREFETCH1. 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	R/W	PREFETCH0. 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†	R/W	CRST. 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 $\overline{\text{G\_RST}}$.
5§	R/W	MABTMODE. Master abort mode. This bit controls how the PCI1450 responds to a master abort when the PCI1450 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	R	Reserved. This bit returns 0 when read.
3	R/W	VGAEN. VGA enable. This bit affects how the PCI1450 responds to VGA addresses. When this bit is set, accesses to VGA addresses will be forwarded.
2	R/W	ISAEN. ISA mode enable. This bit affects how the PCI1450 passes I/O cycles within the 64K-byte ISA range. This bit is not common between sockets. When this bit is set, the PCI1450 will not forward the last 768 bytes of each 1K I/O range to CardBus.
1	R/W	CSERREN. $\overline{\text{CSERR}}$ enable. This bit controls the response of the PCI1450 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	R/W	CPERREN. CardBus parity error response enable. This bit controls the response of the PCI1450 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 cleared only by the assertion of $\overline{\text{G\_RST}}$ 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 $\overline{\text{G\_RST}}$.

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

subsystem vendor ID register

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

‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

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
Description: This 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.

subsystem ID register

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

‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

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
Description: This 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.

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

PC Card 16-bit I/F legacy mode base address register

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

‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

Register: **PC Card 16-bit I/F legacy mode base address**
Type: Read-only, Read/Write
Offset: 44h (Functions 0, 1)
Default: 0000 0001h
Description: The PCI1450 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.

system control register

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

‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

Register: **System control**
Type: Read-only, Read/Write
Offset: 80h (Functions 0, 1)
Default: 0044 9060h
Description: 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.

Table 29. System Control Register Description

BIT	TYPE	FUNCTION
31–30‡§	R/W	SER_STEP. 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 PCI1450 functions. 00 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ signal in $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ IRQSER slots (default) 01 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ signal in $\overline{\text{INTB}}$/$\overline{\text{INTC}}$ IRQSER slots 10 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ signal in $\overline{\text{INTC}}$/$\overline{\text{INTD}}$ IRQSER slots 11 = INTA/INTB signal in INTD/INTA IRQSER slots
29‡§	R/W	INTRTIE. 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 INTA. $\overline{\text{INTA}}$ may then be shifted by using the SER_STEP bits. This bit is global to both PCI1450 functions. 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	R	Reserved. This bit returns 0 when read.
27‡§	R/W	P2CCLK. 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 internal oscillator is enabled. A 43kΩ pulldown resistor should be tied to this terminal.
26‡§	R/W	SMIROUTE. 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‡	R/W	SMISTATUS. 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‡§	R/W	SMIENB. 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	R	Reserved
22‡	R/W	CBRSVD. CardBus reserved terminals signaling. When this bit is set, the RSVD CardBus terminals will be driven low when a CardBus card is inserted. When this bit is low, as default, these signals are 3-stated. 0 = 3-state the CardBus RSVD terminals 1 = Drive the Cardbus RSVD terminals low (default).
21‡	R/W	VCCPROT. 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 zoom 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 zoom video is disabled (default). 1 = Reduced zoom video is enabled.
19‡	R/W	CDREQEN. PC/PCI DMA card enable. When this bit is set, the PCI1450 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‡	R/W	CDMACHAN. 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

§ These bits are global in nature and should be accessed only through function 0.
‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

Table 29. System Control Register Description (continued)

BIT	TYPE	FUNCTION
15‡§	R/W	MRBURSTDN. 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‡§	R/W	MRBURSTUP. Memory read burst enable upstream. When this bit is set, the PCI1450 allows memory read transactions to burst upstream. 0 = MRBURSTUP upstream is disabled (default). 1 = MRBURSTUP upstream is enabled.
13	R	SOCACTIVE. 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	R	Reserved. This bit returns 1 when read. This is the power rail bit in functions 0 and 1.
11	R	PWRSTREAM. 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	R	DELAYUP. 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	R	DELAYDOWN. 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	R	INTERROGATE. 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	R	Reserved. This bit returns 0 when read.
6‡§	R/W	PWRSAVINGS. Power savings mode enable. When this bit is set, the PCI1450 will consume less power with no performance loss. This bit is shared between the two PCI1450 functions. 0 = Power savings mode disabled 1 = Power savings mode enabled (default)
5‡§	R/W	SUBSYSRW. *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. 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‡§	R/W	CB_DPAR. 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‡§	R/W	CDMA_EN. PC/PCI DMA enable. Enables PC/PCI DMA when set, and disables the IRQMUX7 and IRQMUX6 signaling. 0 = Centralized DMA disabled (default) 1 = Centralized DMA enabled
2	R	Reserved. This bit returns 0 when read.

§ These bits are global in nature and should be accessed only through function 0.
‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

Table 29. System Control Register Description (continued)

BIT	TYPE	FUNCTION
1‡§	R/W	KEEPCLK. Keep clock. When this bit is set, the PCI1450 will always follow $\overline{\text{CLKRUN}}$ protocol to maintain the system PCLK and the CCLK (CardBus clock). This bit is global to the PCI1450 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 PCI1450, 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. If this bit is 0 and bit 7 (RIENB) of the *card control register* is 0, then the output (Y13) will be 3-stated. This pin is encoded as: 0 = $\overline{\text{RI\_OUT}}$ signal is routed to pin Y13 if bit 7 of the *card control register* is 1*. (default) 1 = $\overline{\text{PME}}$ signal is routed on pin Y13 of the PCI1450 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 is 3-stated.

§ These bits are global in nature and should be accessed only through function 0.
‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

multimedia control register

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
Description: This register provides port mapping for the PCI1450 zoom video/data ports. See *zoomed video support* for details on the PCI1450 zoom video support. Access this register only through function 0.

Table 30. Multimedia Control Register Description

BIT	TYPE	FUNCTION
7‡	R/W	ZVOUTEN. ZV output enable. This bit enables the output for the PCI1450 outsourcing ZV terminals. When this bit is reset, ‘0’, these terminals are in a high impedance state. 0 = PCI1450 ZV output terminals disabled (default) 1 = PCI1450 ZV output terminals enabled
6‡	R/W	PORTSEL. ZV port select. This bit controls the multiplexing control over which PC Card ZV port data will be driven to the outsourcing PCI1450 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‡	R/W	ZVEN1. PC Card 1 ZV mode enable. Enables the zoom video mode for socket 1. When set, the PCI1450 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‡	R/W	ZVEN0. PC Card 0 ZV mode enable. Enables the zoom video mode for socket 0. When set, the PCI1450 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

‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

general status register

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

Register: **General status**
Type: Read-only
Offset: 85h (Functions 0)
Default: 00h
Description: This register provides the general device status information. The status of the serial EEPROM interface is provided through this register.

Table 31. General Status Register Description

BIT	TYPE	FUNCTION
7–3	R	Reserved. These bits return 0s when read.
2‡§	R	EEDETECT. Serial EEPROM detect. When this bit is cleared, it indicates that the PCI1450 serial EEPROM circuitry has detected an EEPROM. A pull-up resistor must be implemented on the LATCH terminal for this bit to be set. This status bit is encoded as: 0 = EEPROM not detected (default) 1 = EEPROM detected
1‡§	R	DATAERR. 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‡§	R	EEBUSY. Serial EEPROM busy status. This bit indicates the status of the PCI1450 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.
‡ This bit is cleared only by the assertion of G_RST.

GPIO0 control register

Bit	7‡	6‡	5	4‡	3‡	2	1‡	0‡
Name	GPIO0 control							
Type	R/W	R/W	R	R/W	R/WC	R	R/W	R/W
Default	1	0	0	0	0	0	0	0

Register: **GPIO0 control**
Type: Read-only, Read/Write, Read/Write to Clear
Offset: 88h (Functions 0, 1)
Default: 80h
Description: This register is used for control of the general-purpose I/O, GPIO0. This terminal defaults to a general-purpose input, but can be reconfigured as the socket 0 activity LED output, a zoom video enabled status output, or general-purpose output. Access this register only through function 0.

Table 32. GPIO0 Control Register Description

BIT	TYPE	FUNCTION
7–6‡	R/W	GP0. General-purpose 0 mode. These bits select the functionality of the LEDA1/GPIO0 signal. These bits are encoded as: 00 = Signal LEDA1 to indicate PC Card socket 0 activity 01 = Signal ZVSTAT to indicate zoom video output enabled 10 = General-purpose input (GPI) 11 = General-purpose output (GPO)
5	R	Reserved. This bit returns 0 when read. A write has no effect.
4‡	R/W	GPINTEN0. GP interrupt enable. When this bit is set, a socket A card status change (CSC) interrupt is generated when the DELTA0 bit is set.
3‡	R/WC	DELTA0. DATAIN0 change status. This bit is set when the DATAIN0 bit changes state when in GPI mode. Glitches on the GPI terminal may not be detected by software without this bit. This bit is cleared by a write back of 1.
2	R	Reserved. This bit returns 0 when read. A write has no effect.
1‡	R/W	DATAOUT0. General-purpose data output. When in general-purpose output mode, this bit represents the data. Data written to this bit in GPO mode is signaled to the output.
0‡	R/W	DATAIN0. General-purpose data input. When in either general-purpose input or output mode, this bit represents the data on the GPIO terminal. Data signaled on the GPI terminal is identified through this bit.

‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

GPIO1 control register

Bit	7‡	6‡	5	4	3‡	2	1‡	0‡
Name	GPIO1 control							
Type	R/W	R/W	R	R	R/WC	R	R/W	R/W
Default	1	0	0	0	0	0	0	0

Register: **GPIO1 control**
Type: Read-only, Read/Write, Read/Write to Clear
Offset: 89h (Functions 0, 1)
Default: 80h
Description: This register is used for control of the general-purpose I/O, GPIO1. This terminal defaults to a general-purpose input, but can be reconfigured as the socket 1 activity LED output, or general-purpose output. Access this register only through function 0.

Table 33. GPIO1 Control Register Description

BIT	TYPE	FUNCTION
7–6‡	R/W	GP1. General-purpose 1 mode. These bits select the functionality of the LEDA2/GPIO1 signal. These bits are encoded as: 00 = Signal LEDA2 to indicate PC Card socket 1 activity 01 = D3_STAT. This programs GPIO1 as a D3 status pin. D3 status will be asserted if both function 0 and function 1 are placed in the D3 state and PME is enabled via bit 8 of PCI offset A4h. 10 = General-purpose input (GPI) 11 = General-purpose output (GPO)
5	R	Reserved. This bit returns 0 when read. A write has no effect.
4	R	Reserved. This bit returns 0 when read. A write has no effect.
3‡	R/WC	DELTA1. DATAIN1 change status. This bit is set when the DATAIN1 bit changes state when in GPI mode. Glitches on the GPI terminal may not be detected by software without this bit. This bit is cleared by a write back of 1.
2	R	Reserved. This bit returns 0 when read. A write has no effect.
1‡	R/W	DATAOUT1. General-purpose data output. When in general-purpose output mode, this bit represents the data. Data written to this bit in GPO mode is signaled to the output.
0‡	R/W	DATAIN1. General-purpose data input. When in either general-purpose input or output mode, this bit represents the data on the GPIO terminal. Data signaled on the GPI terminal is identified through this bit.

‡ This bit is cleared only by the assertion of G_RST.

GPIO2 control register

Bit	7‡	6‡	5	4‡	3‡	2	1‡	0‡
Name	GPIO2 control							
Type	R/W	R/W	R	R/W	R/WC	R	R/W	R/W
Default	1	0	0	0	0	0	0	0

Register: **GPIO2 control**
Type: Read-only, Read/Write, Read/Write to Clear
Offset: 8Ah (Functions 0, 1)
Default: 80h
Description: This register is used for control of the general-purpose I/O, GPIO2. This terminal defaults to a general-purpose input, but can be reconfigured as the PCI $\overline{\text{LOCK}}$ signal, a zoom video enabled status output, or general-purpose output. Access this register only through function 0.

Table 34. GPIO2 Control Register Description

BIT	TYPE	FUNCTION
7–6‡	R/W	GP2. General-purpose 2 mode. These bits select the functionality of the $\overline{\text{LOCK}}$/GPIO2 signal. These bits are encoded as: 00 = Terminal is configured as the PCI $\overline{\text{LOCK}}$ signal. 01 = Signal ZVSTAT to indicate zoom video output is enabled. 10 = General-purpose input (GPI) 11 = General-purpose output (GPO)
5	R	Reserved. This bit returns 0 when read. A write has no effect.
4‡	R/W	GPINTEN2. GP interrupt enable. When this bit and the DELTA2 bit are set, a socket B CSC is generated.
3‡	R/WC	DELTA2. DATAIN2 change status. This bit is set when the DATAIN2 bit changes state when in GPI mode. Glitches on the GPI terminal may not be detected by software without this bit. This bit is cleared by a write back of 1.
2	R	Reserved. This bit returns 0 when read. A write has no effect.
1‡	R/W	DATAOUT2. General-purpose data output. When in general-purpose output mode, this bit represents the data. Data written to this bit in GPO mode is signaled to the output.
0‡	R/W	DATAIN2. General-purpose data input. When in either general-purpose input or output mode, this bit represents the data on the GPIO terminal. Data signaled on the GPI terminal is identified through this bit.

‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

GPIO3 control register

Bit	7‡	6‡	5	4	3‡	2	1‡	0‡
Name	GPIO3 control							
Type	R/W	R/W	R	R	R/WC	R	R/W	R/W
Default	1	0	0	0	0	0	0	0

Register: **GPIO3 control**
Type: Read-only, Read/Write, Read/Write to Clear
Offset: 8Bh (Functions 0, 1)
Default: 80h
Description: This register is used for control of the general-purpose I/O, GPIO3. This terminal defaults to a general-purpose input, but can be reconfigured as the PCI $\overline{\text{INTA}}$ signal, or general-purpose output. Access this register only through function 0.

Table 35. GPIO3 Control Register Description

BIT	TYPE	FUNCTION
7–6‡	R/W	GP3. General-purpose 3 mode. These bits select the functionality of the $\overline{\text{INTA}}$/GPIO3 signal. These bits are encoded as: 00 = Terminal is configured as the PCI $\overline{\text{INTA}}$ signal. 01 = Reserved. 10 = General-purpose input (GPI) 11 = General-purpose output (GPO)
5	R	Reserved. This bit returns 0 when read. A write has no effect.
4	R	Reserved. This bit returns 0 when read. A write has no effect.
3‡	R/WC	DELTA3. DATAIN3 change status. This bit is set when the DATAIN3 bit changes state when in GPI mode. Glitches on the GPI terminal may not be detected by software without this bit. This bit is cleared by a write back of 1.
2	R	Reserved. This bit returns 0 when read. A write has no effect.
1‡	R/W	DATAOUT3. General-purpose data output. When in general-purpose output mode, this bit represents the data. Data written to this bit in GPO mode is signaled to the output.
0‡	R/W	DATAIN3. General-purpose data input. When in either general-purpose input or output mode, this bit represents the data on the GPIO terminal. Data signaled on the GPI terminal is identified through this bit.

‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

IRQMUX routing register

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

Register: **IRQMUX routing**
Type: Read/Write
Offset: 8Ch (Functions 0, 1)
Default: 0000 0000h
Description: This register is used for the legacy interrupt mux routing feature of the PCI1450, which is described in the *programmable interrupt support* section. If the parallel IRQ interrupt scheme is selected, then all PCI1450 interrupts sent to ISA IRQs will be signaled on the corresponding IRQMUX7–IRQMUX0 signals. These signals are routed directly to various IRQ inputs on the system PIC. The routing information is programmed through this register. Each terminal has at least one secondary function that can be selected by programming the bits appropriately. Access this register only through function 0.

Table 36. IRQMUX Routing Register Description

BIT	TYPE	FUNCTION
31–28‡	R/W	IRQMUX7 routing. These bits select 1 of 15 interrupts that can be routed on the IRQMUX7 pin. When these bits are 0000 and bit 3 in the *system control register* is set, this pin is used for PCREQ DMA signaling. NOTE: These bits should not be configured for IRQ signaling if IRQMUX7 is being used for PCREQ signaling. 0000 = EEPROM SCL routed on IRQMUX7 pin (default) 0000 = PCREQ routed on IRQMUX 7 pin when bit 3 of the *system control register* is 1 0001 = PC/PCI DMA request (PCREQ) routed on IRQMUX7 pin 0010 = IRQ2 routed on IRQMUX7 pin 0011 = IRQ3 routed on IRQMUX7 pin : 1111 = IRQ15 routed on IRQMUX7 pin
27–24‡	R/W	IRQMUX6 routing. These bits select 1 of 15 interrupts that can be routed on the IRQMUX6 pin. When these bits are 0000 and bit 3 is the *system control register* is set, this pin is used for PCGNT DMA signaling. NOTE: These bits should not be configured for IRQ signaling if IRQMUX6 is being used for PCGNT signaling. An EEPROM cannot be used if IRQMUX7 and IRQMUX 6 are being used for DMA PCREQ and PCGNT. 0000 = EEPROM SDA routed on IRQMUX6 pin (default) 0000 = PCGNT routed on IRQMUX6 pin when bit 3 of the *system control register* is 1 0001 = IRQ1 routed on IRQMUX6 pin 0010 = IRQ2 routed on IRQMUX6 pin : 1111 = IRQ15 routed on IRQMUX6 pin

§ These bits are global in nature and should be accessed only through function 0.
‡ This bit is cleared only by the assertion of G_RST.

Table 36. IRQMUX Routing Register Description (continued)

BIT	TYPE	FUNCTION
23–20‡	R/W	IRQMUX5 routing. These bits select 1 of 15 interrupts that the IRQMUX5 signal may be routed. When these bits are in the default state of all 0s, then no routing is selected. 0000 = No IRQ routing selected (default) 0001 = CardBus audio (CBAUDIO) routed on IRQMUX5 pin 0010 =GPE on IRQMUX5 pin 0011 = IRQ3 routed on IRQMUX5 pin : 1111 = IRQ15 routed on IRQMUX5 pin
19–16‡	R/W	IRQMUX4 routing. These bits select 1 of 15 interrupts that the IRQMUX4 signal may be routed. When these bits are in the default state of all 0s, then no routing is selected. 0000 = No IRQ routing selected (default) 0001 = ZVSTAT is routed on IRQMUX4 pin 0010 = RI_OUT is routed on IRQMUX4 pin 0011 = IRQ3 routed on IRQMUX4 pin 0100 = IRQ4 routed on IRQMUX4 pin : 1111 = IRQ15 routed on IRQMUX4 pin
15–12‡	R/W	IRQMUX3 routing. These bits select 1 of 15 interrupts that the IRQMUX3 signal may be routed. When these bits are in the default state of all 0s, then no routing is selected. 0000 = No IRQ routing selected (default) 0001 = LEDA or LEDB routed on IRQMUX3 pin 0010 = RI_OUT routed on IRQMUX3 pin 0011 = IRQ3 routed on IRQMUX3 pin 0100 = IRQ4 routed on IRQMUX3 pin : 1111 = IRQ15 routed on IRQMUX3
11–8‡	R/W	IRQMUX2 routing. These bits select 1 of 15 interrupts that the IRQMUX2 signal may be routed. When these bits are in the default state of all 0s, then no routing is selected. 0000 = No IRQ routing selected (default) 0001 = LEDB routed on IRQMUX2 pin 0010 = Zoomed Video Pixel Clock (PCLK) Input 0011 = IRQ3 routed on IRQMUX2 pin : 1111 = IRQ15 routed on IRQMUX2 pin
7–4‡	R/W	IRQMUX1 routing. These bits select 1 of 15 interrupts that the IRQMUX1 signal may be routed. When these bits are in the default state of all 0s, then no routing is selected. 0000 = No IRQ routing selected (default) 0001 = LEDA routed on IRQMUX1 pin 0010 = IRQ2 routed on IRQMUX1 pin 0011 = IRQ3 routed on IRQMUX1 pin : 1111 = IRQ15 routed on IRQMUX1 pin
3–0‡	R/W	IRQMUX0 routing. These bits select 1 of 15 interrupts that the IRQMUX0 signal may be routed. When these bits are in the default state of all 0s, then no routing is selected. 0000 = No IRQ routing selected (default) 0001 = INTB routed on IRQMUX0 pin 0010 = IRQ2 routed on IRQMUX0 pin 0011 = IRQ3 routed on IRQMUX0 pin : 1111 = IRQ15 routed on IRQMUX0 pin

‡ This bit is cleared only by the assertion of G_RST.

retry status register

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
Description: The contents of this register enable the retry time-out counters and display the retry expiration status. The flags are set when the PCI1450 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.

Table 37. Retry Status Register Description

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

§ These bits are global in nature and should be accessed only through function 0.
‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

card control register

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
Description: This 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 PCI1450 *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 PCI1450 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.

Table 38. Card Control Register Description

BIT	TYPE	FUNCTION
7‡§	R/W	RIENB. 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‡	R/W	ZVENABLE. 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	R/W	Reserved.
4–3	R	Reserved. These bits default to 0.
2‡	R/W	AUD2MUX. CardBus Audio-to-IRQMUX. When this bit is set, the CAUDIO CardBus signal is routed to the corresponding IRQMUX terminal. Function 0, A_CAUDIO, is routed to IRQMUX0 and function 1, B_CAUDIO, is routed to IRQMUX1. If this bit is set for both functions, then function 0 gets routed. 0 = CAUDIO set to CAUDPWM on IRQMUX pin (default) 1 = CAUDIO is not routed.
1‡	R/W	SPKROUTEN. 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‡	R/W	IFG. 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.
‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

device control 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
Description: This 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.

Table 39. Device Control Register Description

BIT	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‡§	R/W	3VCAPABLE. 3-V socket capable force bit. 0 = Not 3-V capable 1 = 3-V capable (default)
5‡	R/W	IO16R2. Diagnostic bit. This bit defaults to 1.
4	R	Reserved. This bit returns 0 when read. A write has no effect.
3‡§	R/W	TEST. TI test bit. Write only 0 to this bit. This bit can be set to shorten the interrogation counter.
2–1‡§	R/W	INTMODE. Interrupt mode. These bits select the interrupt signaling mode. The interrupt mode bits are encoded: 00 = Parallel PCI interrupts only 01 = Parallel IRQ & parallel PCI interrupts 10 = IRQ serialized interrupts & parallel PCI interrupts $\overline{\text{INTA}}$ and $\overline{\text{INTB}}$ 11 = IRQ & PCI serialized interrupts (default)
0‡§	R/W	Reserved. NAND tree enable bit. There is a NAND tree diagnostic structure in the PCI1450, and it tests only the pins that are inputs or I/Os. Any output only terminal on the PCI1450 is excluded from the NAND tree test.

§ These bits are global in nature and should be accessed only through function 0.
‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

diagnostic register

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
Description: This register is provided for internal Texas Instruments test purposes.

Table 40. Diagnostic Register Description

BIT	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‡§	R/W	DIAG. Diagnostic RETRY_DIS. Delayed transaction disable.
3‡§	R/W	DIAG. Diagnostic RETRY_EXT. Extends the latency from 16 to 64.
2‡§	R/W	DIAG. Diagnostic DISCARD_TIM_SEL_CB. Set = 2^{10}, Reset = 2^{15}
1‡§	R/W	DIAG. Diagnostic DISCARD_TIM_SEL_PCI. Set = 2^{10}, Reset = 2^{15}
0‡§	R/W	ASYNC_CSC. 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.
‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

socket DMA register 0

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

Table 41. Socket DMA Register 0 Description

BIT	TYPE	FUNCTION
31–2	R	Reserved. These bits return 0s when read.
1–0‡	R/W	DREQPIN. 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 = $\overline{\text{Socket}}$ not configured for DMA (default) 01 = $\overline{\text{DREQ}}$ uses $\overline{\text{SPKR}}$ 10 = $\overline{\text{DREQ}}$ uses $\overline{\text{IOIS16}}$ 11 = $\overline{\text{DREQ}}$ uses $\overline{\text{INPACK}}$

‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

socket DMA register 1

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
Description: 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.

Table 42. Socket DMA Register 1 Description

BIT	TYPE	FUNCTION
31–16	R	Reserved. These bits return 0s when read.
15–4‡	R/W	DMABASE. 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 hard-wired to 0, forcing this window to within the lower 64K bytes of I/O address space. The lower four bits are hard-wired to 0, and are included in the address decode. Thus, the window is aligned to a natural 16-byte boundary
3‡	R	EXTMODE. Extended addressing. This feature is not supported by the PCI1450, and always returns a 0.
2–1‡	R/W	XFERSIZE. 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‡	R/W	DDMAEN. DDMA registers decode enable. Enables the decoding of the *distributed DMA registers* based upon the value of DMABASE. 0 = Disabled (default) 1 = Enabled

‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

capability ID register

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
Description: This 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.

next item pointer register

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
Description: The contents of this register indicate the next item in the linked list of the PCI power management capabilities. Since the PCI1450 functions only include one capabilities item, this register returns 0s when read.

power management capabilities register

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
Description: This register contains information on the capabilities of the PC Card function related to power management. Both PCI1450 CardBus bridge functions support D0, D2, and D3 power states.

Table 43. Power Management Capabilities Register Description

BIT	TYPE	FUNCTION
15† 14–11	R/W R	PME support. This 5-bit field indicates the power states from which the PCI1450 device functions may assert $\overline{\text{PME}}$. A 0b (zero) for any bit indicates that the function cannot assert the $\overline{\text{PME}}$ signal while in that power state. These five bits return 0Fh when read. Each of these bits is described below: Bit 15 – defaults to the value 1 indicating the $\overline{\text{PME}}$ signal can be asserted from the $D3_{cold}$ state. This bit is R/W because wake-up support from $D3_{cold}$ is contingent on the system providing an auxiliary power source to the V_{CC} terminals. If the system designer chooses not to provide an auxiliary power source to the V_{CC} terminals for $D3_{cold}$ wake-up support, then BIOS should write a 0 to this bit. Bit 14 – contains the value 1 indicating that the $\overline{\text{PME}}$ signal can be asserted from the $D3_{hot}$ state. Bit 13 – contains the value 1 to indicate that the $\overline{\text{PME}}$ signal can be asserted from the D2 state. Bit 12 – contains the value 1 to indicate that the $\overline{\text{PME}}$ signal can be asserted from the D1 state. Bit 11 – contains the value 1 indicating that the $\overline{\text{PME}}$ signal can be asserted from the D0 state.
10	R	D2_Support. This bit returns a 1 when read, indicating that the function supports the D2 device power state.
9	R	D1_Support. This bit returns a 1 when read, indicating that the function supports the D1 device power state.
8–6	R	Reserved. These bits return 000b when read.
5	R	DSI. Device specific initialization. This bit returns 0 when read.
4	R	AUX_PWR. Auxiliary power source. This bit is meaningful only if bit 15 ($D3_{cold}$ supporting $\overline{\text{PME}}$) is set. When this bit is set, it indicates that support for $\overline{\text{PME}}$ in $D3_{cold}$ requires auxiliary power supplied by the system by way of a proprietary delivery vehicle. 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	R	PMECLK. When this bit is 1, it indicates that the function relies on the presence of the PCI clock for $\overline{\text{PME}}$ operation. When this bit is 0, it indicates that no PCI clock is required for the function to generate $\overline{\text{PME}}$. Functions that do not support PME generation in any state must return 0 for this field.
2–0	R	Version. 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.

† This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$ 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 $\overline{\text{G\_RST}}$.

power management control/status register

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: 000000h
Description: This register determines and changes the current power state of the PCI1450 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 $\overline{G\_RST}$ only bits.

Table 44. Power Management Control/Status Register Description

BIT	TYPE	FUNCTION
15†	R/WC	PMESTAT. 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	R	DATASCALE. 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	R	DATASEL. 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†	R/W	PME enable. 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 $\overline{G\_RST}$.
7–2	R	Reserved. These bits return 0s when read.
1–0	R/W	PWRSTATE. 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 = $D3_{hot}$

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

power management control/status register bridge support extensions

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

Table 45. PMCSR_BSE Bridge Support Extensions

BIT	TYPE	FUNCTION
7	R	BPCC_Enable. Bus power/clock control enable. This bit returns 1 when read. This bit is encoded as: 0 = Bus power/clock control is disabled. 1 = Bus power/clock control is enabled (default). A 0 indicates that the bus power/clock control policies defined in the PCI 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	R	$\overline{B2\_B3}$. 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 will be stopped (B2). (Default)
5–0	R	Reserved. These bits return 0s when read.

$\overline{\text{GPE}}$ control/status register

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
Description: If the $\overline{\text{GPE}}$ (general-purpose event) function is programmed onto the IRQMUX5 pin by writing 0010b to bits 23–20 of the multifunction routing register (PCI offset 8Ch), then this register may be used to program which events will cause $\overline{\text{GPE}}$ to be asserted and report the status.

Table 46. $\overline{\text{GPE}}$ Control/Status Register Description

BIT	TYPE	FUNCTION
15–11	R	Reserved. These bits return 0s when read.
10‡	R/WC	ZV1_STS. PC Card socket 1 status. This bit is set on a change in status of the ZVENABLE bit in function 1.
9‡	R/WC	ZV0_STS. PC Card socket 0 status. This bit is set on a change in status of the ZVENABLE bit in function 0.
8‡	R/WC	VPP12_STS. 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	R	Reserved. These bits return 0s when read.
2‡	R/W	ZV1_EN. 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‡	R/W	ZV0_EN. 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‡	R/W	VPP12_EN. 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.

‡ This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$.

ExCA compatibility registers (functions 0 and 1)

The ExCA (exchangeable card architecture) registers implemented in the PCI1450 are register-compatible with the popular 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 19 for an ExCA I/O mapping illustration. Table 47 identifies each ExCA register and its respective ExCA offset.

The TI PCI1450 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 20 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 PCI1450 to ensure that all possible PCI1450 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 $\overline{\text{G\_RST}}$. This is necessary to retain device context when transitioning from D3 to D0.

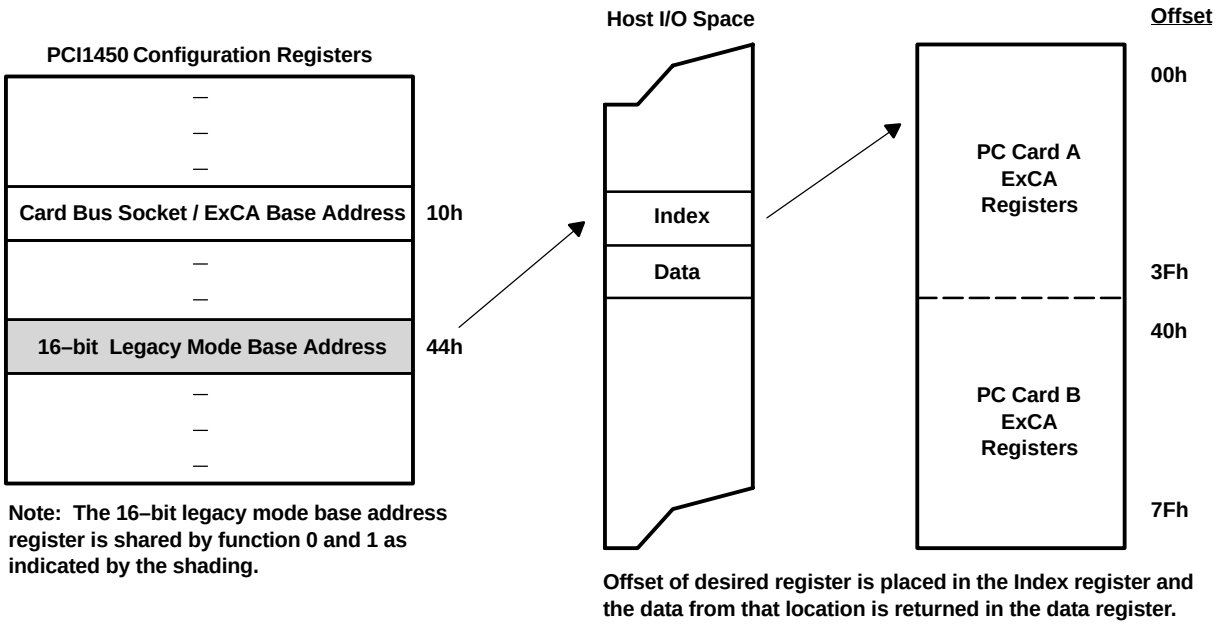


Figure 19. ExCA Register Access Through I/O

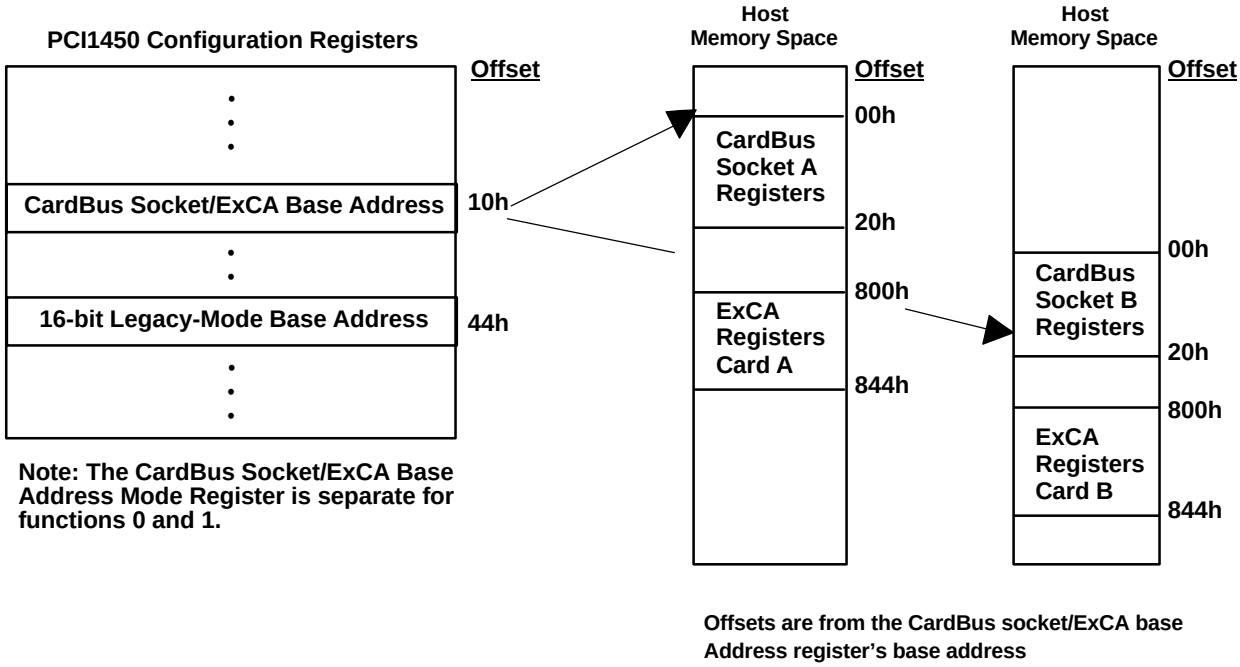


Figure 20. ExCA Register Access Through Memory

Table 47. 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
Reserved	826	26	66
Reserved	827	27	67

† One or more bits in this register are cleared only by the assertion of $\overline{\text{G\_RST}}$ 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 $\overline{\text{G\_RST}}$.

Table 47. ExCA Registers and Offsets (continued)

REGISTER NAME	PCI MEMORY ADDRESS OFFSET	ExCA OFFSET (CARD A)	ExCA OFFSET (CARD B)
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	-	-

ExCA identification and revision register (Index 00h)

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
Description: 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.

Table 48. 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 PCI1450. The PCI1450 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 PCI1450. Host software may read this field to determine compatibility to the 82365SL register set. This field defaults to 0100b upon reset.

ExCA interface status register (Index 01h)

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
Description: 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.

Table 49. 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 PCI1450 whether or not the memory card is write protected. Further, write protection for an entire PCI1450 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.

ExCA power control register (Index 02h)

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
Description: 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.

Table 50. 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 PCI1450. 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 PCI1450 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 $\overline{\text{G\_RST}}$ 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 $\overline{\text{G\_RST}}$.

ExCA interrupt and general control register (Index 03h)

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

Table 51. 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 $\overline{\text{G\_RST}}$ 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 $\overline{\text{G\_RST}}$.

ExCA card status-change register (Index 04h)

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
Description: 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*.

Table 52. 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 PCI1450 interrupt was due to a change on the READY signal at the PC Card interface indicating that PC Card is now able 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 PCI1450 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 PCI1450 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 PCI1450 is configured for ring indicate operation this bit indicates the status of the RI pin.

† This bit is cleared only by the assertion of $\overline{\text{G\_RST}}$ 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 $\overline{\text{G\_RST}}$.

ExCA card status-change interrupt configuration register (Index 05h)

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

Table 53. 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 $\overline{\text{G\_RST}}$ 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 $\overline{\text{G\_RST}}$.

ExCA address window enable register (Index 06h)

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
Description: 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 PCI1450 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*.

Table 54. 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

ExCA I/O window control register (Index 07h)

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

Table 55. 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 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 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 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 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

ExCA I/O window 0 & 1 start-address low-byte register (Index 08h, 0Ch)

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O window 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
Description: 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.

ExCA I/O window 0 & 1 start-address high-byte register (Index 09h, ODh)

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O window 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
Description: 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.

ExCA I/O window 0 & 1 end-address low-byte register (Index 0Ah, 0Eh)

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O window 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
Description: 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.

ExCA I/O window 0 & 1 end-address high-byte register (Index 0Bh, 0Fh)

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O window 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
Description: 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.

ExCA memory window 0–4 start-address low-byte register (Index 10h/18h/20h/28h/30h)

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory window 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
Description: 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.

ExCA memory window 0–4 start-address high-byte register (Index 11h/19h/21h/29h/31h)

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory window 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
Description: 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.

Table 56. ExCA Memory Window 0–4 Start-Address High-Byte Register 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.

ExCA memory window 0–4 end-address low-byte register (Index 12h/1Ah/22h/2Ah/32h)

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory window 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
Description: 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.

ExCA memory window 0–4 end-address high-byte register (Index 13h/1Bh/23h/2Bh/33h)

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory window 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
Description: 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.

Table 57. ExCA Memory Window 0–4 End-Address High-Byte Register 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.

ExCA memory window 0–4 offset-address low-byte register (Index 14h/1Ch/24h/2Ch/34h)

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory window 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
Description: 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.

ExCA memory window 0–4 offset-address high-byte register (15h/1Dh/25h/2Dh/35h)

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
Description: 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.

Table 58. ExCA Memory Window 0–4 Offset-Address High-Byte Register 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.

ExCA I/O window 0 & 1 offset-address low-byte register (Index 36h, 38h)

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O window 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 memory 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
Description: 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.

ExCA I/O window 0 & 1 offset-address high-byte register (Index 37h, 39h)

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O window 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 memory 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
Description: 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.

ExCA card detect and general control register (Index 16h)

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
Description: 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 59 describes each bit in the *ExCA card detect and general control register*.

Table 59. 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.

ExCA global control register (Index 1Eh)

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
Description: 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.

Table 60. 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 PCI1450 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 PCI1450 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 PCI1450 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 PCI1450 is in power-down mode. In power-down mode the PCI1450 card outputs are 3-stated until an active cycle is executed on the card interface. Following an active cycle the outputs are again 3-stated. The PCI1450 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

ExCA memory window 0–4 page register (Index 40h, 41h, 42h, 43h, 44h)

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory window 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 window 0–4 page**
Type: Read/Write
Offset: CardBus Socket Address + 840h, 841h, 842h, 843h, 844h
Default: 00h
Description: 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.

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 PCI1450 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 21 below. Table 61 illustrates the location of the socket registers in relation to the CardBus socket/ExCA base address.

Table 61. 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 $\overline{\text{G\_RST}}$ 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 $\overline{\text{G\_RST}}$.

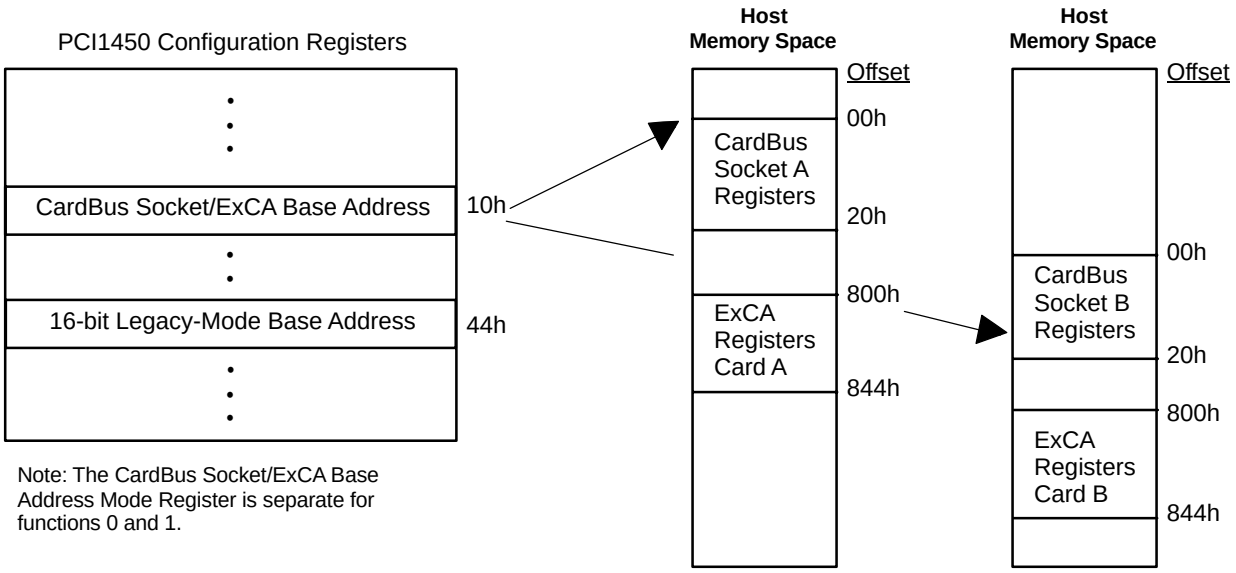


Figure 21. Accessing CardBus Socket Registers Through PCI Memory

socket event register

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
Description: 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.

Table 62. 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 PCI1450 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 PCI1450 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 PCI1450 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 $\overline{\text{G\_RST}}$ 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 $\overline{\text{G\_RST}}$.

socket mask 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
Description: This register allows software to control the CardBus card events which generate a status change interrupt. Table 63 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*.

Table 63. 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 $\overline{\text{G\_RST}}$ 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 $\overline{\text{G\_RST}}$.

socket present state 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
Description: 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 PCI1450 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.

Table 64. 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 PCI1450 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 PCI1450 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 PCI1450 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 PCI1450 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

Table 64. Socket Present State Register Description (continued)

BIT	TYPE	FUNCTION
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 PCI1450. 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

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

socket force event register

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
Description: 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.

Table 65. 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 PCI1450 re-interrogates 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.

socket control register

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
Description: This register provides control of the voltages applied to the socket's V_{PP} and V_{CC}. The PCI1450 ensures that the socket is powered up only at acceptable voltages when a CardBus card is inserted.

Table 66. 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 PCI1450 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 PCI1450 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 $\overline{\text{G\_RST}}$ 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 $\overline{\text{G\_RST}}$.

socket power management register

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
Description: 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.

Table 67. 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.

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 68 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 PCI1450 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 PCI1450 will implement these obsolete register bits as nonfunctional, read-only bits. The reserved registers shown in Table 68 are implemented as read-only, and return 0s when read. Writes to reserved registers have no effect.

Table 68. 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		

DMA current address / base address register

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
Description: 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.

DMA page register

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

Register: **DMA page**
Type: Read/Write
Offset: DMA Base Address + 02h
Default: 0000h
Size: One byte
Description: 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.

DMA current count / base count register

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
Description: 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.

DMA command register

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

Table 69. 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 PCI1450, 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.

DMA status register

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

Table 70. 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 PCI1450, 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 PCI1450, 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

DMA request register

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
Description: 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.

DMA mode register

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
Description: This register is used to set the DMA transfer mode.

Table 71. DDMA Mode Register Description

BIT	TYPE	FUNCTION
7–6	R/W	DMAMODE. Mode select bits. The PCI1450 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 PCI1450 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 PCI1450 PC Card interface to memory, and a memory read transfer moves data from memory to the PCI1450 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.

DMA master clear register

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
Description: This register is used to reset the DDMA controller, and resets all DDMA registers.

DMA multichannel mask register

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
Description: The PCI1450 uses only the least-significant bit of this register to mask the PC Card DMA channel. The PCI1450 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.

Table 72. 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)

absolute maximum ratings over operating temperature ranges (unless otherwise noted)†

Supply voltage range, V_{CC} . . . . . –0.5 V to 4.6 V
Clamping voltage range, V_{CCP}, V_{CCA}, V_{CCB}, V_{CCZ}, V_{CCI} . . . . . –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_{CCZ} + 0.5 V
Misc . . . . . –0.5 V to V_{CCI} + 0.5 V
Fail safe . . . . . –0.5 V to V_{CC} + 0.5 V
Output voltage range, V_O: 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_{CCZ} + 0.5 V
Misc . . . . . –0.5 V to V_{CCI} + 0.5 V
Fail safe . . . . . –0.5 V to V_{CC} + 0.5 V
Input clamp current, I_{IK} ($V_I < 0$ or $V_I > V_{CC}$) (see Note 1) . . . . . ±20 mA
Output clamp current, I_{OK} ($V_O < 0$ or $V_O > V_{CC}$) (see Note 2) . . . . . ±20 mA
Storage temperature range, T_{stg} . . . . . –65°C to 150°C
Virtual junction temperature, T_J . . . . . 150°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. Applies for external input and bidirectional buffers. $V_I > V_{CC}$ does not apply to fail-safe terminals. PCI terminals are measured with respect to V_{CCP} instead of V_{CC}. PC Card terminals are measured with respect to V_{CCA} or V_{CCB}. ZV terminals are measured with respect to V_{CCZ}, and miscellaneous signals are measured with respect to V_{CCI}. The limit specified applies for a dc condition.
2. Applies for external output and bidirectional buffers. $V_O > V_{CC}$ does not apply to fail-safe terminals. PCI terminals are measured with respect to V_{CCP} instead of V_{CC}. PC Card terminals are measured with respect to V_{CCA} or V_{CCB}. ZV terminals are measured with respect to V_{CCZ}, and miscellaneous signals are measured with respect to V_{CCI}. The limit specified applies for a dc condition.

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	Commercial	3.3 V	3	3.3	3.6	V
			5 V	4.75	5	5.25	
$V_{CCA/B}$	PC Card I/O voltage	Commercial	3.3 V	3	3.3	3.6	V
			5 V	4.75	5	5.25	
V_{CCZ}	ZV Port I/O voltage	Commercial	3.3 V	3	3.3	3.6	V
			5 V	4.75	5	5.25	
V_{CCI}	Miscellaneous 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		$V_{CCA/B}$	
		ZV		2		V_{CCZ}	V
		Misc‡	5 V	2		V_{CCI}	V
		Fail safe§	3.3 V	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	
		ZV		0		0.8	V
		Misc‡	5 V	0		0.8	V
		Fail safe§	3.3 V	0		0.8	V
V_I	Input voltage	PCI	3.3 V	0		V_{CCP}	V
		PC Card	5 V	0		$V_{CCA/B}$	
		ZV		0		V_{CCZ}	
		Misc‡	5 V	0		V_{CCI}	
		Fail safe§	3.3 V	0		V_{CC}	
V_O¶	Output voltage	PCI	3.3 V	0		V_{CCP}	V
		PC Card	5 V	0		$V_{CCA/B}$	
		ZV		0		V_{CCZ}	
		Misc‡	5 V	0		V_{CCI}	
		Fail safe§	3.3 V	0		V_{CC}	V
t_t	Input transition times (t_r and t_f)	PCI and PC Card		1		4	ns
		ZV, misc, and fail safe		0		6	
T_A	Operating ambient temperature range			0	25	70	°C
T_J#	Virtual junction temperature			0	25	115	°C

† Applies to external inputs and bidirectional buffers without hysteresis
‡ Miscellaneous pins are V13, W13, Y13, U12, V12, W12, U11, V11, W11, Y11, Y10, W10, Y09, W09, V09, U09, Y08, all IRQMUXx pins, LEDAx pins, SUSPEND, SPKROUT, RI_OUT, INTA, INTB, and power switch control pins.
§ Fail-safe pins are A11, B14, C09, G03, H20, U03, W06, and Y03 (card detect and voltage sense pins).
¶ Applies to external output buffers
# These junction temperatures reflect simulation conditions. The customer is responsible for verifying junction temperature.
NOTE 3: Unused pins (input or I/O) must be held high or low to prevent them from floating.

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		
		ZV		I_{OH} = –4 mA	V_{CC}–0.6		
		MISC		I_{OH} = –4 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	
		ZV		I_{OL} = 4 mA		0.5	
		MISC		I_{OL} = 4 mA		0.5	
		$\overline{SERR}$		I_{OL} = 12 mA		0.5	
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_{CCZ}. For miscellaneous pins, V_I = V_{CCI}.
‡ For I/O pins, input leakage (I_{IL} and I_{IH}) includes I_{OZ} leakage of the disabled output.

NOTES: 4. V_{OH} and I_{OL} are not tested on $\overline{SERR}$ (GFN pin U19, GJG pin U19)and $\overline{RI\_OUT}$ (GFN pin Y13, GJG pin P12) because they are open-drain outputs.
5. I_{IH} is not tested on LATCH (GFN pin W12, GJG pin V11) because it is pulled up with an internal resistor.

PCI clock/reset timing requirements over recommended ranges of supply voltage and operating free-air temperature (see Figure 23 and Figure 24)

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

PCI timing requirements over recommended ranges of supply voltage and operating free-air temperature (see Note 7, Figure 22, and Figure 25)

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 AD31–0, C/$\overline{\text{BE3}}$–0, $\overline{\text{FRAME}}$, $\overline{\text{TRDY}}$, $\overline{\text{IRDY}}$, $\overline{\text{STOP}}$, 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.

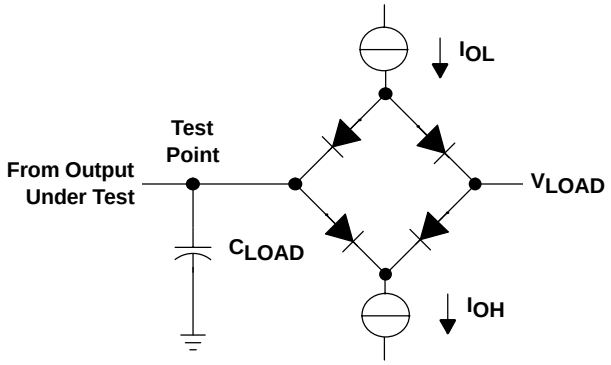
PARAMETER MEASUREMENT INFORMATION

LOAD CIRCUIT PARAMETERS

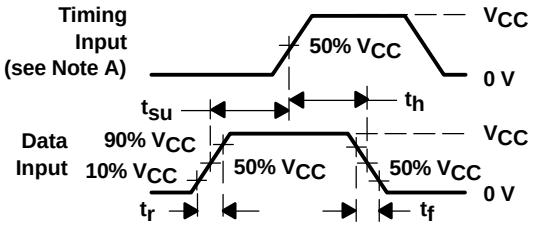
TIMING PARAMETER		C_{LOAD}† (pF)	I_{OL} (mA)	I_{OH} (mA)	V_{LOAD}‡ (V)
t_{en}	t_{PZH}	50	8	–8	0
	t_{PZL}				3
t_{dis}	t_{PHZ}	50	8	–8	1.5
	t_{PLZ}				
t_{pd}		50	8	–8	‡

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

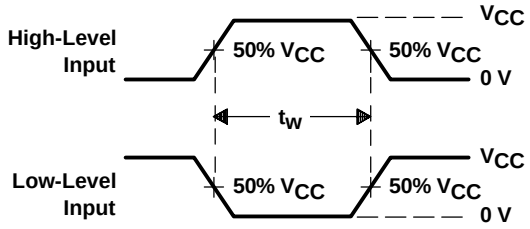
‡ $\frac{V_{LOAD} - V_{OL}}{I_{OL}}$ = 50 Ω, where V_{OL} = 0.6 V, I_{OL} = 8 mA



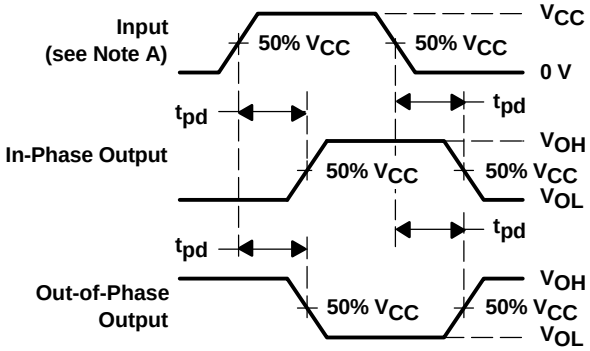
LOAD CIRCUIT



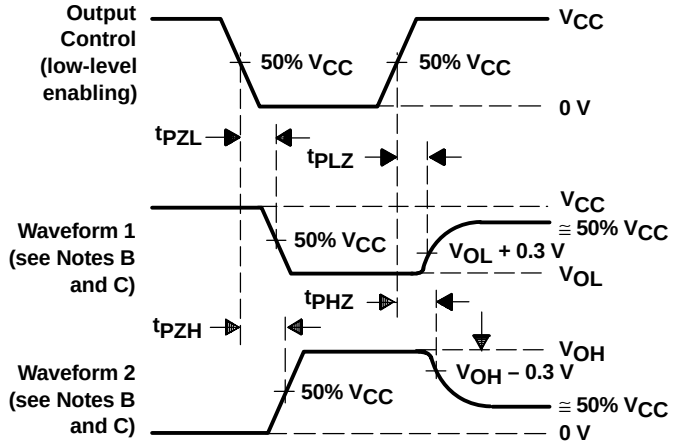
VOLTAGE WAVEFORMS
SETUP AND HOLD TIMES
INPUT RISE AND FALL TIMES



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 22. Load Circuit and Voltage Waveforms

PCI BUS PARAMETER MEASUREMENT INFORMATION

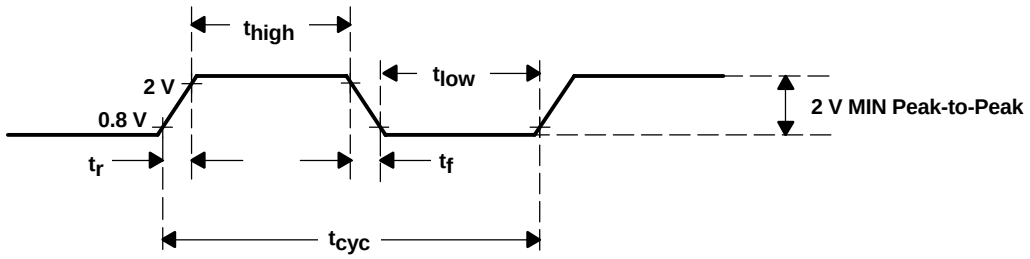


Figure 23. PCLK Timing Waveform

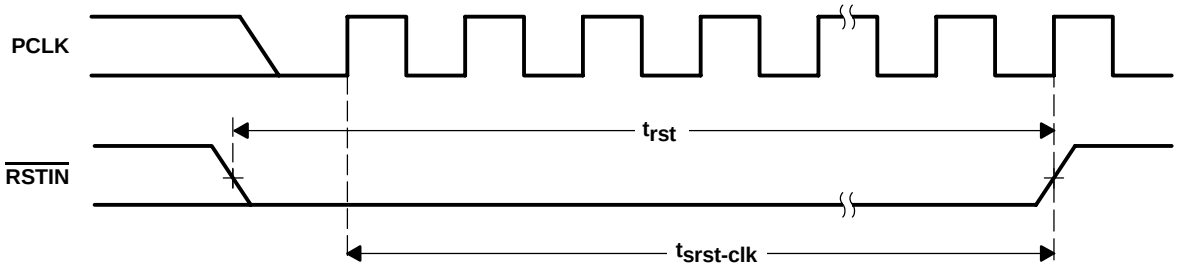


Figure 24. $\overline{\text{RSTIN}}$ Timing Waveforms

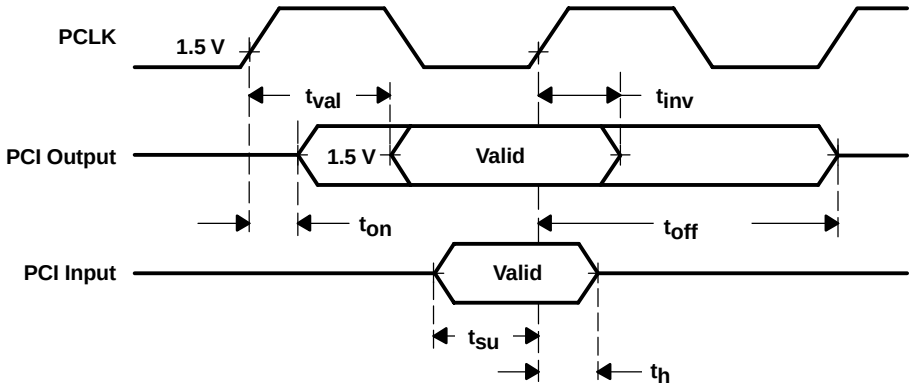


Figure 25. Shared Signals Timing Waveforms

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 73 shows address setup time in PCLK cycles and nanoseconds for I/O and memory cycles. Table 74 and Table 75 show command active time in PCLK cycles and nanoseconds for I/O and memory cycles. Table 76 shows address hold time in PCLK cycles and nanoseconds for I/O and memory cycles.

Table 73. 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 74. 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 75. 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 76. 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

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 26)

		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.

timing requirements over recommended ranges of supply voltage and operating free-air temperature, I/O cycles (see Figure 27)

		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

switching characteristics over recommended ranges of supply voltage and operating free-air temperature, miscellaneous (see Figure 28)

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	
		$\overline{\text{IREQ}}$ to IRQ15–IRQ3	T28		30	
		$\overline{\text{STSCHG}}$ to IRQ15–IRQ3			30	

PC Card PARAMETER MEASUREMENT INFORMATION

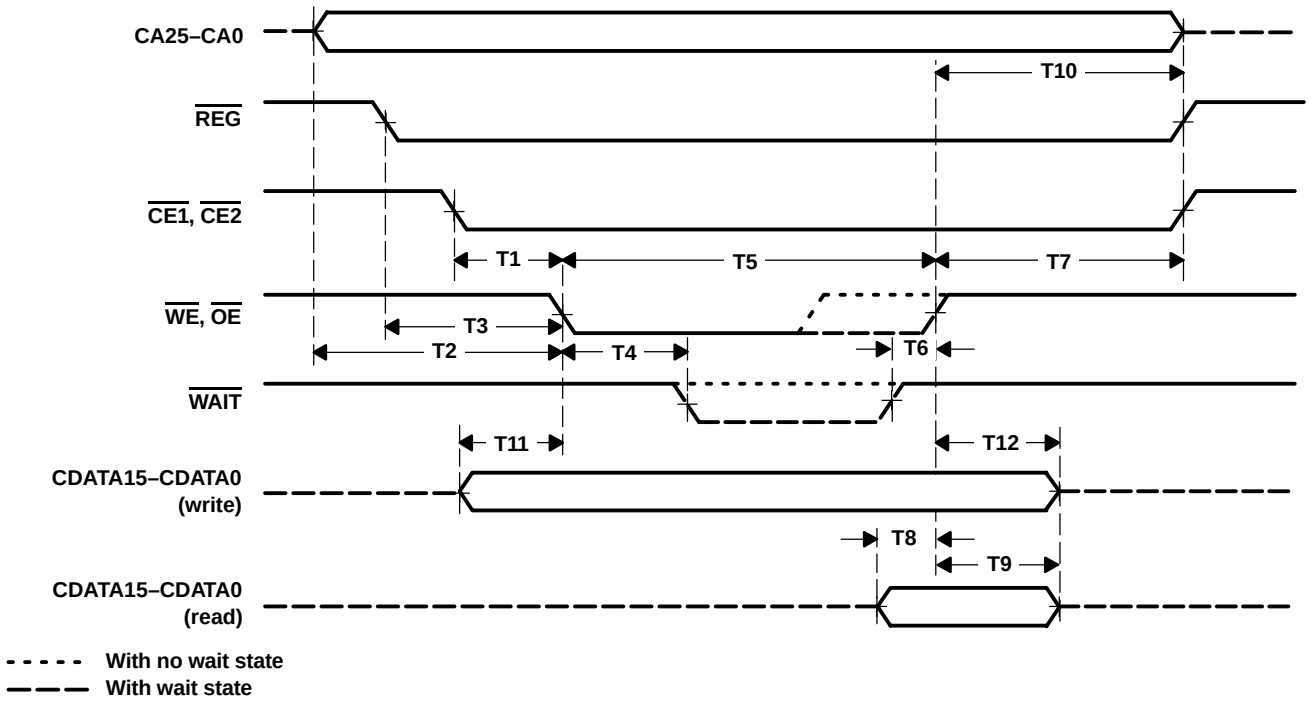


Figure 26. PC Card Memory Cycle

PC Card PARAMETER MEASUREMENT INFORMATION

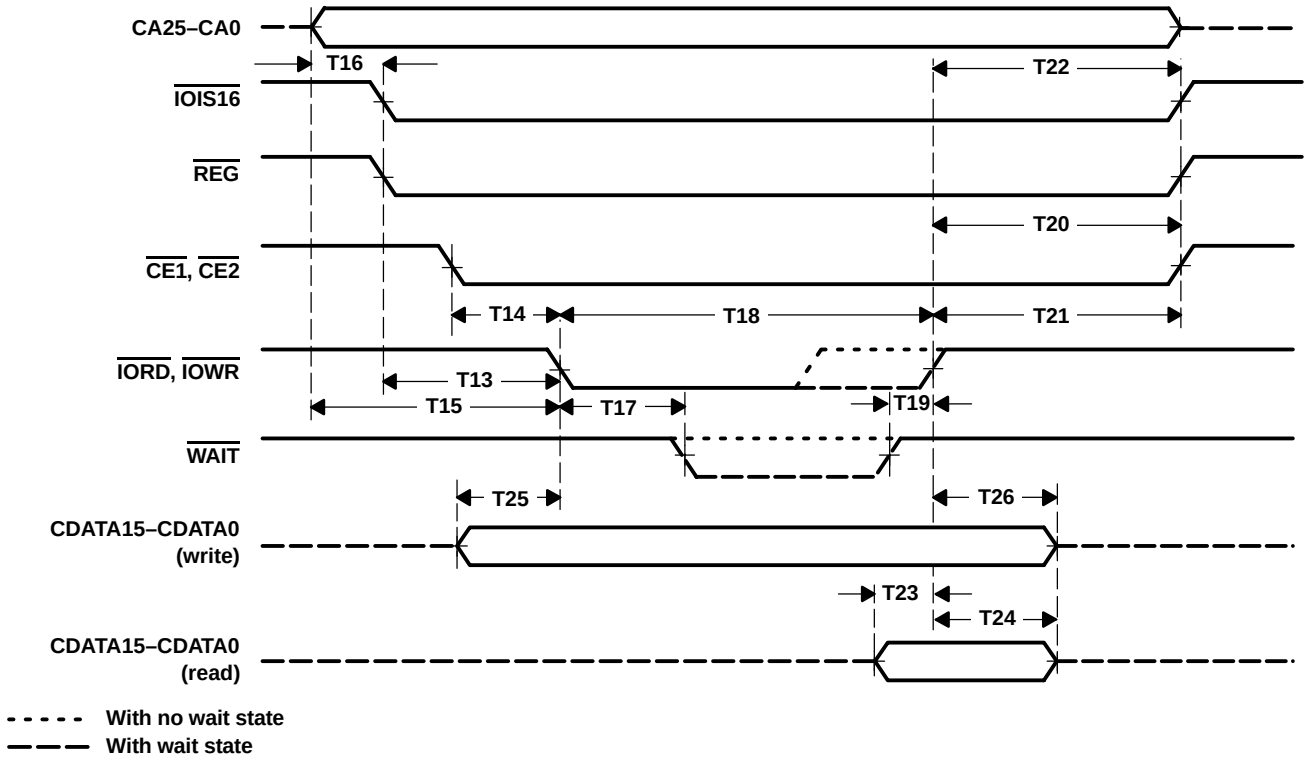


Figure 27. PC Card I/O Cycle

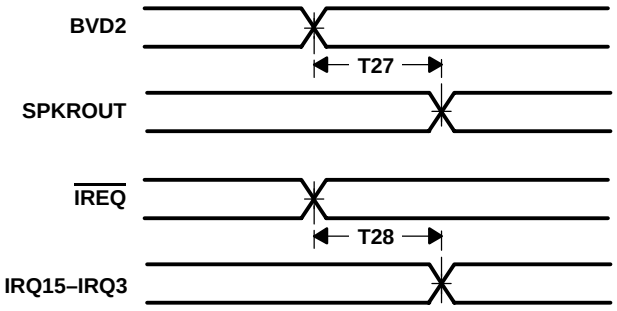
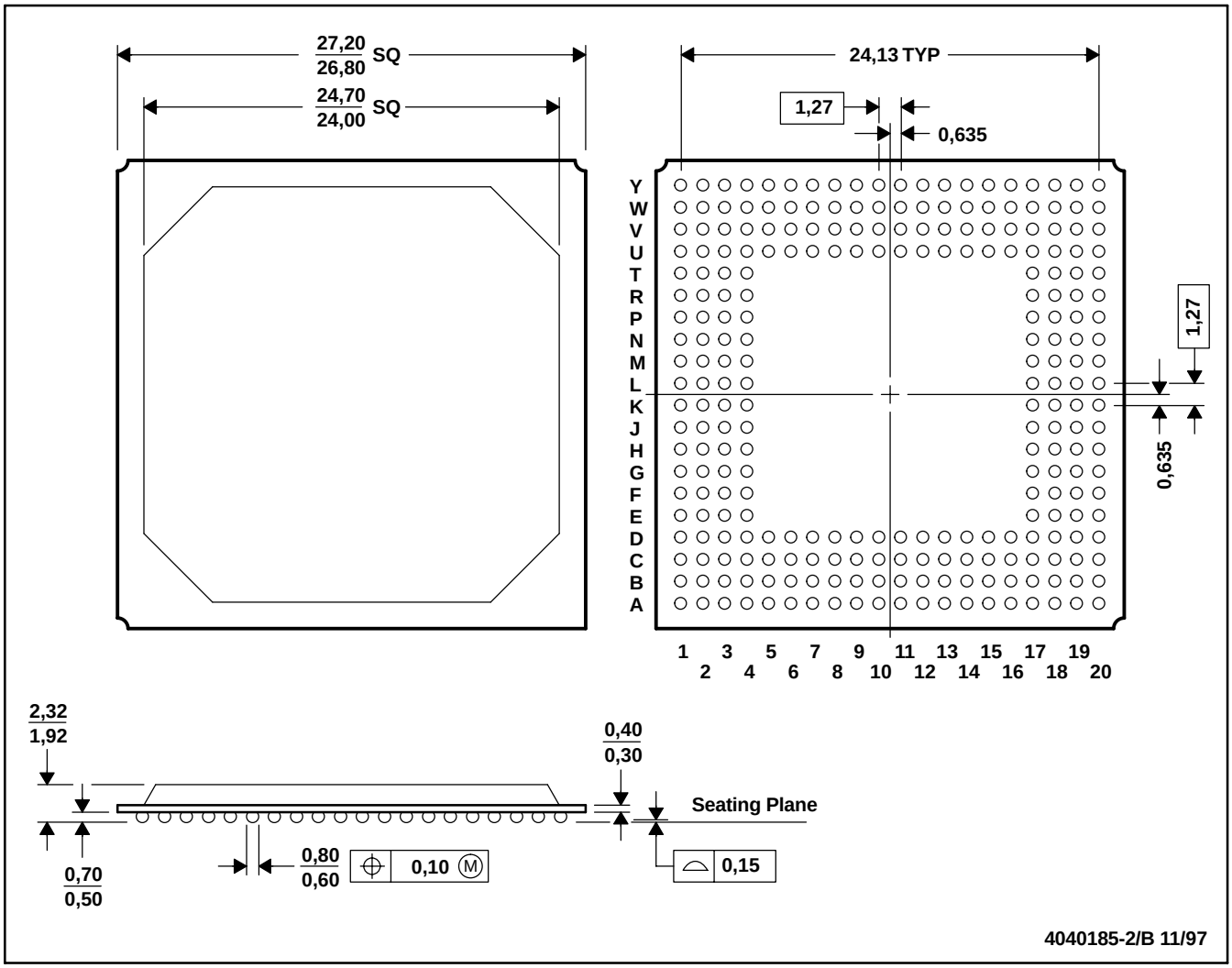


Figure 28. Miscellaneous PC Card Delay Times

MECHANICAL DATA

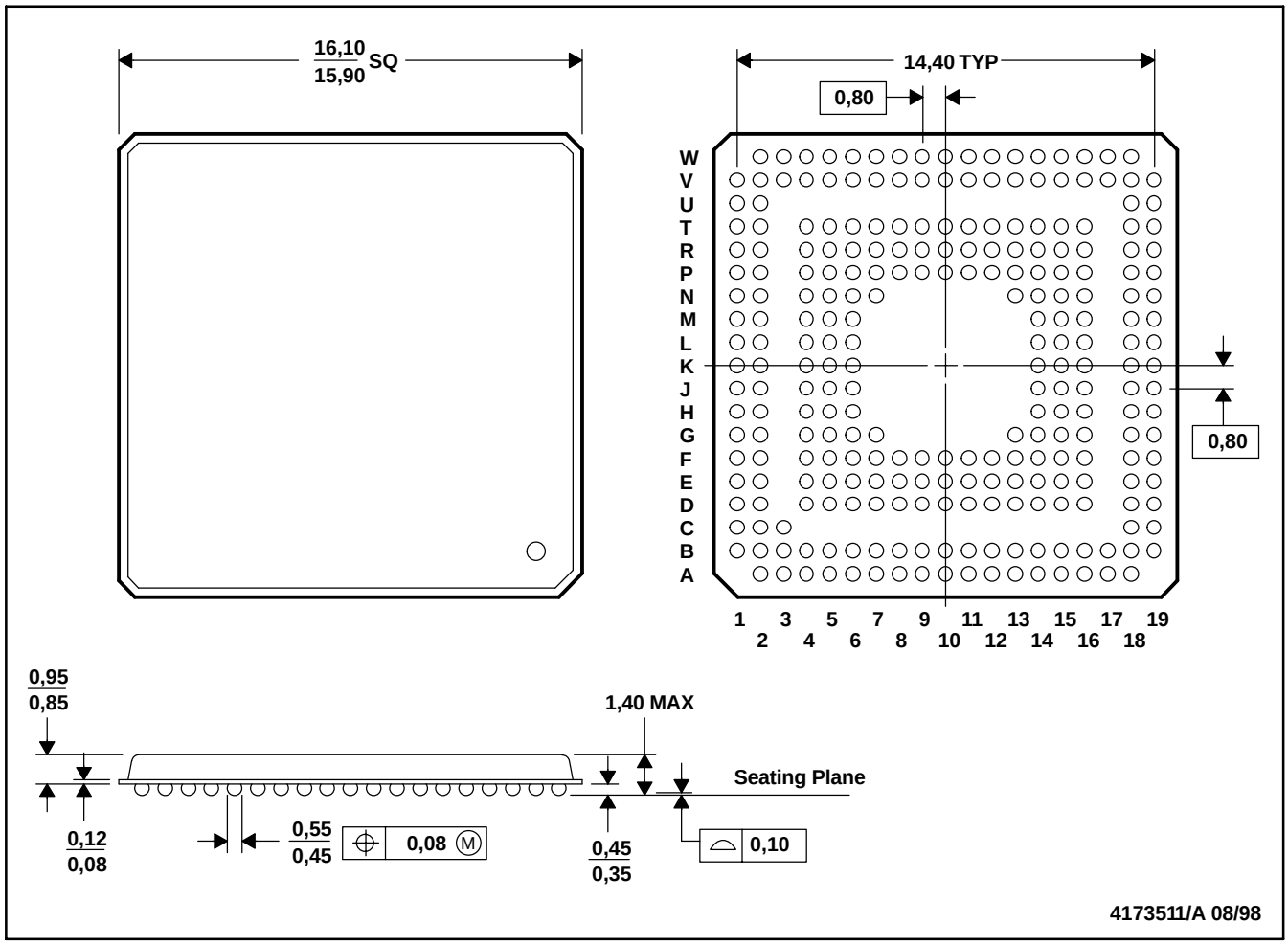
GFN (S-PBGA-N256) **PLASTIC BALL GRID ARRAY**



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

MECHANICAL DATA

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 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.

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.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265

Copyright © 2003, Texas Instruments Incorporated

Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

- View PCI1450GJG on WIN SOURCE
- Texas Instruments Information

Optimize Your Supply Chain with WIN SOURCE Solutions

- Global Sourcing Solution
- Obsolete Management
- Cost Control Management
- Shortage Management
- Alternative Solution
- Excess Inventory Management