TEXAS INSTRUMENTS

PCI1451 GJG

PC Card Controller

Data Manual

1999

PCIBus Solutions

TEXAS INSTRUMENTS

Printed in U.S.A.
11/99

SCPS054

PCI1451
PC Card Controller
Data Manual

Literature Number: SCPS054
November 1999

IMPORTANT NOTICE

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

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

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

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

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

Contents

List of Illustrations

List of Tables

1 Introduction

1.1 Description

The Texas Instruments PCI1451 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 PCI1451 provides features which make 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 define the new 32-bit PC Card, CardBus, capable of full 32-bit data transfers at 33 MHz. The PCI1451 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 PCI1451 is compliant with the latest *PCI Bus Power Management Specification*. It is also compliant with the *PCI Local Bus Specification*, 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 PCI1451 is register compatible with the Intel 82365SL-DF ExCA controller. The PCI1451 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 PCI1451 can also be programmed to accept fast posted writes to improve system bus utilization.

The PCI1451 provides an internally buffered zoom video (ZV) path. This reduces the design effort of PC board manufacturers to add a ZV compatible solution and ensures compliance with the CardBus loading specifications. 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 PCI1451 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 PCI1451 inputs must be pulled up using a 43-kΩ resistor.

1.2 Features

The PCI1451 supports the following features:

- 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
- Two PC Card or CardBus slots with hot insertion and removal
- Serial interface to TI™ TPS2206 dual power switch
- 132 Mbyte/sec. burst transfers to maximize data throughput on both the PCI bus and the CardBus bus
- Serialized IRQ with PCI interrupts
- Eight programmable multifunction pins
- Interrupt modes supported: serial ISA/serial PCI, serial ISA/parallel PCI, parallel PCI only.
- Serial EEPROM interface for loading subsystem ID and subsystem vendor ID
- Zoomed video with internal buffering
- Dedicated pin for PCI $\overline{\text{CLKRUN}}$
- Four general-purpose event registers
- Multifunction PCI device with separate configuration space for each socket
- Five PCI memory windows and two I/O windows available to each PC Card16 socket
- Two I/O windows and two memory windows available to each CardBus socket
- ExCA™-compatible registers are mapped in memory or I/O space
- Distributed DMA and PC/PCI DMA
- Intel™ 82365SL-DF register compatible
- 16-bit DMA on both PC Card sockets
- 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
- Socket activity LED pins
- PCI bus lock ($\overline{\text{LOCK}}$)
- Packaged in a 257-pin Micro-Star BGA

1.3 Related Documents

- *1997 PC Card™ Standard*
- *PCI Bus Power Management Interface Specification* (Revision 1.1)
- *Advanced Configuration and Power Interface (ACPI) Specification* (Revision 2.0)
- *PCI Local Bus Specification* (Revision 2.2)
- *PC 98/99*
- *PCI Bus Interface Specification for PCI-to-CardBus Bridges*
- *PCI Bus Power Management Specification for PCI to CardBus Bridges Specification*

1.4 Ordering Information

ORDERING NUMBER	NAME	VOLTAGE	PACKAGE
PCI1451	PC Card Controller	3.3 V, 5-V tolerant I/Os	257-ball Micro-Star BGA

2 Terminal Descriptions

The PCI1451 is packaged in a 257-ball MicroStar BGA package.

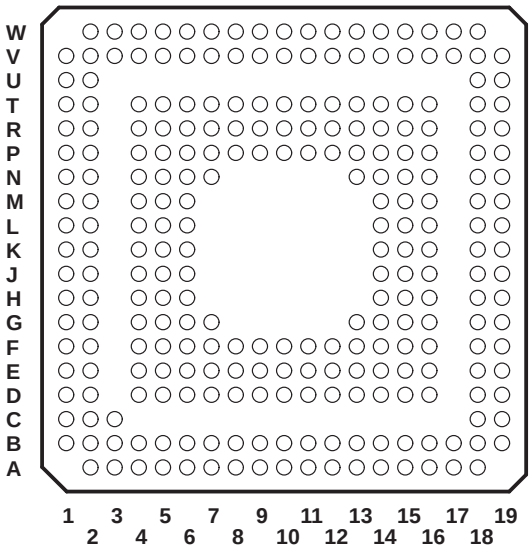


Figure 2–1. PCI1451 GJG Terminal Diagram

Table 2–1 shows the GJG terminal assignments for the CardBus and 16-bit PC Card signal names.

Table 2–2 shows the CardBus PC Card signal names sorted alphanumerically to the GJG terminal number.

Table 2–3 shows the 16-bit PC Card signal names sorted alphanumerically to the GJG terminal number.

Table 2–1. GJG Terminals Sorted Alphanumerically for CardBus // 16-Bit Signals

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

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

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

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

SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.	SIGNAL NAME	TERM. NO.
A_CAD0	A8	A_$\overline{\text{CFRAME}}$	F2	AD26	M16	B_CC/$\overline{\text{BE3}}$	A12
A_CAD1	E8	A_$\overline{\text{CGNT}}$	E4	AD27	M15	B_$\overline{\text{CCD1}}$	J19
A_CAD2	B8	A_$\overline{\text{CINT}}$	K2	AD28	L16	B_$\overline{\text{CCD2}}$	D10
A_CAD3	A7	A_$\overline{\text{CIRDY}}$	F4	AD29	L19	B_CCLK	A17
A_CAD4	G7	A_CPAR	B1	AD30	L18	B_$\overline{\text{CCLKRUN}}$	B10
A_CAD5	D7	A_$\overline{\text{CPERR}}$	D1	AD31	L15	B_$\overline{\text{CDEVSEL}}$	A18
A_CAD6	B7	A_$\overline{\text{CREQ}}$	J4	B_CAD0	J18	B_$\overline{\text{CFRAME}}$	D15
A_CAD7	F7	A_$\overline{\text{CRST}}$	H2	B_CAD1	J15	B_$\overline{\text{CGNT}}$	B18
A_CAD8	A6	A_$\overline{\text{CSERR}}$	K4	B_CAD2	J16	B_$\overline{\text{CINT}}$	E11
A_CAD9	F6	A_$\overline{\text{CSTOP}}$	D2	B_CAD3	H19	B_$\overline{\text{CIRDY}}$	B16
A_CAD10	A5	A_CSTSCHG	L1	B_CAD4	H14	B_CPAR	C19
A_CAD11	B5	A_$\overline{\text{CTRDY}}$	F5	B_CAD5	H16	B_$\overline{\text{CPERR}}$	C18
A_CAD12	A4	A_CVS1	K1	B_CAD6	H18	B_$\overline{\text{CREQ}}$	A13
A_CAD13	D5	A_CVS2	H6	B_CAD7	G13	B_$\overline{\text{CRST}}$	F13
A_CAD14	D4	A_RSVD	B2	B_CAD8	G18	B_$\overline{\text{CSERR}}$	F11
A_CAD15	B4	A_RSVD	E7	B_CAD9	G15	B_$\overline{\text{CSTOP}}$	B19
A_CAD16	B3	A_RSVD	M5	B_CAD10	G14	B_CSTSCHG	A10
A_CAD17	G5	AD0	V12	B_CAD11	F19	B_$\overline{\text{CTRDY}}$	B17
A_CAD18	G2	AD1	T12	B_CAD12	F15	B_CVS1	B11
A_CAD19	G4	AD2	R12	B_CAD13	F16	B_CVS2	A14
A_CAD20	H1	AD3	N13	B_CAD14	E18	B_RSVD	D9
A_CAD21	H4	AD4	T13	B_CAD15	F14	B_RSVD	D18
A_CAD22	H5	AD5	R13	B_CAD16	E16	B_RSVD	H15
A_CAD23	J1	AD6	P13	B_CAD17	A15	C/$\overline{\text{BE0}}$	T14
A_CAD24	J5	AD7	W14	B_CAD18	B15	C/$\overline{\text{BE1}}$	T16
A_CAD25	J6	AD8	R14	B_CAD19	E14	C/$\overline{\text{BE2}}$	R16
A_CAD26	K6	AD9	P14	B_CAD20	B14	C/$\overline{\text{BE3}}$	M14
A_CAD27	L5	AD10	W15	B_CAD21	E13	$\overline{\text{CLKRUN}}$	K18
A_CAD28	M6	AD11	V15	B_CAD22	B13	CLOCK	T11
A_CAD29	M1	AD12	T15	B_CAD23	D13	DATA	V11
A_CAD30	M4	AD13	W16	B_CAD24	D12	$\overline{\text{DEVSEL}}$	U19
A_CAD31	N7	AD14	V16	B_CAD25	E12	$\overline{\text{FRAME}}$	T18
A_CAUDIO	L6	AD15	W17	B_CAD26	D11	GND	A3
A_$\overline{\text{CBLOCK}}$	C2	AD16	R15	B_CAD27	E10	GND	A16
A_CC/$\overline{\text{BE0}}$	D6	AD17	R19	B_CAD28	F9	GND	C1
A_CC/$\overline{\text{BE1}}$	A2	AD18	R18	B_CAD29	A9	GND	D8
A_CC/$\overline{\text{BE2}}$	G6	AD19	P16	B_CAD30	B9	GND	F12
A_CC/$\overline{\text{BE3}}$	J2	AD20	P19	B_CAD31	E9	GND	G1
A_$\overline{\text{CCD1}}$	F8	AD21	P18	B_CAUDIO	F10	GND	G19
A_$\overline{\text{CCD2}}$	L4	AD22	N14	B_$\overline{\text{CBLOCK}}$	D16	GND	M2
A_CCLK	E2	AD23	N15	B_CC/$\overline{\text{BE0}}$	G16	GND	N16
A_$\overline{\text{CCLKRUN}}$	L2	AD24	M19	B_CC/$\overline{\text{BE1}}$	D19	GND	T4
A_$\overline{\text{CDEVSEL}}$	E5	AD25	M18	B_CC/$\overline{\text{BE2}}$	E15	GND	T7

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

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

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

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

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

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

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 2–4. Power Supply

TERMINAL NAME	TERMINAL NO.	FUNCTION
GND	A3, A16, C1, D8, F12, G1, G19, M2, N16, T4, T7, V14, W12	Device ground terminals
V_{CC}	A11, D14, E1, E6, E19, J14, P1, P15, T5, V10, V13	Power supply terminal for core logic (3.3 Vdc)
V_{CCA}	B6, F1, K5	Clamp voltage for PC Card A interface. Indicates Card A signaling environment.
V_{CCB}	B12, F18	Clamp voltage for PC Card B interface. Indicates Card B signaling environment.
V_{CCP}	N18, W13	Clamp voltage for PCI signaling (3.3 Vdc or 5 Vdc)

Table 2–5. PC Card Power Switch

TERMINAL NAME	TERMINAL NO.	I/O	FUNCTION
CLOCK	T11	I/O	3-line power switch clock. Information on the DATA line is sampled at the rising edge of CLOCK. This terminal defaults as an input which means an external clock source must be used. If the internal ring oscillator is used, then an external CLOCK source is not required. The internal oscillator may be enabled by setting bit 27 (P2CCLK) of the system control register (PCI offset 80h, see Section 4.29) to a 1b. A 43-kΩ pulldown resistor should be tied to this terminal.
DATA	V11	O	3-line power switch data. DATA is used to serially communicate socket power-control information to the power switch.
LATCH	W11	O	3-line power switch latch. LATCH is asserted by the PCI4450 to indicate to the PC Card power switch that the data on the DATA line is valid.

Table 2–6. PCI System

TERMINAL NAME	TERMINAL NO.	I/O	FUNCTION
$\overline{\text{CLKRUN}}$	K18	I/O	PCI clock run. $\overline{\text{CLKRUN}}$ is used by the central resource to request permission to stop the PCI clock or to slow it down, and the PCI4450 responds accordingly. If $\overline{\text{CLKRUN}}$ is not implemented, then this termomal should be tied low. $\overline{\text{CLKRUN}}$ is enabled by default by bit 1 (KEEPCLK) in the system control register (PCI offset 80h, see Section 4.29).
PCLK	K15	I	PCI bus clock. PCLK provides timing for all transactions on the PCI bus. All PCI signals are sampled at the rising edge of PCLK.
$\overline{\text{PRST}}$	K19	I	PCI bus reset. When the PCI bus reset is asserted, $\overline{\text{PRST}}$ causes the PCI4450 to place all output buffers in a high-impedance state and reset all internal registers. When $\overline{\text{PRST}}$ is asserted, the device is completely nonfunctional. After $\overline{\text{PRST}}$ is deasserted, the PCI4450 is in its default state. When the SUSPEND mode is enabled, the device is protected from the $\overline{\text{PRST}}$ and the internal registers are preserved. All outputs are placed in a high-impedance state, but the contents of the registers are preserved.
$\overline{\text{GRST}}$	P11	I	Global reset. When the global reset is asserted, the $\overline{\text{GRST}}$ signal causes the PCI4450 to place all output buffers in a high-impedance state and reset all internal registers. When $\overline{\text{GRST}}$ is asserted, the device is completely in its default state. For systems that require wake-up from D3, $\overline{\text{GRST}}$ will normally be asserted only during initial boot. $\overline{\text{PRST}}$ should be asserted following initial boot so that PME context is retained when transitioning from D3 to D0. For systems that do not require wake-up from D3, $\overline{\text{GRST}}$ should be tied to $\overline{\text{PRST}}$.

Table 2–7. PCI Address and Data

TERMINAL NAME	TERMINAL NO.	I/O	FUNCTION
AD31	L15	I/O	PCI address/data bus. These signals make up the multiplexed PCI address and data bus on the primary interface. During the address phase of a primary bus PCI cycle, AD31–AD0 contain a 32-bit address or other destination information. During the data phase, AD31–AD0 contain data.
AD30	L18		
AD29	L19		
AD28	L16		
AD27	M15		
AD26	M16		
AD25	M18		
AD24	M19		
AD23	N15		
AD22	N14		
AD21	P18		
AD20	P19		
AD19	P16		
AD18	R18		
AD17	R19		
AD16	R15		
AD15	W17		
AD14	V16		
AD13	W16		
AD12	T15		
AD11	V15		
AD10	W15		
AD9	P14		
AD8	R14		
AD7	W14		
AD6	P13		
AD5	R13		
AD4	T13		
AD3	N13		
AD2	R12		
AD1	T12		
AD0	V12		
C/$\overline{\text{BE3}}$	M14	I/O	PCI bus commands and byte enables. These signals are multiplexed on the same PCI terminals. During the address phase of a primary bus PCI cycle, C/$\overline{\text{BE3}}$–C/$\overline{\text{BE0}}$ define the bus command. During the data phase, this 4-bit bus is used as byte enables. The byte enables determine which byte paths of the full 32-bit data bus carry meaningful data. C/$\overline{\text{BE0}}$ applies to byte 0 (AD7–AD0), C/$\overline{\text{BE1}}$ applies to byte 1 (AD15–AD8), C/$\overline{\text{BE2}}$ applies to byte 2 (AD23–AD16), and C/$\overline{\text{BE3}}$ applies to byte 3 (AD31–AD24).
C/$\overline{\text{BE2}}$	R16		
C/$\overline{\text{BE1}}$	T16		
C/$\overline{\text{BE0}}$	T14		
PAR	V17	I/O	PCI bus parity. In all PCI bus read and write cycles, the PCI4450 calculates even parity across the AD31–AD0 and C/$\overline{\text{BE3}}$–C/$\overline{\text{BE0}}$ buses. As an initiator during PCI cycles, the PCI4450 outputs this parity indicator with a one-PCLK delay. As a target during PCI cycles, the calculated parity is compared to the initiator's parity indicator. A compare error results in the assertion of a parity error ($\overline{\text{PERR}}$).

Table 2–8. PCI Interface Control

TERMINAL NAME	TERMINAL NO.	I/O	FUNCTION
$\overline{\text{DEVSEL}}$	U19	I/O	PCI device select. The PCI4450 asserts $\overline{\text{DEVSEL}}$ to claim a PCI cycle as the target device. As a PCI initiator on the bus, the PCI4450 monitors $\overline{\text{DEVSEL}}$ until a target responds. If no target responds before timeout occurs, then the PCI4450 terminates the cycle with an initiator abort.
$\overline{\text{FRAME}}$	T18	I/O	PCI cycle frame. $\overline{\text{FRAME}}$ is driven by the initiator of a bus cycle. $\overline{\text{FRAME}}$ is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When $\overline{\text{FRAME}}$ is deasserted, the PCI bus transaction is in the final data phase.
$\overline{\text{GNT}}$	K14	I	PCI bus grant. $\overline{\text{GNT}}$ is driven by the PCI bus arbiter to grant the PCI4450 access to the PCI bus after the current data transaction has completed. $\overline{\text{GNT}}$ may or may not follow a PCI bus request, depending on the PCI bus parking algorithm.
$\overline{\text{LOCK}}$ (MFUNC7)	N19	I/O	PCI bus lock. MFUNC7/$\overline{\text{LOCK}}$ can be configured as PCI $\overline{\text{LOCK}}$ and used to gain exclusive access downstream. Since this functionality is not typically used, other functions may be accessed through this terminal. MFUNC7/$\overline{\text{LOCK}}$ defaults to and can be configured through the multifunction routing status register (PCI offset 8Ch, see Section 4.36).
IDSEL/MFUNC7	N19	I	Initialization device select. IDSEL selects the PCI4450 during configuration space accesses. IDSEL can be connected to one of the upper 24 PCI address lines on the PCI bus. If the LATCH terminal (W12/W11) has an external pulldown resistor, then this terminal is configurable as MFUNC7 and IDSEL defaults to the AD23 terminal.
$\overline{\text{IRDY}}$	T19	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.
$\overline{\text{PERR}}$	V18	I/O	PCI parity error indicator. $\overline{\text{PERR}}$ is driven by a PCI device to indicate that calculated parity does not match PAR when $\overline{\text{PERR}}$ is enabled through bit 6 of the command register (PCI offset 04h, see Section 4.4).
$\overline{\text{REQ}}$	L14	O	PCI bus request. $\overline{\text{REQ}}$ is asserted by the PCI4450 to request access to the PCI bus as an initiator.
$\overline{\text{SERR}}$	W18	O	PCI system error. $\overline{\text{SERR}}$ is an output that is pulsed from the PCI4450 when enabled through bit 8 of the command register (PCI offset 04h, see Section 4.4), indicating a system error has occurred. The PCI4450 need not be the target of the PCI cycle to assert this signal. When $\overline{\text{SERR}}$ is enabled by bit 1 in the bridge control register (PCI offset 3Eh, see Section 4.25), this signal also pulses, indicating that an address parity error has occurred on a CardBus interface.
$\overline{\text{STOP}}$	V19	I/O	PCI cycle stop signal. $\overline{\text{STOP}}$ is driven by a PCI target to request the initiator to stop the current PCI bus transaction. $\overline{\text{STOP}}$ is used for target disconnects and is commonly asserted by target devices that do not support burst data transfers.
$\overline{\text{TRDY}}$	U18	I/O	PCI target ready. $\overline{\text{TRDY}}$ indicates the primary bus target's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCLK when both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. Until both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted, wait states are inserted.

Table 2–9. System Interrupt

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

Table 2–10. PC/PCI DMA

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

Table 2–11. Zoomed Video

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

Table 2–12. Miscellaneous

TERMINAL NAME	TERMINAL NO.	I/O	FUNCTION
MFUNC0	W10	I/O	Multifunction terminal 0. Defaults as a general-purpose input (GPI0), and can be programmed to perform various functions. See Section 4.36, *Multifunction Routing Status Register,* for configuration details.
MFUNC1	P10	I/O	Multifunction terminal 1. Defaults as a general-purpose input (GPI1), and can be programmed to perform various functions. See Section 4.36, *Multifunction Routing Status Register,* for configuration details.
MFUNC2	P9	I/O	Multifunction terminal 2. Defaults as a general-purpose input (GPI2), and can be programmed to perform various functions. See Section 4.36, *Multifunction Routing Status Register,* for configuration details.
MFUNC3	R9	I/O	Multifunction terminal 3. Defaults as a general-purpose input (GPI3), and can be programmed to perform various functions. See Section 4.36, *Multifunction Routing Status Register,* for configuration details.
MFUNC4	T9	I/O	Multifunction terminal 4. Defaults as a high–impedance reserved input, and can be programmed to perform various functions. See Section 4.36, *Multifunction Routing Status Register,* for configuration details.
MFUNC5	W5	I/O	Multifunction terminal 5. Defaults as a high-impedance reserved input, and can be programmed to perform various functions. See Section 4.36, *Multifunction Routing Status Register,* for configuration details.
MFUNC6	R5	I/O	Multifunction terminal 6. Defaults as a high-impedance reserved input, and can be programmed to perform various functions. See Section 4.36, *Multifunction Routing Status Register,* for configuration details.
IDSEL/MFUNC7	N19	I/O	IDSEL and multifunction terminal 7. Defaults as IDSEL, but may be used as a multifunction terminal. See Section 4.36, *Multifunction Routing Status Register* and Section 3.4, *PC Card Applications Overview,* for configuration details.
SCL	V9	I/O	Serial ROM clock. This terminal provides the SCL serial clock signaling in a two-wire serial ROM implementation, and is sensed at reset for serial ROM detection.
SDA	W9	I/O	Serial ROM data. This terminal provides the SDA serial data signaling in a two-wire serial ROM implementation.
$\overline{\text{SPKROUT}}$	T10	O	Speaker output. SPKROUT is the output to the host system that can carry $\overline{\text{SPKR}}$ or CAUDIO through the PCI4450 from the PC Card interface. SPKROUT is driven as the XOR combination of card $\overline{\text{SPKR}}$//CAUDIO inputs.
$\overline{\text{SUSPEND}}$	R10	I	Suspend. $\overline{\text{SUSPEND}}$ is used to protect the internal registers from clearing when $\overline{\text{PRST}}$ is asserted. See Section 3.6.7, *Suspend Mode* for details.

Table 2–13. 16-Bit PC Card Address and Data (slots A and B)

TERMINAL NAME	TERMINAL NO. SLOT A†	TERMINAL NO. SLOT B‡	I/O	FUNCTION
A25	G4	E14	O	PC Card address. 16-bit PC Card address lines. A25 is the most significant bit.
A24	G5	A15		
A23	F2	D15		
A22	F5	B17		
A21	E5	A18		
A20	D2	B19		
A19	C2	D16		
A18	B2	D18		
A17	B3	E16		
A16	E2	A17		
A15	F4	B16		
A14	D1	C18		
A13	B1	C19		
A12	G6	E15		
A11	A4	F15		
A10	F6	G15		
A9	D4	E18		
A8	A2	D19		
A7	G2	B15		
A6	H1	B14		
A5	H4	E13		
A4	H5	B13		
A3	J1	D13		
A2	J5	D12		
A1	J6	E12		
A0	K6	D11		
D15	A6	G18	I/O	PC Card data. 16-bit PC Card data lines. D15 is the most significant bit.
D14	E7	H15		
D13	B7	H18		
D12	G7	H14		
D11	B8	J16		
D10	N7	E9		
D9	M4	B9		
D8	M6	F9		
D7	F7	G13		
D6	D7	H16		
D5	A7	H19		
D4	E8	J15		
D3	A8	J18		
D2	M5	D9		
D1	M1	A9		
D0	L5	E10		

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

Table 2–14. 16-Bit PC Card Interface Control (slots A and B)

TERMINAL NAME	TERMINAL NO. SLOT A†	TERMINAL NO. SLOT B‡	I/O	FUNCTION
BVD1 (STSCHG/RI)	L1	A10	I	Battery voltage detect 1. BVD1 is generated by 16-bit memory PC Cards that include batteries. BVD1 and BVD2 indicate the condition of the batteries on a memory PC Card. Both BVD1 and BVD2 are kept high when the battery is good. When BVD2 is low and BVD1 is high, the battery is weak and should be replaced. When BVD1 is low, the battery is no longer serviceable and the data in the memory PC Card is lost. See Section 5.6, *ExCA Card Status-Change Interrupt Configuration Register*, for the enable bits. See Section 5.5, *ExCA Card Status-Change Register* and Section 5.2, *ExCA Interface Status Register*, for the status bits for this signal. Status change. STSCHG is used to alert the system to a change in the READY, write protect, or battery voltage dead condition of a 16-bit I/O PC Card. Ring indicate. RI is used by 16-bit modem cards to indicate a ring detection.
BVD2 (SPKR)	L6	F10	I	Battery voltage detect 2. BVD2 is generated by 16-bit memory PC Cards that include batteries. BVD2 and BVD1 indicate the condition of the batteries on a memory PC Card. Both BVD1 and BVD2 are high when the battery is good. When BVD2 is low and BVD1 is high, the battery is weak and should be replaced. When BVD1 is low, the battery is no longer serviceable and the data in the memory PC Card is lost. See Section 5.6, *ExCA Card Status-Change Interrupt Configuration Register*, for the enable bits. See Section 5.5, *ExCA Card Status-Change Register* and Section 5.2, *ExCA Interface Status Register*, for the status bits for this signal. Speaker. SPKR is an optional binary audio signal available only when the card and socket have been configured for the 16-bit I/O interface. The audio signals from cards A and B are combined by the PCI4450 and are output on SPKROUT. DMA request. BVD2 can be used as the DMA request signal during DMA operations to a 16-bit PC Card that supports DMA. The PC Card asserts BVD2 to indicate a request for a DMA operation.
CD1 CD2	F8 L4	J19 D10	I	PC Card detect 1 and PC Card detect 2. CD1 and CD2 are internally connected to ground on the PC Card. When a PC Card is inserted into a socket, CD1 and CD2 are pulled low. For signal status, see Section 5.2, *ExCA Interface Status Register*.
CE1 CE2	D6 A5	G16 G14	O	Card enable 1 and card enable 2. CE1 and CE2 enable even- and odd-numbered address bytes. CE1 enables even-numbered address bytes, and CE2 enables odd-numbered address bytes.
INPACK	J4	A13	I	Input acknowledge. INPACK is asserted by the PC Card when it can respond to an I/O read cycle at the current address. DMA request. INPACK can be used as the DMA request signal during DMA operations from a 16-bit PC Card that supports DMA. If used as a strobe, then the PC Card asserts this signal to indicate a request for a DMA operation.
IORD	D5	F16	O	I/O read. IORD is asserted by the PCI4450 to enable 16-bit I/O PC Card data output during host I/O read cycles. DMA write. IORD is used as the DMA write strobe during DMA operations from a 16-bit PC Card that supports DMA. The PCI4450 asserts IORD during DMA transfers from the PC Card to host memory.
IOWR	B4	F14	O	I/O write. IOWR is driven low by the PCI4450 to strobe write data into 16-bit I/O PC Cards during host I/O write cycles. DMA read. IOWR is used as the DMA write strobe during DMA operations from a 16-bit PC Card that supports DMA. The PCI4450 asserts IOWR during transfers from host memory to the PC Card.
OE	B5	F19	O	Output enable. OE is driven low by the PCI4450 to enable 16-bit memory PC Card data output during host memory read cycles. DMA terminal count. OE is used as terminal count (TC) during DMA operations to a 16-bit PC Card that supports DMA. The PCI4450 asserts OE to indicate TC for a DMA write operation.
READY (IREQ)	K2	E11	I	Ready. The ready function is provided by READY when the 16-bit PC Card and the host socket are configured for the memory-only interface. READY is driven low by the 16-bit memory PC Cards to indicate that the memory card circuits are busy processing a previous write command. READY is driven high when the 16-bit memory PC Card is ready to accept a new data transfer command. Interrupt request. IREQ is asserted by a 16-bit I/O PC Card to indicate to the host that a device on the 16-bit I/O PC Card requires service by the host software. IREQ is high (deasserted) when no interrupt is requested.

† Terminal name for slot A is preceded with A_. For example, the full name for terminal B5 is A_OE.

‡ Terminal name for slot B is preceded with B_. For example, the full name for terminal F19 is B_OE.

Table 2–14. 16-Bit PC Card Interface Control (slots A and B) (continued)

TERMINAL NAME	TERMINAL NO. SLOT A†	TERMINAL NO. SLOT B‡	I/O	FUNCTION
$\overline{\text{REG}}$	J2	A12	O	Attribute memory select. $\overline{\text{REG}}$ remains high for all common memory accesses. When $\overline{\text{REG}}$ is asserted, access is limited to attribute memory ($\overline{\text{OE}}$ or $\overline{\text{WE}}$ active) and to the I/O space ($\overline{\text{IORD}}$ or $\overline{\text{IOWR}}$ 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. $\overline{\text{REG}}$ is used as a DMA acknowledge ($\overline{\text{DACK}}$) during DMA operations to a 16-bit PC Card that supports DMA. The PCI4450 asserts $\overline{\text{REG}}$ to indicate a DMA operation. $\overline{\text{REG}}$ is used in conjunction with the DMA read ($\overline{\text{IOWR}}$) or DMA write ($\overline{\text{IORD}}$) strobes to transfer data.
RESET	H2	F13	O	PC Card reset. RESET forces a hard reset to a 16-bit PC Card.
$\overline{\text{WAIT}}$	K4	F11	I	Bus cycle wait. $\overline{\text{WAIT}}$ is driven by a 16-bit PC Card to delay the completion of (i.e., extend) the memory or I/O cycle in progress.
$\overline{\text{WE}}$	E4	B18	O	Write enable. $\overline{\text{WE}}$ is used to strobe memory write data into 16-bit memory PC Cards. $\overline{\text{WE}}$ is also used for memory PC Cards that employ programmable memory technologies. DMA terminal count. $\overline{\text{WE}}$ is used as TC during DMA operations to a 16-bit PC Card that supports DMA. The PCI4450 asserts $\overline{\text{WE}}$ to indicate TC for a DMA read operation.
WP ($\overline{\text{IOIS16}}$)	L2	B10	I	Write protect. WP applies to 16-bit memory PC Cards. WP reflects the status of the write-protect switch on 16-bit memory PC Cards. For 16-bit I/O cards, WP is used for the 16-bit port ($\overline{\text{IOIS16}}$) function. I/O is 16 bits. $\overline{\text{IOIS16}}$ applies to 16-bit I/O PC Cards. $\overline{\text{IOIS16}}$ 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.
$\overline{\text{VS1}}$ $\overline{\text{VS2}}$	K1 H6	B11 A14	I/O	Voltage sense 1 and voltage sense 2. $\overline{\text{VS1}}$ and $\overline{\text{VS2}}$, 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 C1 is A_$\overline{\text{WE}}$.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminal A19 is B_$\overline{\text{WE}}$.

Table 2–15. CardBus PC Card Interface System (slots A and B)

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

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

Table 2–16. CardBus PC Card Address and Data (slots A and B)

TERMINAL NAME	TERMINAL NO. SLOT A†	TERMINAL NO. SLOT B‡	I/O	FUNCTION
CAD31	N7	E9	I/O	PC Card address and data. These signals make up the multiplexed CardBus address and data bus on the CardBus interface. During the address phase of a CardBus cycle, CAD31–CAD0 contain a 32-bit address. During the data phase of a CardBus cycle, CAD31–CAD0 contain data. CAD31 is the most significant bit.
CAD30	M4	B9		
CAD29	M1	A9		
CAD28	M6	F9		
CAD27	L5	E10		
CAD26	K6	D11		
CAD25	J6	E12		
CAD24	J5	D12		
CAD23	J1	D13		
CAD22	H5	B13		
CAD21	H4	E13		
CAD20	H1	B14		
CAD19	G4	E14		
CAD18	G2	B15		
CAD17	G5	A15		
CAD16	B3	E16		
CAD15	B4	F14		
CAD14	D4	E18		
CAD13	D5	F16		
CAD12	A4	F15		
CAD11	B5	F19		
CAD10	A5	G14		
CAD9	F6	G15		
CAD8	A6	G18		
CAD7	F7	G13		
CAD6	B7	H18		
CAD5	D7	H16		
CAD4	G7	H14		
CAD3	A7	H19		
CAD2	B8	J16		
CAD1	E8	J15		
CAD0	A8	J18		
CC/$\overline{\text{BE3}}$	J2	A12	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–CAD16), and CC/$\overline{\text{BE3}}$ applies to byte 3 (CAD31–CAD24).
CC/$\overline{\text{BE2}}$	G6	E15		
CC/$\overline{\text{BE1}}$	A2	D19		
CC/$\overline{\text{BE0}}$	D6	G16		
CPAR	B1	C19	I/O	CardBus parity. In all CardBus read and write cycles, the PCI4450 calculates even parity across the CAD and CC/$\overline{\text{BE}}$ buses. As an initiator during CardBus cycles, the PCI4450 outputs CPAR with a one-CCLK delay. As a target during CardBus cycles, the calculated parity is compared to the initiator'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 C2 is A_CPAR.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminal B20 is B_CPAR.

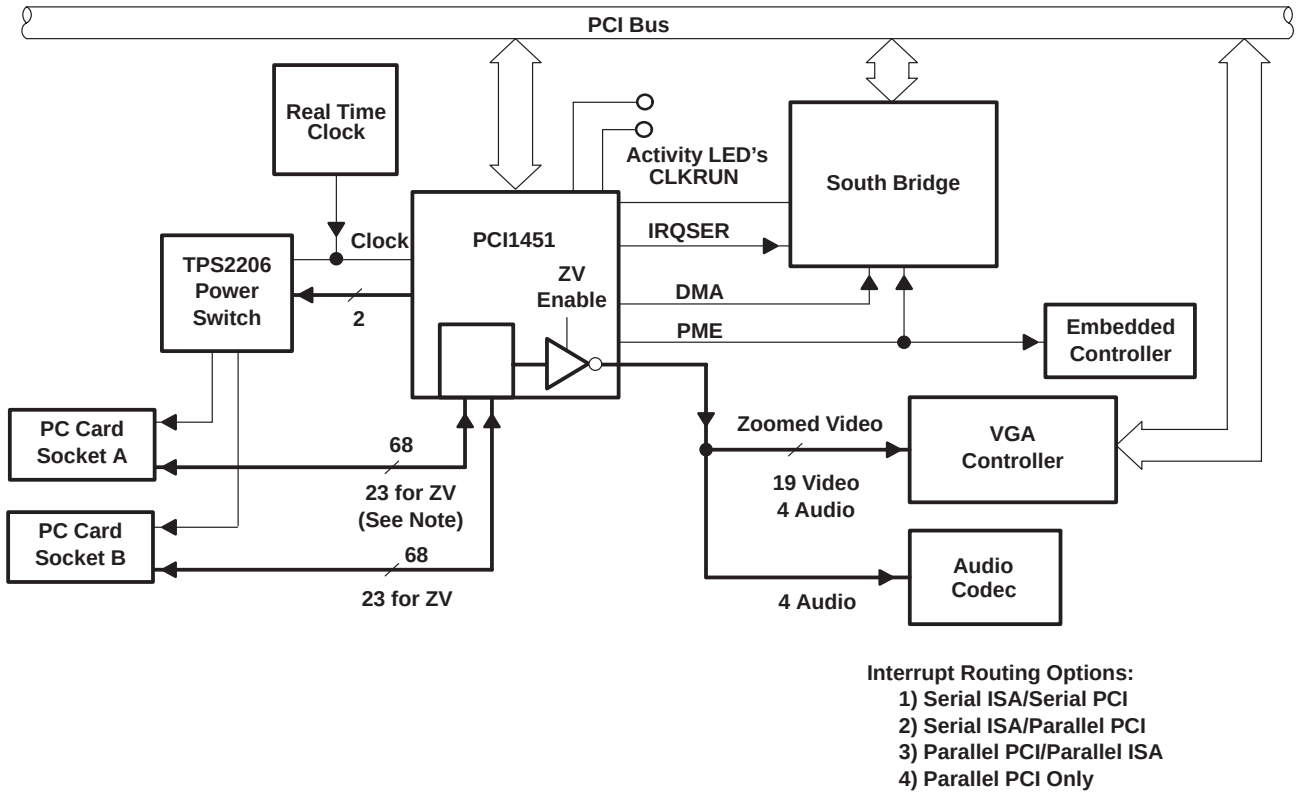
Table 2–17. CardBus PC Card Interface Control (slots A and B)

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

† Terminal name for slot A is preceded with A_. For example, the full name for terminal M1 is A_CAUDIO.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminal C11 is B_CAUDIO.

3 Feature/Protocol Descriptions

Figure 3–1 shows a simplified system implementation example using the PCI1451. The PCI interface includes all address/data and control signals for PCI protocol. Highlighted in this diagram is the functionality supported by the PCI1451. The PCI1451 supports PC/PCI DMA, PCI Way DMA (distributed DMA), $\overline{\text{PME}}$ wake-up from $D3_{cold}$ through D0, 4 interrupt modes, an integrated zoomed video port, and 12 multifunction pins (8 MFUNC and 4 GPIO pins) that can be programmed for a wide variety of functions.



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 3–1. PCI1451 System Block Diagram

3.1 I/O Characteristics

Figure 3–2 shows a 3-state bidirectional buffer. Section 8.2, *Recommended Operating Conditions*, provides the electrical characteristics of the inputs and outputs. The PCI1451 meets the ac specifications of the *1995 PC Card Standard* and the *PCI Local Bus Specification*.

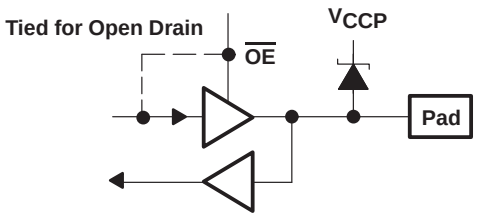


Figure 3–2. 3-State Bidirectional Buffer

3.2 Clamping Voltages

The I/O sites can be pulled through a clamping diode to a voltage rail for protection. The 3.3-V core power supply is independent of the clamping voltages. The clamping (protection) diodes are required if the signaling environment on an I/O is system dependent. For example, PCI signaling can be either 3.3 Vdc or 5 Vdc, and the PCI1451 must reliably accommodate both voltage levels. This is accomplished by using a 3.3-V buffer with a clamping diode to V_{CCP}. If a system design requires a 5-V PCI bus, then the V_{CCP} would be connected to the 5-V power supply.

A standard die has only one clamping voltage for the sites as shown in Figure 3–2. After the terminal assignments are fixed, the fabrication facility will support a design by splitting the clamping voltage for customization. The PCI1451 requires three separate clamping voltages since it supports a wide range of features. The three voltages are listed and defined in Section 8.2, *Recommended Operating Conditions*.

3.3 Peripheral Component Interconnect (PCI) Interface

This section describes the PCI interface of the PCI1451, and how the device responds to and participates in PCI bus cycles. The PCI1451 provides all required signals for PCI master/slave devices and may operate in either 5-V or 3.3-V PCI signaling environments by connecting the V_{CCP} terminals to the desired signaling level.

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

The bus locking protocol defined in the *PCI Local Bus Specification* is not highly recommended, but is provided on the PCI1451 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 PCI1451 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 *PCI Local Bus Specification*, and the issue is resolved by the PCI master using $\overline{\text{LOCK}}$.

3.3.2 Loading The Subsystem Identification (EEPROM Interface)

The subsystem vendor ID register (see Section 4.26) and subsystem ID register (see Section 4.27) 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 PCI1451 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 bit 5 (SUBSYSRW) in the system control register (PCI offset 80h, see Section 4.29). Once this bit is cleared (0), the BIOS may write a subsystem identification value into the registers at PCI 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 PCI1451 loads the doubleword of data from the serial EEPROM after a reset of the primary bus. The $\overline{\text{SUSPEND}}$ input gates the $\overline{\text{PRST}}$ and $\overline{\text{GRST}}$ from the entire PCI1451 core, including the serial EEPROM state machine (see Section 3.6.7, *Suspend Mode*, for details on using $\overline{\text{SUSPEND}}$). The PCI1451 provides a two-line serial bus interface to the serial EEPROM.

The system designer must implement a pullup resistor on the PCI1451 SDA terminal to indicate the serial EEPROM mode. Only when this pullup resistor is present will the PCI1451 attempt to load data through the serial EEPROM interface. Note that a pullup resistor is also required on the SCL terminal to implement the EEPROM interface correctly. The serial EEPROM interface is a two-pin interface with one data signal (SDA) and one clock signal (SCL). Figure 3–3 illustrates a typical PCI1451 application using the serial EEPROM interface.

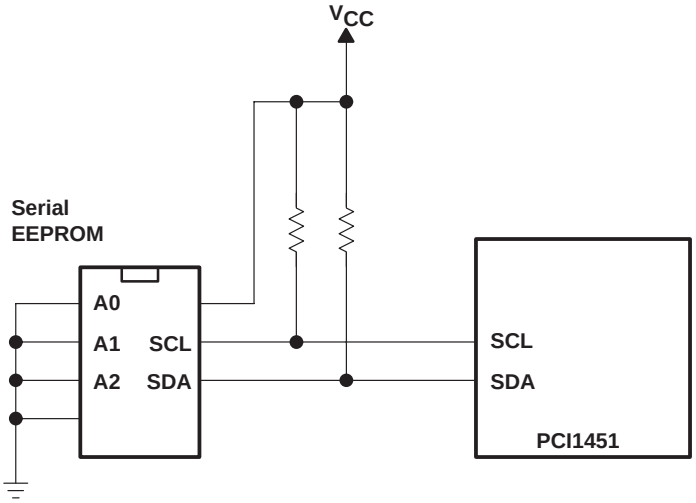


Figure 3–3. Serial EEPROM Application

As stated above, when the PCI1451 is reset by $\overline{\text{GRST}}$, the subsystem data is read automatically from the EEPROM. The PCI1451 masters the serial EEPROM bus and reads four bytes as described in Figure 3–4.

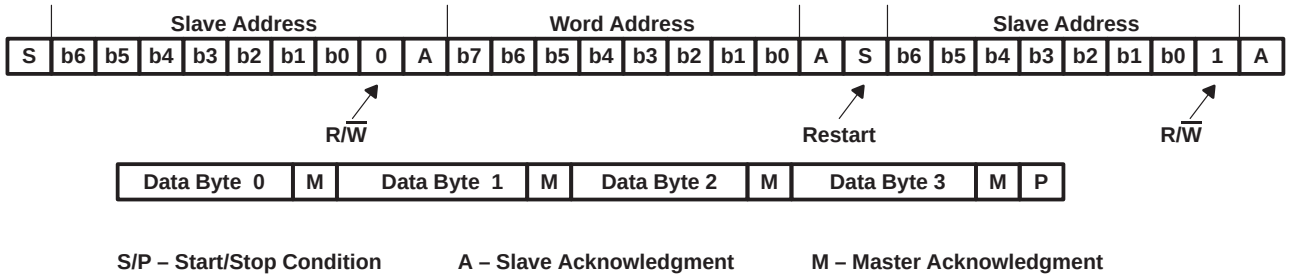


Figure 3–4. EEPROM Interface Subsystem Data Collection

The EEPROM is addressed at word address 00h, as indicated in Figure 3–4, and the address autoincrements 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 PCI1451. 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 3–3, assumes the 1010b high address nibble. The lower three address bits are terminal inputs to the chip, and the sample application shows these terminal inputs tied to GND.

The serial EEPROM interface signals require pullup resistors. The serial EEPROM protocol allows bidirectional transfers. Both the SCL and SDA signals are placed in a high-impedance state and pulled high when the bus is not active. 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 3–5 illustrates this protocol.

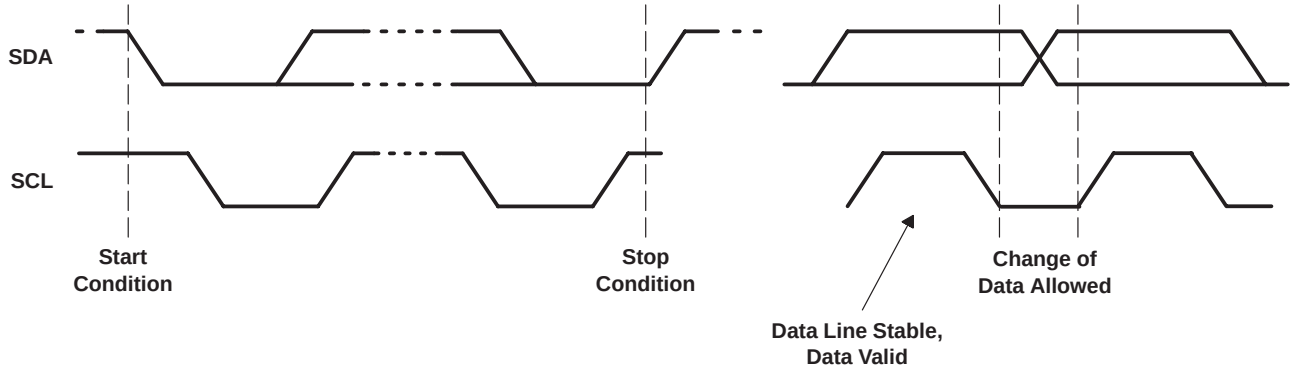


Figure 3–5. Serial EEPROM Start/Stop Conditions and BIt Transfers

Each address byte and data transfer is followed by an acknowledge bit, as indicated in Figure 3–4. When the PCI1451 transmits the addresses, it returns the SDA signal to the high state and places the line in a high-impedance state. The PCI1451 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 PCI1451 transmitter and the EEPROM receiver. Figure 3–6 illustrates general acknowledges.

During the data byte transfers from the serial EEPROM to the PCI1451, the EEPROM clocks the SCL signal. After the EEPROM transmits the data to the PCI1451, it returns the SDA signal to the high state and places the line in a high-impedance state. The EEPROM then generates an SCL clock cycle and expects the PCI1451 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 PCI1451 receiver. Figure 3–6 illustrates general acknowledges.

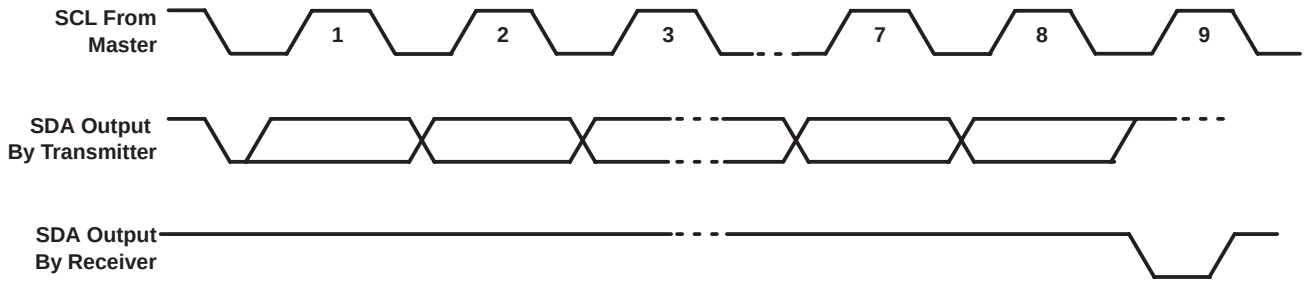


Figure 3–6. Serial EEPROM Protocol – Acknowledge

EEPROM interface status information is communicated through the general status register (PCI offset 85h, see Section 4.31). Bit 2 (EEDETECT) in this register indicates whether or not the PCI1451 serial EEPROM circuitry detects the pullup resistor on SDA. An error condition, such as a missing acknowledge, results in bit 1 (DATAERR) being set. Bit 0 (EEBUSY) is set while the subsystem ID register is loading (serial EEPROM interface is busy).

3.3.3 Serial Bus EEPROM Application

When the PCI bus is reset and the serial bus interface is detected, the PCI1451 attempts to read the subsystem identification and other register defaults from a serial EEPROM. The registers and corresponding bits that may be loaded with defaults through the EEPROM are provided in Table 3–1.

Table 3–1. Registers and Bits Loadable Through Serial EEPROM

PCI OFFSET	EEPROM OFFSET REFERENCE	REGISTER NAME	BITS LOADED FROM EEPROM TO CORRESPONDING BITS IN REGISTER
PCI 43h	21h	Subsystem ID (see Section 4.27)	Byte 1
PCI 42h	22h	Subsystem ID (see Section 4.27)	Byte 0
PCI 41h	23h	Subsystem vendor ID (see Section 4.26)	Byte 1
PCI 40h	24h	Subsystem vendor ID (see Section 4.26)	Byte 0
PCI 80h	25h	System control (see Section 4.29)	Byte 0, bits 6, 5, 4, 3, 1, 0
PCI 81h	26h	System control (see Section 4.29)	Byte 1, bits 7, 6
PCI 82h	27h	System control (see Section 4.29)	Byte 2, bits 6–0
PCI 83h	28h	System control (see Section 4.29)	Byte 3, bits 7, 6, 5, 3, 2, 0
PCI 86h	29h	Reserved	No bits loaded
PCI 89h	2Ah	General-purpose event enable (see Section 4.33)	Bits 7, 6, 3, 2, 1, 0
PCI 8Bh	2Bh	General-purpose output (see Section 4.35)	Bits 3–0
PCI 8Ch	2Ch	Multifunction routing status (see Section 4.36)	Byte 0
PCI 8Dh	2Dh	Multifunction routing status (see Section 4.36)	Byte 1
PCI 8Eh	2Eh	Multifunction routing status (see Section 4.36)	Byte 2
PCI 8Fh	2Fh	Multifunction routing status (see Section 4.36)	Byte 3
PCI 91h	30h	Card control (see Section 4.38)	Bits 7, 2, 1
PCI 92h	31h	Device control (see Section 4.39)	Bits 7–0
PCI 93h	32h	Diagnostic (see Section 4.40)	Bits 7, 4–0
PCI A2h	33h	Power management capabilities (see Section 4.45)	Bit 15
ExCA 00h	34h	ExCA identificaton and revision (see Section 5.1)	Bits 7–0

The EEPROM data format is detailed in Figure 3–7. This format must be followed for the PCI1451 to load initializations properly from a serial EEPROM. Any undefined condition results in a terminated load and sets the DATAERR bit in the general status register (see Section 4.31).

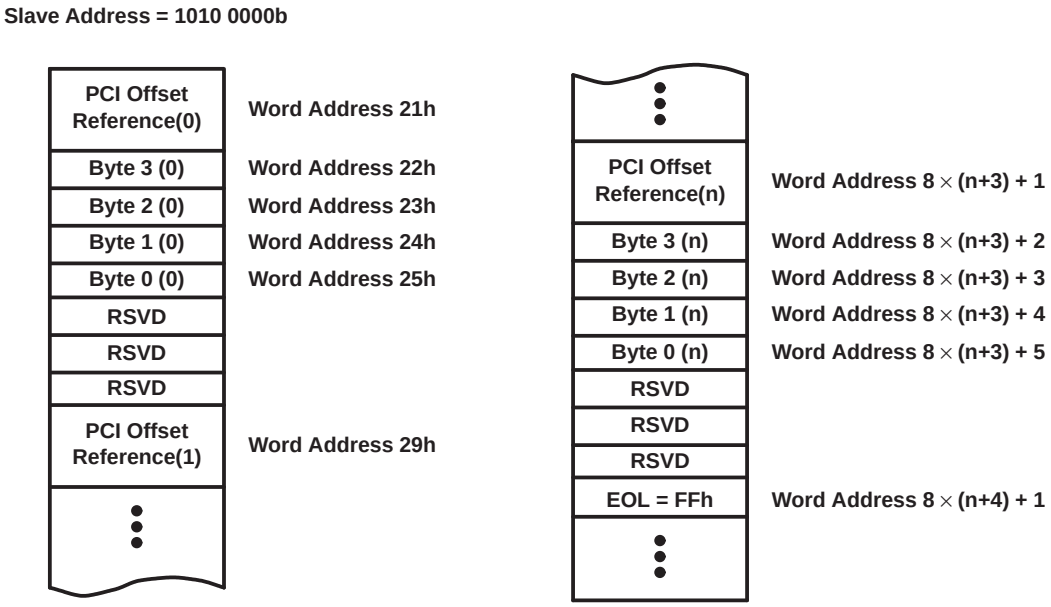


Figure 3–7. EEPROM Data Format

The byte at EEPROM word address 00h must contain either a valid PCI offset, as listed in Table 3–1, or an end-of-list (EOL) indicator. The EOL indicator is a byte value of FFh, and indicates the end of the data to load from the EEPROM. Only doubleword registers are loaded from the EEPROM, and all bit fields must be considered when programming the EEPROM.

The serial EEPROM is addressed at slave address 1010 0000b by the PCI1451. All hardware addres 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 3–3) assumes the 1010b high-address nibble. The lower three address bits are terminal inputs to the chip, and the sample application shows these terminal inputs tied to GND.

When a valid offset reference is read, four bytes are read from the EEPROM, MSB first, as illustrated in Figure 3–4. The address autoincrements after every byte transfer according to the doubleword read protocol. Note that the word addresses align with the data format illustrated in Figure 3–7. The PCI1451 continues to load data from the serial EEPROM until an end-of-list indicator is read. Three reserved bytes are stuffed to maintain eight-byte data structures.

Note that the eight-byte data structure is important to provide correct addressing per the doubleword read format shown in Figure 3–4. In addition, the reference offsets must be loaded in the EEPROM in sequential order, that is, 01h, 02h, 03h, 04h. If the offsets are not sequential, the registers may be loaded incorrectly.

3.4 PC Card Applications Overview

This section describes the PC Card interfaces of the PCI1451. A discussion on PC Card recognition details the card interrogation procedure. The card powering procedure is also discussed, including the protocol of the P^2C power switch interface. The internal ZV buffering provided by the PCI1451 and programming model is also detailed. Also, standard PC Card register models are described, as well as a brief discussion of the PC Card software protocol layers.

3.4.1 PC Card Insertion/Removal and Recognition

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

The scheme uses the $\overline{\text{CD1}}$, $\overline{\text{CD2}}$, $\overline{\text{VS1}}$, and $\overline{\text{VS2}}$ signals ($\overline{\text{CCD1}}$, $\overline{\text{CCD2}}$, CVS1, CVS2 for CardBus). A PC Card designer connects these four pins in a certain configuration depending on the type of card and the supply voltage. The encoding scheme for this, defined in the *1997 PC Card Standard*, is shown in Table 3–2.

Table 3–2. PC Card – Card Detect and Voltage Sense Connections

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

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

A power switch with a PCMCIA-to-peripheral control (P^{2}C) interface is required for the PC Card powering interface. The TI TPS2206 (or TPS2202A) Dual-Slot PC Card Power-Interface Switch provides the P^{2}C interface to the CLOCK, DATA, and LATCH terminals of the PCI1451. Figure 3–8 shows the terminal assignments of the TPS2206. Figure 3–9 illustrates a typical application where the PCI1451 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 PCI1451 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 (P2CCLK) of the system control register (PCI offset 80h). This bit is encoded as follows:

0 = CLOCK terminal 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.

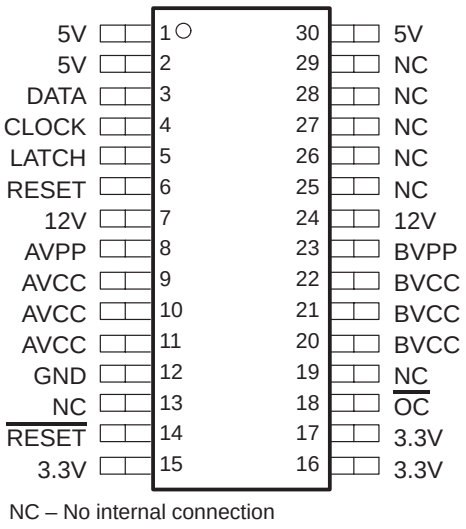


Figure 3–8. TPS2206 Terminal Assignments

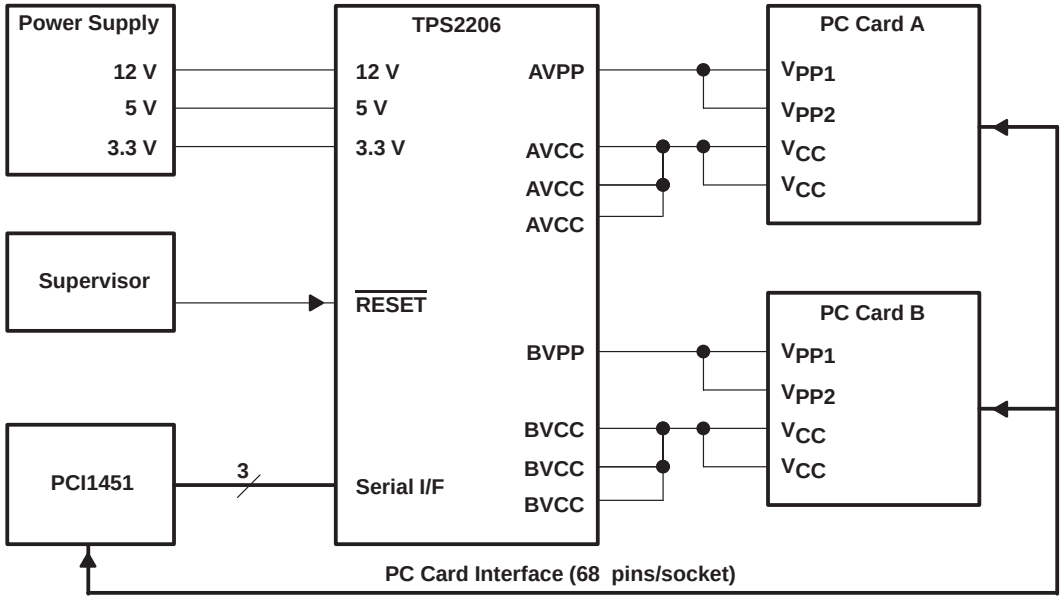
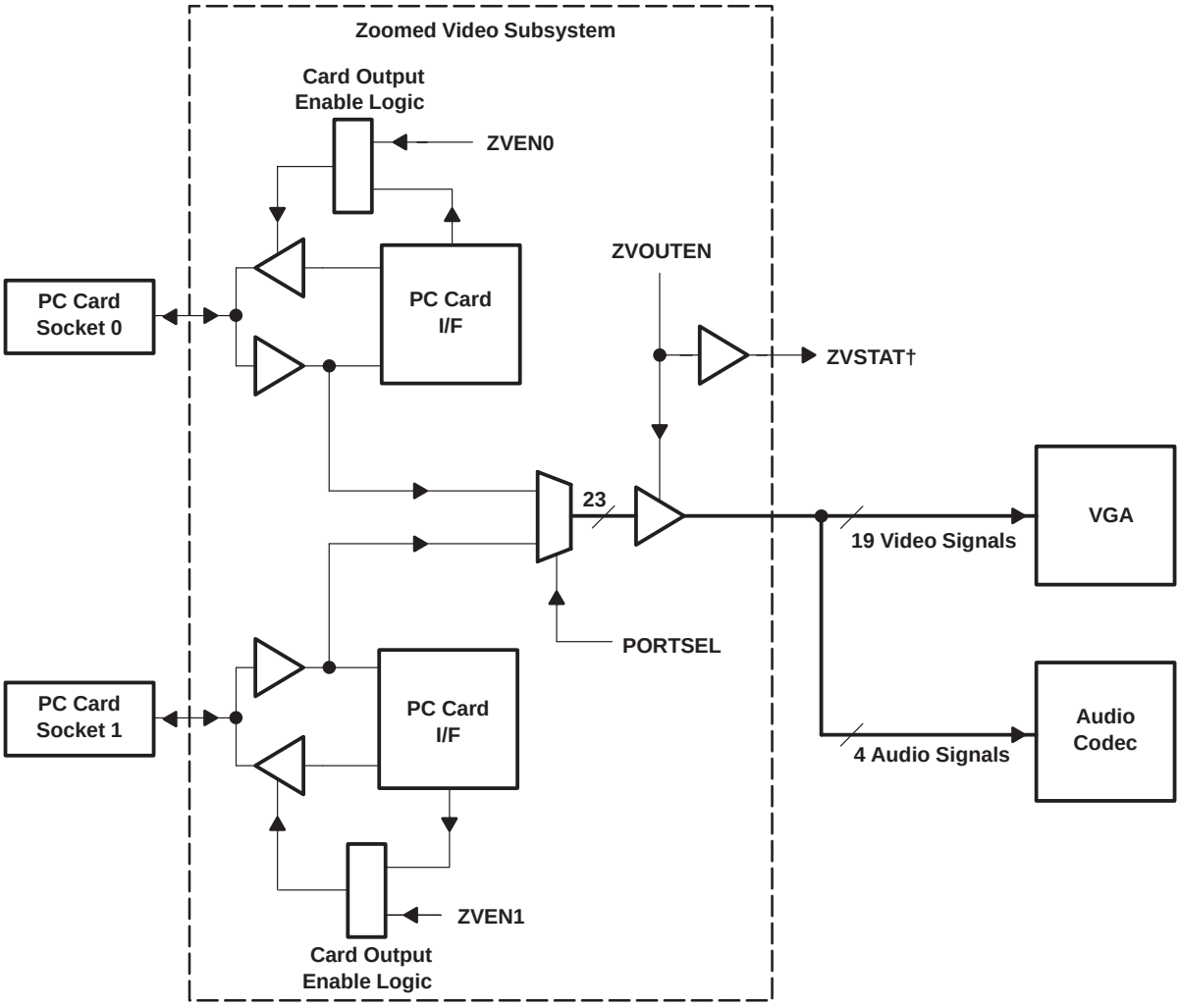


Figure 3–9. TPS2206 Typical Application

3.4.3 Zoomed Video Support

The zoomed video (ZV) port on the PCI1451 provides an internally buffered 16-bit ZV PC Card data path. This internal routing is programmed through the multimedia control register. Figure 3–9 summarizes the zoomed video subsystem implemented in the PCI1451, and details the bit functions found in the multimedia control register.

An output port (PORTSEL) is always selected. The PCI1451 defaults to socket 0 (see the multimedia control register). When ZVOUTEN is enabled, the zoomed video output terminals are enabled and allow the PCI1451 to route the zoomed video data. However, no data is transmitted unless either bit 0 (ZVEN0) or bit 1 (ZVEN1) is enabled in the multimedia control register. If the PORTSEL maps to a card port that is disabled (ZVEN =0 or ZVEN1 = 0), then the zoomed video port is driven low (i.e., no data is transmitted).



† ZVSTAT must be enabled through the GPIO Control Register.

Figure 3–10. Zoomed Video Subsystem

3.4.4 Zoomed Video Auto Detect

Zoomed video auto detect, when enabled, allows the PCI1451 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 PCI1451 automatically switches the internal zoomed video MUX to route the zoomed video stream to the PCI1451'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 PCI1451 can be programmed to switch a third zoomed video source by programming MFUNC2 or MFUNC3 as a zoomed video pixel clock sense pin and connecting this pin to the pixel clock of the third zoomed video source. ZVSTAT may then be programmed onto MFUNC4, MFUNC1, or MFUNC0 and this signal may switch the zoomed video buffers from the third zoomed video source. To account for the possibility of several zoomed video sources being enabled at the same time, a programmable priority scheme may be enabled.

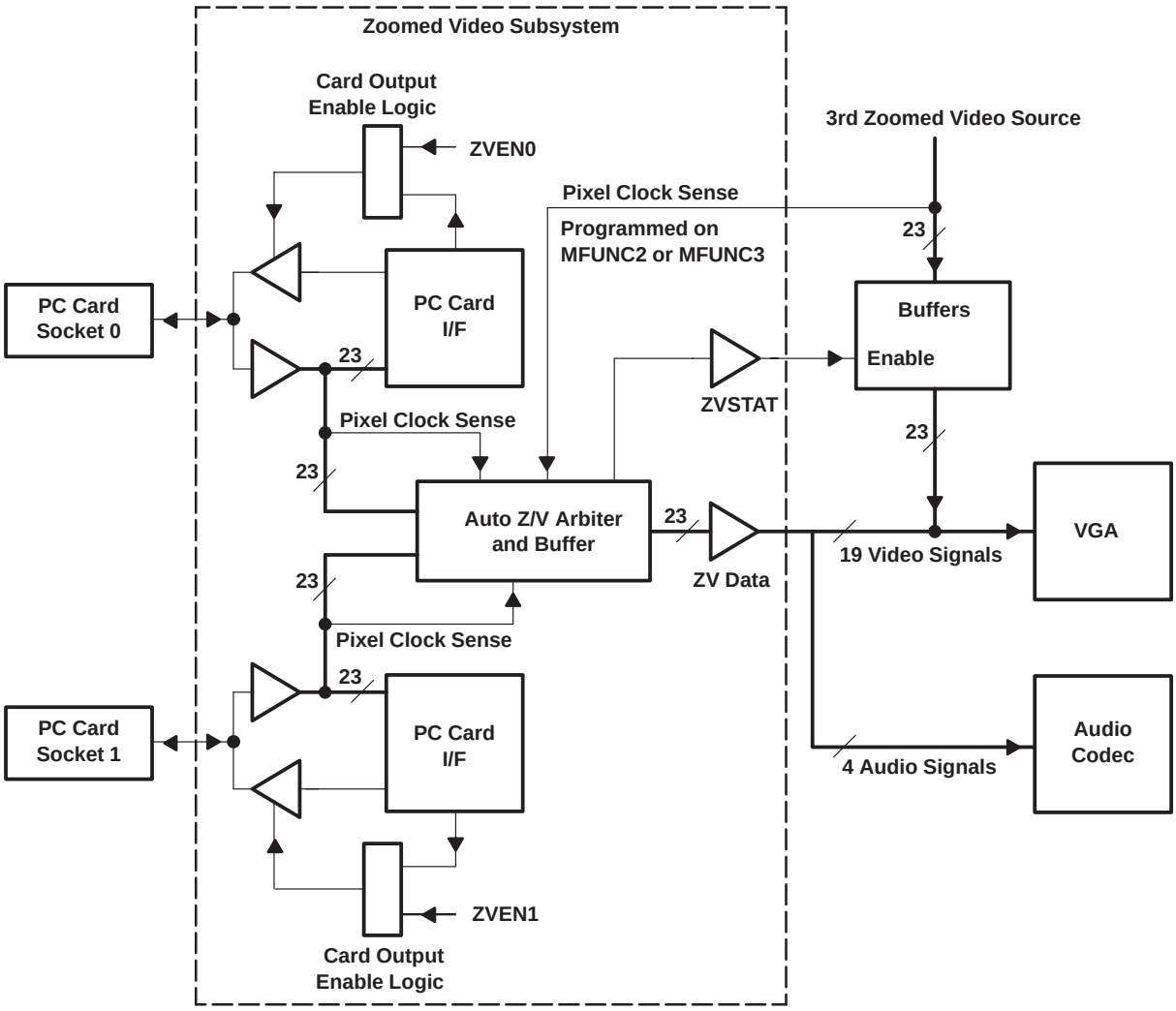


Figure 3–11. Zoomed Video With Auto Detect Enabled

The PCI1451 defaults with zoomed video auto-detect disabled so that it will function exactly like the PCI1250A and PCI1451. To enable zoomed video auto-detect and the programmable priority scheme, the following bits must be set:

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

  000 = Slot A, Slot B, External Source

  001 = Slot A, External Source, Slot B

  010 = Slot B, Slot A, External Source

  011 = Slot B, External Source, Slot A

  100 = External Source, Slot A, Slot B

  101 = External Source, Slot B, Slot A

  110 = External Source, Slot B, Slot A

  111 = Reserved

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

- MFUNC routing register (PCI offset 8Ch), bits 14–12 or 10–8: Write 111b to program MFUNC3 or MFUNC2 as a pixel clock input pin.
- MFUNC routing register (PCI offset 8Ch), bits 18–16, 6–4, or 2–0: Write 111b to program MFUNC4, MFUNC1, or MFUNC0 pin.

3.4.5 Ultra Zoomed Video

Ultra zoomed video is an enhancement to the PCI1451'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 1451 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 1451 prefetches an extra 16 bits (32 bits total) during each PCI read transaction. If the PCI Bus becomes busy, then the 1451 has an extra 16 bits of data to perform back-to-back 16-bit transactions to the PC Card before having to fetch more data. This feature is built into the DMA engine and software is not required to enable this enhancement.

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

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

Additional functionality added for the 1451 versus the 1250A/1251 series is the $\overline{\text{D3\_STAT}}$ (D3 status) terminal. This terminal is asserted under the following two conditions (both conditions must be true before $\overline{\text{D3\_STAT}}$ is asserted):

- Function 0 and Function 1 are placed in D3
- $\overline{\text{PME}}$ is enabled on either function

The intent of including this feature in the PCI1451 is to use this pin to switch an external V_{CC}/V_{AUX} switch. This feature can be programmed on MFUNC7, MFUNC6, MFUNC2, or MFUNC1 by writing 100b to the appropriate multifunction routing status register bits (PCI offset 8Ch).

3.4.7 Internal Ring Oscillator

The internal ring oscillator provides an internal clock source for the PCI1451 so that neither the PCI clock nor an external clock is required in order for the PCI1451 to power down a socket or interrogate a PC Card. This internal oscillator operates nominally at 16 kHz and can be enabled by setting bit 27 (P2CCLK) of the system control register (PCI offset 80h) to a 1b. This function is disabled by default.

3.4.8 Integrated Pullup Resistors

The *1997 PC Card Standard* requires pullup resistors on various terminals to support both CardBus and 16-bit card configurations. Unlike the PCI1450/4450 which required external pullup resistors, the PCI1451 has integrated all of these pullup resistors, except for the WP(IOIS16)/CLKRUN pullup resistor.

SIGNAL NAME	GJG PIN NUMBER	
	SOCKET A	SOCKET B
ADDR14/CPERR	D1	C18
READY/CINT	K2	E11
ADDR15/CIRDY	F4	B16
CD1/CCD1	F8	J19
VS1/CVS1	K1	B11
ADDR19/CBLOCK	C2	D16
ADDR20/CSTOP	D2	B19
ADDR21/CDEVSEL	E5	A18
ADDR22/CTRDY	F5	B17
VS2/CVS2	H6	A14
RESET/CRST	H2	F13
WAIT/CSERR	K4	F11
INPACK/CREQ	J4	A13
BVD2(SPKR)/CAUDIO	L6	F10
BVD1(STSCHG)/CSTSCHG	L1	A10
CD2/CCD2	L4	D10
WP(IOIS16)/CLKRUN	L2†	B10†

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

3.4.9 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 SPKR. This terminal, also used in CardBus applications, is referred to as CAUDIO. SPKR passes a TTL level digital audio signal to the PCI1451. 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 PCI1451 to produce SPKROUT. Figure 3–12 illustrates the SPKROUT connection.

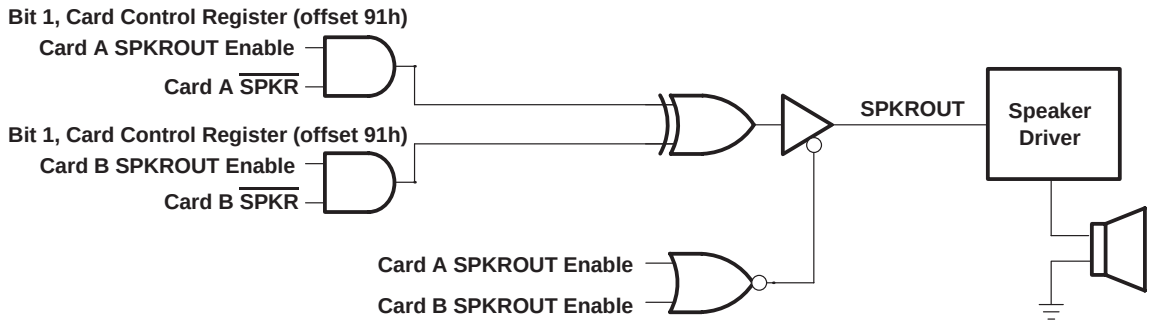


Figure 3–12. SPKROUT Connection to Speaker Driver

The SPKROUT signal is typically driven only by PC modem cards. To verify the SPKROUT on the PCI1451, a sample circuit was constructed, and this simplified schematic is provided below. The PCI1130/1131 required a pullup resistor on the SUSPEND/SPKROUT terminal. Since the PCI1451 does not multiplex any other function on SPKROUT, this terminal does not require a pullup resistor.

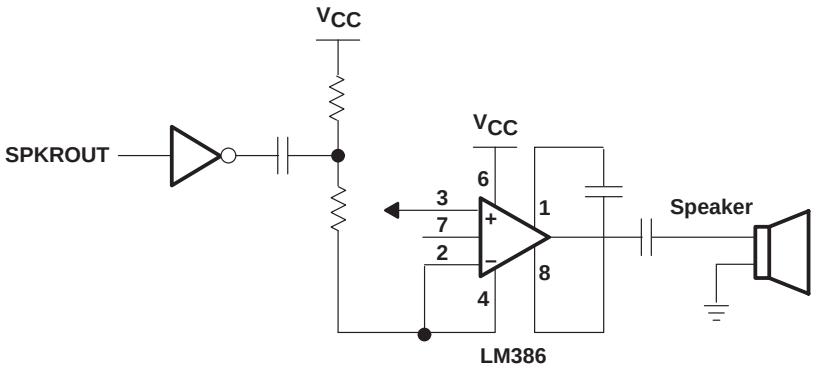


Figure 3–13. Simplified Test Schematic

3.4.10 LED Socket Activity Indicators

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

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

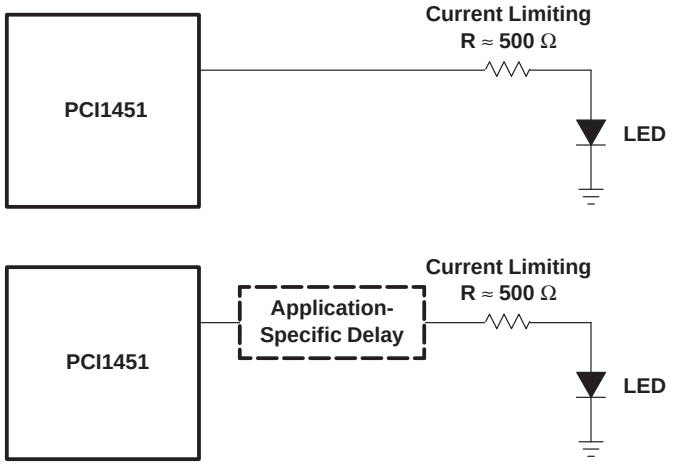


Figure 3–14. Two Sample LED Circuits

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

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

3.4.11 PC Card 16 DMA Support

The PCI1451 supports both PC/PCI (centralized) DMA and a distributed DMA slave engine for 16-bit PC Card DMA support. The distributed DMA (DDMA) slave register set provides the programmability necessary for the slave DDMA engine. Table 3–3 provides the DDMA register configuration.

Table 3–3. Distributed DMA Registers

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

3.4.12 CardBus Socket Registers

The PCI1451 contains all registers for compatibility with the *1997 PC Card Standard*. These registers exist as the CardBus socket registers, and are listed in Table 3–4.

Table 3–4. CardBus Socket Registers

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

3.5 Programmable Interrupt Subsystem

Interrupts provide a way for I/O devices to let the microprocessor know that they require servicing. The dynamic nature of PC Cards, and the abundance of PC Card I/O applications require substantial interrupt support from the PCI1451. The PCI1451 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 PCI1451 is therefore backward compatible with existing interrupt control register definitions, and new registers have been defined where required.

The PCI1451 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 PCI1451, PC Card interrupts are classified as either card status change (CSC) or as functional interrupts.

The method by which any type of PCI1451 interrupt is communicated to the host interrupt controller varies from system to system. The PCI1451 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.

3.5.1 PC Card Functional And Card Status Change Interrupts

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

Card status change (CSC) type interrupts, defined as events at the PC Card interface which are detected by the PCI1451, may warrant notification of host card and socket services software for service. CSC events include both card insertion and removal from PC Card sockets, as well as transitions of certain PC Card signals.

Table 3–5 summarizes the sources of PC Card interrupts and the type of card associated with them. CSC and functional interrupt sources are dependent upon the type of card inserted in the PC Card socket. The three types of cards that may be inserted into any PC Card socket are: 16-bit memory card, 16-bit I/O card, and CardBus cards. Functional interrupt events are valid only for 16-bit I/O and CardBus cards, that is, the functional interrupts are not valid for 16-bit memory cards. Furthermore, card insertion and removal type CSC interrupts are independent of the card type.

Table 3–5. 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 PCI1451 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 PCI1451 interrupt scheme may be used to notify the host system, as in indicated in Table 3–5, denoted by the power cycle complete event. This interrupt source is considered a PCI1451 internal event because it does not depend on a signal change at the PC Card interface, but rather the completion of applying power to the socket.

3.5.2 Interrupt Masks And Flags

Host software may individually mask, or disable, most of the potential interrupt sources listed in Table 3–6 by setting the appropriate bits in the PCI1451. By individually masking the interrupt sources listed in these tables, software can control which events will cause a PCI1451 interrupt. Host software has some control over which system interrupt the PCI1451 will assert by programming the appropriate routing registers. The PCI1451 allows host software to route PC Card CSC and PC Card functional interrupts to separate system interrupts. Interrupt routing is somewhat specific to the interrupt signaling method used. This will be discussed in more detail in the following sections.

When an interrupt is signaled by the PCI1451, the interrupt service routine must be able to discern which of the events in Table 3–6 caused the interrupt. Internal registers in the PCI1451 provide flags which report which of the interrupt sources was the cause of an interrupt. By reading these status bits, the interrupt service routine can determine which action is to be taken.

Table 3–6 details the registers and bits associated with masking and reporting potential interrupts. All interrupts may be masked except the functional PC Card interrupts, and an interrupt status flag is available for all types of interrupts.

Table 3–6. PCI1451 Interrupt Masks and Flags Registers

Card Type	Event	Mask	Flag
16-bit Memory	Battery conditions (BVD1, BVD2)	ExCA Offset 05h/45h/805h Bits 1 and 0	ExCA Offset 04h/44h/804h Bits 1 and 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 and 1	Socket event register Bits 2 and 1

There is no mask bit to stop the PCI1451 from passing PC Card functional interrupts through to the appropriate interrupt scheme. Functional interrupts should not be fired until the PC Card is initialized and powered.

There are various methods of clearing the interrupt flag bits listed in Table 3–6. The flag bits in the ExCA registers (16-bit PC Card related interrupt flags) may be cleared by two different methods. One method is an explicit write of 1 to the flag bit to clear, and the other is a reading of the flag bit register. The selection of flag bit clearing is made by bit 2 in the global control register (ExCA offset 1Eh/5Eh/81Eh), and defaults to the flag cleared on read method.

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

3.5.3 Using Parallel PCI Interrupts

Parallel PCI interrupts are available when in pure parallel PCI interrupt mode and are routed on MFUNC0–MFUNC2. The PCI interrupt signaling is dependent upon the interrupt mode and is summarized in Table 3–7. The interrupt mode is selected in the device control register (92h).

Table 3–7. Interrupt Pin Register Cross Reference

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

3.6 Power Management Overview

In addition to the low-power CMOS technology process used for the PCI1451, various features are designed into the device to allow implementation of popular power saving techniques. These features and techniques are discussed in this section.

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

$\overline{\text{CLKRUN}}$ is the primary method of power management on the PCI bus side of the PCI1451. Since some chipsets do not implement $\overline{\text{CLKRUN}}$, this is not always available to the system designer, and alternate power savings features are provided.

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

3.6.2 CardBus PC Card Power Management

The PCI1451 implements its own card power management engine that can turn off the CCLK to a socket when there is no activity to the CardBus PC Card. The CCLK can also be configured as divide by 16 instead of stopped. The $\overline{\text{CLKRUN}}$ protocol is followed on the CardBus interface to control this clock management.

3.6.3 PCI Bus Power Management

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

The four power management states of PCI functions are: D0 - Fully On state, D1 and D2 - intermediate states, and D3 - Off state. Similarly, bus power states of the PCI bus are B0–B3. The bus power states B0–B3 are derived from the device power state of the upstream bridge device.

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

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

Power Management Register Block

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

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

3.6.4 CardBus Device Class Power Management

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

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

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

The Texas Instruments PCI1451 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 PCI1451, another reset pin is defined: $\overline{GRST}$ (global reset). $\overline{GRST}$ 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 $\overline{GRST}$, 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 Section 3.6.5.
- 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 PCI1451 V_{CC} pins. This switch should be a *make before break* type of switch, so that V_{CC} to the PCI1451 is not interrupted.

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

$\overline{PME}$ context bit means that the bit is cleared only by the assertion of $\overline{GRST}$ 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 $\overline{GRST}$ is asserted.

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

Global reset only bits:

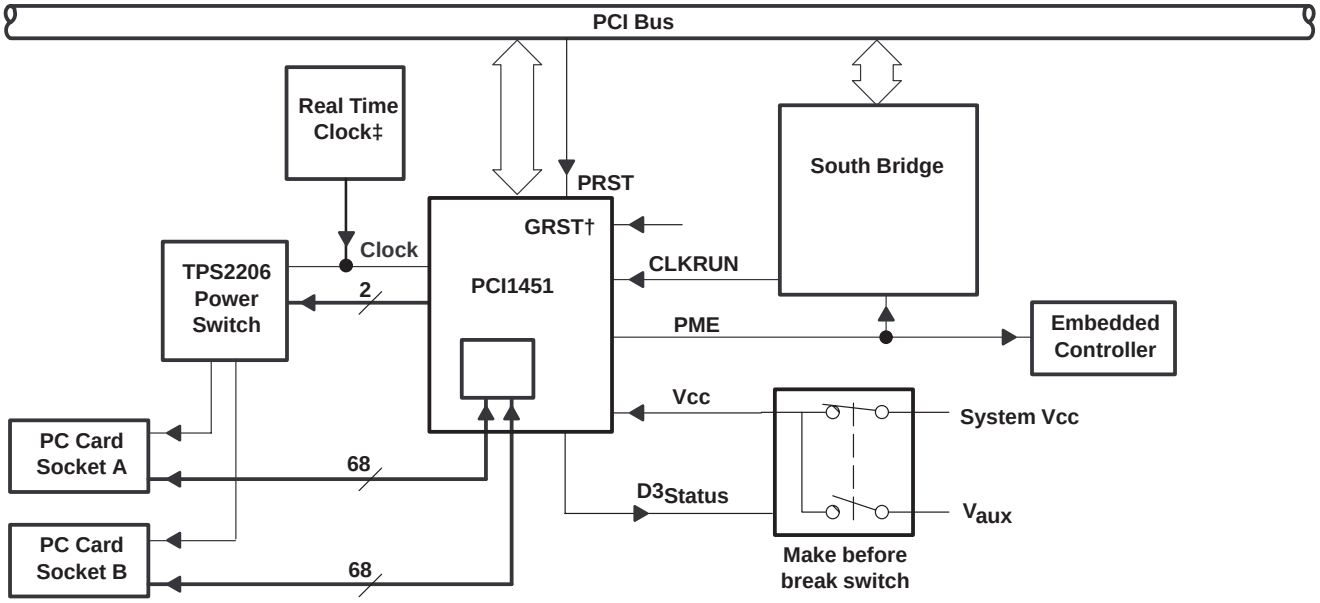
- Subsystem ID/subsystem vendor ID (PCI offset 40h): bits 31–0
- PC Card 16-bit legacy mode base address register (PCI offset 44h): bits 31–1
- System control register (PCI offset 80h): bits 31–29, 27–24, 22–14, 6–3, 1, 0
- Multimedia control register (PCI offset 84h): bits 7–0
- General status register (PCI offset 85h): bits 2–0
- General-purpose event status register (PCI offset 88h): bits 7, 6, 3–0
- General-purpose event enable register (PCI offset 89h): bits 7, 6, 3–0
- General-purpose input register (PCI offset 8Ah): bits 3–0
- General-purpose output register (PCI offset 8Bh): bits 3–0
- MFUNC routing register (PCI offset 8Ch): bits 31–0
- Retry status register (PCI offset 90h): bits 7–1

- Card control register (PCI offset 91h): bits 7, 6, 2, 1, 0
- Device control register (PCI offset 92h): bits 7–0
- Diagnostic register (PCI offset 93h): bits 7–0
- Socket DMA register 0 (PCI offset 94h): bits 1–0
- Socket DMA register 1 (PCI offset 98h): bits 15–0
- $\overline{\text{GPE}}$ control/status register (PCI offset A8h): bits 10, 9, 8, 2, 1, 0

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

- Bridge control register (PCI offset 3Eh): bit 6
- Power management capabilities register (PCI offset A2h): bit 15
- Power management control/status register (PCI offset A4h): bits 15, 8
- ExCA power control register (ExCA 802h/842h): bits 7, 4, 3, 1, 0
- ExCA interrupt and general control (ExCA 803h/843h): bit 6, 5
- ExCA card status change register (ExCA 804h/844h): bits 3–0
- ExCA card status change interrupt register (ExCA 805h/845h): bits 3–0
- CardBus socket event register (CardBus offset 00h): bits 3–0
- CardBus socket mask register (CardBus offset 04h): bits 3–0
- CardBus socket control register (CardBus offset 10h): bits 6, 5, 4, 2, 1, 0

3.6.6 System Diagram Implementing CardBus Device Class Power Management



† The system connection to $\overline{\text{GRST}}$ is implementation specific. $\overline{\text{GRST}}$ should be applied whenever V_{CC} is applied to the PCI1451. $\overline{\text{PRST}}$ should be applied for subsequent warm resets.

‡ Not required if internal oscillator is used.

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

3.6.7 Suspend Mode

The $\overline{\text{SUSPEND}}$ signal, provided for backward compatibility, gates the $\overline{\text{PRST}}$ (PCI reset) signal and the $\overline{\text{GRST}}$ (global reset) signal from the PCI1451. Besides gating $\overline{\text{PRST}}$ and $\overline{\text{GRST}}$, $\overline{\text{SUSPEND}}$ also gates PCLK inside the PCI1451 in order to minimize power consumption.

Gating PCLK does not create any issues with respect to the power switch interface in the PCI1451. This is because the PCI1451 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 PCI1451:

- Use an external clock to the PCI1451 CLOCK pin
- Use the internal oscillator

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

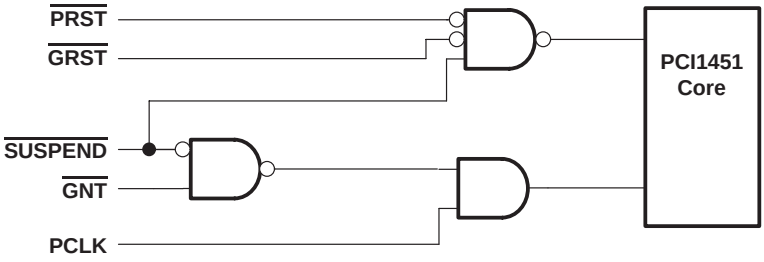


Figure 3–16. $\overline{\text{SUSPEND}}$ Functional Illustration

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

A requirement for implementing suspend mode is that the PCI bus must not be parked on the PCI1451 when $\overline{\text{SUSPEND}}$ is asserted. The PCI1451 responds to $\overline{\text{SUSPEND}}$ being asserted by placing the $\overline{\text{REQ}}$ pin in a high impedance state. The PCI1451 will also gate the internal clock and reset.

The GPIOs, MFUNC signals, and $\overline{\text{RI\_OUT}}$ signals are all active during $\overline{\text{SUSPEND}}$, unless they are disabled in the appropriate PCI1451 registers.

3.6.9 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 PCI1451 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 PCI1451 $\overline{\text{RI\_OUT}}$ function; however, it does not illustrate the masking of CSC events. See *interrupt masks and flags* for a detailed description of CSC interrupt masks and flags.

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

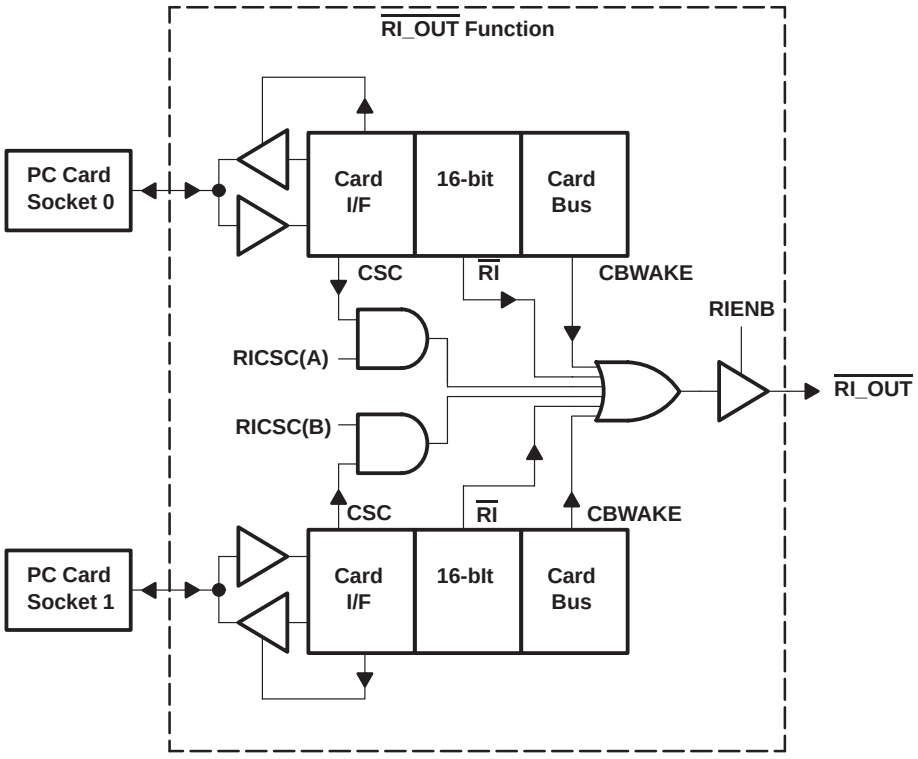


Figure 3–17. $\overline{\text{RI\_OUT}}$ Functional Illustration

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

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

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

4 PC Card Controller Programming Model

This chapter describes the PCI1451 PCI configuration registers that make up the 256-byte PCI configuration header for each PCI1451 function. As noted below, some bits are global in nature and should be accessed only through function 0.

4.1 PCI Configuration Registers (Functions 0 and 1)

The PCI1451 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 Local Bus Specification* as a CardBus bridge header, is *PC 98/PC 99* compliant as well. Table 4–1 illustrates the PCI configuration header, which includes both the predefined portion of the configuration space and the user definable registers.

Table 4–1. Functions 0 and 1 PCI Configuration Register Map

REGISTER NAME				OFFSET
Device ID		Vendor ID		00h
Status		Command		04h
Class code			Revision ID	08h
BIST	Header type	Latency timer	Cache line size	0Ch
CardBus socket/ExCA base address				10h
Secondary status		Reserved	Capability pointer	14h
CardBus latency timer	Subordinate bus number	CardBus bus number	PCI bus number	18h
CardBus memory base register 0				1Ch
CardBus memory limit register 0				20h
CardBus memory base register 1				24h
CardBus memory limit register 1				28h
CardBus I/O base register 0				2Ch
CardBus I/O limit register 0				30h
CardBus I/O base register 1				34h
CardBus I/O limit register 1				38h
Bridge control		Interrupt pin	Interrupt line	3Ch
Subsystem ID		Subsystem vendor ID		40h
PC Card 16-bit I/F legacy mode base address				44h
Reserved				48h–7Fh
System control				80h
Reserved	Reserved	General status	Multimedia control	84h
General-purpose output	General-purpose input	General-purpose event enable	General-purpose event status	88h
Multifunction routing status				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

4.2 Vendor ID Register

The vendor ID register contains a value allocated by the PCI SIG that identifies the manufacturer of the PCI device. The vendor ID assigned to Texas Instruments is 104Ch.

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

Register: **Vendor ID**
Type: Read-only
Offset: 00h (Functions 0, 1)
Default: 104Ch

4.3 Device ID Register

The device ID register contains a value assigned to the PCI1451 by Texas Instruments. The device identification for the PCI1451 is AC52h.

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

Register: **Device ID**
Type: Read-only
Offset: 02h (Functions 0, 1)
Default: AC52h

4.4 Command Register

The command register provides control over the PCI1451 interface to the PCI bus. All bit functions adhere to the definitions in the *PCI Local Bus Specification*, see Table 4–2. None of the bit functions in this register are shared between the two PCI1451 PCI functions. Two command registers exist in the PCI1451, one for each function. Software manipulates the two PCI1451 functions as separate entities when enabling functionality through the command register. The SERR_EN (bit 8) and PERR_EN (bit 6) enable bits in this register are internally wired OR between the two functions, and these control bits appear separate per function to software.

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

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

Table 4–2. PCI Command Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–10	RSVD	R	Reserved. Bits 15–10 return 0s when read.
9	FBB_EN	R	Fast back-to-back enable. The PCI1451 does not generate fast back-to-back transactions; therefore, bit 9 returns 0 when read.
8	SERR_EN	R/W	System error ($\overline{\text{SERR}}$) enable. Bit 8 controls the enable for the $\overline{\text{SERR}}$ driver on the PCI interface. $\overline{\text{SERR}}$ can be asserted after detecting an address parity error on the PCI bus. Both bits 8 and 6 must be set for the PCI1451 to report address parity errors. 0 = Disables the $\overline{\text{SERR}}$ output driver (default). 1 = Enables the $\overline{\text{SERR}}$ output driver.
7	STEP_EN	R	Address/data stepping control. The PCI1451 does not support address/data stepping; therefore, bit 7 is hardwired to 0.
6	PERR_EN	R/W	Parity error response enable. Bit 6 controls the PCI1451's response to parity errors through $\overline{\text{PERR}}$. Data parity errors are indicated by asserting $\overline{\text{PERR}}$, while address parity errors are indicated by asserting $\overline{\text{SERR}}$. 0 = PCI1451 ignores detected parity error (default). 1 = PCI1451 responds to detected parity errors.
5	VGA_EN	R/W	VGA palette snoop. When bit 5 is set to 1, palette snooping is enabled (that is, the PCI1451 does not respond to palette register writes and snoops the data). When this bit is 0, the PCI1451 treats all palette accesses like all other accesses.
4	MWI_EN	R	Memory write and invalidate enable. Bit 4 controls whether a PCI initiator device can generate memory write and invalidate commands. The PCI1451 controller does not support memory write and invalidate commands, it uses memory write commands instead; therefore, this bit is hardwired to 0.
3	SPECIAL	R	Special cycles. Bit 3 controls whether or not a PCI device ignores PCI special cycles. The PCI1451 does not respond to special cycle operations; therefore, this bit is hardwired to 0.
2	MAST_EN	R/W	Bus master control. Bit 2 controls whether or not the PCI1451 can act as a PCI bus initiator (master). The PCI1451 can take control of the PCI bus only when this bit is set. 0 = Disables the PCI1451's ability to generate PCI bus accesses (default). 1 = Enables the PCI1451's ability to generate PCI bus accesses.
1	MEM_EN	R/W	Memory space enable. Bit 1 controls whether or not the PCI1451 may claim cycles in PCI memory space. 0 = Disables the PCI1451's response to memory space accesses (default). 1 = Enables the PCI1451's response to memory space accesses.
0	IO_EN	R/W	I/O space control. Bit 0 controls whether or not the PCI1451 may claim cycles in PCI I/O space. 0 = Disables the PCI1451 from responding to I/O space accesses (default). 1 = Enables the PCI1451 to respond to I/O space accesses.

4.5 Status Register

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

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

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

Table 4–3. Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	PAR_ERR	R/C	Detected parity error. Bit 15 is set when a parity error (either address or data) is detected. Write a 1 to clear this bit.
14	SYS_ERR	R/C	Signaled system error. Bit 14 is set when $\overline{\text{SERR}}$ is enabled and the PCI1451 signals a system error to the host. Write a 1 to clear this bit.
13	MABORT	R/C	Received master abort. Bit 13 is set when a cycle initiated by the PCI1451 on the PCI bus has been terminated by a master abort. Write a 1 to clear this bit.
12	TABT_REC	R/C	Received target abort. Bit 12 is set when a cycle initiated by the PCI1451 on the PCI bus was terminated by a target abort. Write a 1 to clear this bit.
11	TABT_SIG	R/C	Signaled target abort. Bit 11 is set by the PCI1451 when it terminates a transaction on the PCI bus with a target abort. Write a 1 to clear this bit.
10–9	PCI_SPEED	R	$\overline{\text{DEVSEL}}$ timing. Bits 10 and 9 encode the timing of $\overline{\text{DEVSEL}}$ and are hardwired to 01b indicating that the PCI1451 asserts this signal at a medium speed on nonconfiguration cycle accesses.
8	DATAPAR	R/C	Data parity error detected. Write a 1 to clear this bit. 0 = The conditions for setting this bit have not been met. 1 = A $\overline{\text{data parity}}$ error occurred and the following conditions were met: a. $\overline{\text{PERR}}$ was asserted by any PCI device including the PCI1451. b. The PCI1451 was the bus master during the data parity error. c. Bit 6 (PERR_EN) is set in the command register (PCI offset 04h, see Section 4.4).
7	FBB_CAP	R	Fast back-to-back capable. The PCI1451 cannot accept fast back-to-back transactions; therefore, bit 7 is hardwired to 0.
6	UDF	R	User-definable feature support. The PCI1451 does not support the user definable features; therefore, bit 6 is hardwired to 0.
5	66MHZ	R	66-MHz capable. The PCI1451 operates at a maximum PCLK frequency of 33 MHz; therefore, bit 5 is hardwired to 0.
4	CAPLIST	R	Capabilities list. Bit 4 returns 1 when read. This bit indicates that capabilities in addition to standard PCI capabilities are implemented. The linked list of PCI power management capabilities is implemented in this function.
3–0	RSVD	R	Reserved. Bits 3–0 return 0s when read.

4.6 Revision ID Register

The revision ID register indicates the slicon revision of the PCI1451.

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

4.7 PCI Class Code Register

The PCI class code register recognizes the PCI1451 functions 0 and 1 as a bridge device (06h) and CardBus bridge device (07h) with a 00h programming interface.

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

4.8 Cache Line Size Register

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

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

Register: **Cache line size**
Type: Read/Write
Offset: 0Ch (Functions 0, 1)
Default: 00h

4.9 Latency Timer Register

The latency timer register specifies the latency timer for the PCI1451, in units of PCI clock cycles. When the PCI1451 is a PCI bus initiator and asserts $\overline{\text{FRAME}}$, the latency timer begins counting from zero. If the latency timer expires before the PCI1451 transaction has terminated, then the PCI1451 terminates the transaction when its $\overline{\text{GNT}}$ is deasserted.

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

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

4.10 Header Type Register

The header type register returns 82h when read, indicating that the PCI1451 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 are user-definable extension registers.

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

Register: **Header type**
Type: Read-only
Offset: 0Eh (Functions 0, 1)
Default: 82h

4.11 BIST Register

Since the PCI1451 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 PCI1451 functions.

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

Register: **BIST**
Type: Read-only
Offset: 0Fh (Functions 0, 1)
Default: 00h

4.12 CardBus Socket/ExCA Base-Address Register

This register is programmed with a base address referencing the CardBus socket registers and the memory-mapped ExCA register set. Bits 31–12 are read/write and allow the base address to be located anywhere in the 32-bit PCI memory address space on a 4-Kbyte boundary. Bits 11–0 are read-only, returning 0s when read. When software writes all 1s to this register, the value read back will be FFFF F000h, indicating that at least 4K bytes of memory address space are required. The CardBus registers start at offset 000h, and the memory-mapped ExCA registers begin at offset 800h. This register is not shared by functions 0 and 1, mapping each socket control register separately.

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

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

4.13 Capability Pointer Register

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

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

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

4.14 Secondary Status Register

The secondary status register is compatible with the PCI-PCI bridge secondary status register and indicates CardBus related device information to the host system. This register is very similar to the status register (offset 06h, see Section 4.5), and status bits are cleared by a writing a 1. This register is not shared by the two socket functions, but is accessed on a per socket basis.

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

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

Table 4–4. Secondary Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	CBPARITY	R/C	Detected parity error. Bit 15 is set when a CardBus parity error (either address or data) is detected. Write a 1 to clear this bit.
14	CBSERR	R/C	Signaled system error. Bit 14 is set when $\overline{\text{CSERR}}$ is signaled by a CardBus card. The PCI1451 does not assert the $\overline{\text{CSERR}}$ signal. Write a 1 to clear this bit.
13	CBMABORT	R/C	Received master abort. Bit 13 is set when a cycle initiated by the PCI1451 on the CardBus bus has been terminated by a master abort. Write a 1 to clear this bit.
12	REC_CBTA	R/C	Received target abort. Bit 12 is set when a cycle initiated by the PCI1451 on the CardBus bus was terminated by a target abort. Write a 1 to clear this bit.
11	SIG_CBTA	R/C	Signaled target abort. Bit 11 is set by the PCI1451 when it terminates a transaction on the CardBus bus with a target abort. Write a 1 to clear this bit.
10–9	CB_SPEED	R	$\overline{\text{CDEVSEL}}$ timing. Bits 10 and 9 encode the timing of $\overline{\text{CDEVSEL}}$ and are hardwired to 01b, indicating that the PCI1451 asserts this signal at a medium speed.
8	CB_DPAR	R/C	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 PCI1451 was the bus master during the data parity error. c. Bit 0 (CPERREN) is set in the bridge control register (PCI offset 3Eh, see Section 4.25).
7	CBFBB_CAP	R	Fast back-to-back capable. The PCI1451 cannot accept fast back-to-back transactions; therefore, bit 7 is hardwired to 0.
6	CB_UDF	R	User-definable feature support. The PCI1451 does not support the user-definable features; therefore, bit 6 is hardwired to 0.
5	CB66MHZ	R	66 MHz capable. The PCI1451 CardBus interface operates at a maximum CCLK frequency of 33 MHz; therefore, bit 5 is hardwired to 0.
4–0	RSVD	R	Reserved. Bits 4–0 return 0s when read.

4.15 PCI Bus Number Register

The PCI bus number register is programmed by the host system to indicate the bus number of the PCI bus to which the PCI1451 is connected. The PCI1451 uses this register, in conjunction with the CardBus bus number and subordinate bus number registers, to determine when to forward PCI configuration cycles to its secondary buses.

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

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

4.16 CardBus Bus Number Register

The CardBus bus number register is programmed by the host system to indicate the bus number of the CardBus bus to which the PCI1451 is connected. The PCI1451 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 PCI1451 controller function.

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

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

4.17 Subordinate Bus Number Register

The subordinate bus number register is programmed by the host system to indicate the highest numbered bus below the CardBus bus. The PCI1451 uses this register, in conjunction with the PCI bus number and CardBus bus number registers, to determine when to forward PCI configuration cycles to its secondary buses. This register is separate for each CardBus controller function.

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

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

4.18 CardBus Latency Timer Register

The CardBus latency timer register is programmed by the host system to specify the latency timer for the PCI1451 CardBus interface, in units of CCLK cycles. When the PCI1451 is a CardBus initiator and asserts $\overline{\text{CFRAME}}$, the CardBus latency timer begins counting. If the latency timer expires before the PCI1451 transaction has terminated, then the PCI1451 terminates the transaction at the end of the next data phase. A recommended minimum value for this register of 20h allows most transactions to be completed.

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

Register: **CardBus latency timer**
Type: Read/Write
Offset: 1Bh (Functions 0, 1)
Default: 00h

4.19 Memory Base Registers 0, 1

The memory base registers indicate the lower address of a PCI memory address range. These registers are used by the PCI1451 to determine when to forward a memory transaction to the CardBus bus and when to forward a CardBus cycle to the PCI bus. Bits 31–12 of these registers are read/write and allow the memory base to be located anywhere in the 32-bit PCI memory space on 4-Kbyte boundaries. Bits 11–0 are read-only and always return 0s. Write transactions to these bits have no effect. Bits 8 (PREFETCH0) and 9 (PREFETCH1) of the bridge control register (PCI offset 3Eh, see Section 4.25) specify whether memory windows 0 and 1 are prefetchable or nonprefetchable. The memory base register or the memory limit register must be nonzero in order for the PCI1451 to claim any memory transactions through CardBus memory windows (that is, these windows are not enabled by default to pass the first 4K bytes of memory to CardBus).

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

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

4.20 Memory Limit Registers 0, 1

The memory limit registers indicate the upper address of a PCI memory address range. These registers are used by the PCI1451 to determine when to forward a memory transaction to the CardBus bus and when to forward a CardBus cycle to the PCI bus. Bits 31–12 of these registers are read/write and allow the memory base to be located anywhere in the 32-bit PCI memory space on 4-Kbyte boundaries. Bits 11–0 are read-only and always return 0s. Write transactions to these bits have no effect. Bits 8 (PREFETCH0) and 9 (PREFETCH1) of the bridge control register (PCI offset 3Eh, see Section 4.25) specify whether memory windows 0 and 1 are prefetchable or nonprefetchable. The memory base register or the memory limit register must be nonzero in order for the PCI1451 to claim any memory transactions through CardBus memory windows (that is, these windows are not enabled by default to pass the first 4K bytes of memory to CardBus).

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

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

4.21 I/O Base Registers 0, 1

The I/O base registers indicate the lower address of a PCI I/O address range. These registers are used by the PCI1451 to determine when to forward an I/O transaction to the CardBus bus and when to forward a CardBus cycle to the PCI bus. The lower 16 bits of this register locate the bottom of the I/O window within a 64-Kbyte page and the upper 16 bits (31–16) are all 0s which locate this 64-Kbyte page in the first page of the 32-bit PCI I/O address space.

Bits 31–16 and bits 1–0 are read-only and always return 0s, forcing I/O windows to be aligned on a natural doubleword boundary in the first 64-Kbyte page of PCI I/O address space. These I/O windows are enabled when either the I/O base register or the I/O limit register are nonzero. The I/O windows are not enabled by default to pass the first doubleword of I/O to CardBus.

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

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

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

4.22 I/O Limit Registers 0, 1

The I/O limit registers indicate the upper address of a PCI I/O address range. These registers are used by the PCI1451 to determine when to forward an I/O transaction to the CardBus bus and when to forward a CardBus cycle to the PCI bus. The lower 16 bits of this register locate the top of the I/O window within a 64-Kbyte page and the upper 16 bits are a page register which locates this 64-Kbyte page in 32-bit PCI I/O address space. Bits 15–2 are read/write and allow the I/O limit address to be located anywhere in the 64-Kbyte page (indicated by bits 31–16 of the appropriate I/O base register) on doubleword boundaries.

Bits 31–16 are read-only and always return 0s when read. The page is set in the I/O base register. Bits 1–0 are read-only and always return 0s, forcing I/O windows to be aligned on a natural doubleword boundary. Writes to read-only bits have no effect. The PCI1451 assumes that the lower 2 bits of the limit address are 1s.

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

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

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

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

4.23 Interrupt Line Register

The interrupt line register communicates interrupt line routing information to the host system. This register is not used by the PCI1451, since there are many programmable interrupt signaling options. This register is considered reserved; however, host software may read and write to this register. Each PCI1451 function has an interrupt line register.

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

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

4.24 Interrupt Pin Register

The value read from this register is function dependent and depends on bit 29 (INTRTIE) bit in the system control register (PCI offset 80h, see Section 4.29) and bits 2 and 1 (INTMODE field) in the device control register (PCI offset 92h, see Section 4.39). 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 4–5 for a complete description of the register contents.

PCI function 0

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

PCI function 1

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

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

Table 4–5. Interrupt Pin Register Cross Reference

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 ($\overline{\text{INTA}}$)

4.25 Bridge Control Register

The bridge control register provides control over various PCI1451 bridging functions. Bit 5 in this register is global in nature and is accessed only through function 0.

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

Register: **Bridge control**
Type: Read-only, Read/Write
Offset: 3Eh (Function 0, 1)
Default: 0340h

Table 4–6. Bridge Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–11	RSVD	R	Reserved. Bits 15–11 return 0s when read.
10	POSTEN	R/W	Write posting enable. Enables write posting to and from the CardBus sockets. Write posting enables posting of write data on burst cycles. Operating with write posting disabled inhibits performance on burst cycles. Note that bursted write data can be posted, but various write transactions may not. Bit 10 is socket dependent and is not shared between functions 0 and 1.
9	PREFETCH1	R/W	Memory window 1 type. This bit specifies whether or not memory window 1 is prefetchable. Bit 9 is socket dependent. This bit is encoded as: 0 = Memory window 1 is nonprefetchable. 1 = Memory window 1 is prefetchable (default).
8	PREFETCH0	R/W	Memory window 0 type. This bit specifies whether or not memory window 0 is prefetchable. Bit 8 is encoded as: 0 = Memory window 0 is nonprefetchable. 1 = Memory window 0 is prefetchable (default).
7	INTR	R/W	PCI interrupt – IREQ routing enable. Bit 7 selects whether PC Card functional interrupts are routed to PCI interrupts or to the IRQ specified in the ExCA registers. 0 = Functional interrupts are routed to PCI interrupts (default). 1 = Functional interrupts are routed by ExCA registers.
6	CRST	R/W	CardBus reset. When bit 6 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).
5	MABTMODE	R/W	Master abort mode. Bit 5 controls how the PCI1451 responds to a master abort when the PCI1451 is an initiator on the CardBus interface. This bit is common between each socket. 0 = Master aborts not reported (default). 1 = Signal target abort on PCI and signal $\overline{\text{SERR}}$, if enabled.
4	RSVD	R	Reserved. Bit 4 returns 0 when read.
3	VGAEN	R/W	VGA enable. Bit 3 affects how the PCI1451 responds to VGA addresses. When this bit is set, accesses to VGA addresses are forwarded.
2	ISAEN	R/W	ISA mode enable. Bit 2 affects how the PCI1451 passes I/O cycles within the 64-Kbyte ISA range. This bit is not common between sockets. When this bit is set, the PCI1451 does not forward the last 768 bytes of each 1K I/O range to CardBus.
1	CSERREN	R/W	$\overline{\text{CSERR}}$ enable. Bit 1 controls the response of the PCI1451 to $\overline{\text{CSERR}}$ signals on the CardBus bus. This bit is separate for each socket. 0 = $\overline{\text{CSERR}}$ is not forwarded to PCI $\overline{\text{SERR}}$. 1 = $\overline{\text{CSERR}}$ is forwarded to PCI $\overline{\text{SERR}}$.
0	CPERREN	R/W	CardBus parity error response enable. Bit 0 controls the response of the PCI1451 to CardBus parity errors. This bit is separate for each socket. 0 = CardBus parity errors are ignored. 1 = CardBus parity errors are reported using $\overline{\text{CPERR}}$.

4.26 Subsystem Vendor ID Register

The subsystem vendor ID register, used for system and option card identification purposes, may be required for certain operating systems. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register (PCI offset 80h, see Section 4.29).

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

Register: **Subsystem vendor ID**
Type: Read-only (Read/Write if enabled by SUBSYSRW)
Offset: 40h (Functions 0, 1)
Default: 0000h

4.27 Subsystem ID Register

The subsystem ID register, used for system and option card identification purposes, may be required for certain operating systems. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register (PCI offset 80h, see Section 4.29). If an EEPROM is present, then the subsystem ID and subsystem vendor ID will be loaded from EEPROM after a reset.

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

Register: **Subsystem ID**
Type: Read-only (Read/Write if enabled by SUBSYSRW)
Offset: 42h (Functions 0, 1)
Default: 0000h

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

The PCI1451 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. See Chapter 5, *ExCA Compatibility Registers*, for register offsets.

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

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

4.29 System Control Register

System-level initializations are performed through programming this doubleword register. Bits 31–29, 27, 26, 24, 15, 14, 6–3, 1, and 0 are global in nature and are accessed only through function 0.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	System control															
Type	R/W	R/W	R/W	R	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

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

Table 4–7. System Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–30	SER_STEP	R/W	Serialized PCI interrupt routing step. Bits 31 and 30 configure the serialized PCI interrupt stream signaling and accomplish an even distribution of interrupts signaled on the four PCI interrupt slots. These bits are global to both PCI1451 functions. 00 = INTA/INTB signal in INTA/INTB IRQSER slots (default) 01 = INTA/INTB signal in INTB/INTC IRQSER slots 10 = INTA/INTB signal in INTC/INTD IRQSER slots 11 = INTA/INTB signal in INTD/INTA IRQSER slots
29	INTRTIE	R/W	Tie internal PCI interrupts. When bit 29 is set, the INTA and INTB signals are tied together internally and are signaled as INTA. INTA may then be shifted by using bits 31 and 30 (SER_STEP). This bit is global to both PCI1451 functions. 0 = INTA and INTB are not tied together internally (default). 1 = INTA and INTB are tied together internally.
28	RSVD	R	Reserved. Bit 28 returns 0 when read.
27	P2CCLK	R/W	P2C power switch CLOCK. Bit 27 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 PCI1451 generated CLOCK. A 43kΩ pulldown resistor should be tied to this terminal.
26	SMIROUTE	R/W	SMI interrupt routing. Bit 26 is shared between functions 0 and 1, and selects whether IRQ2 or CSC is signaled when a write occurs to power a PC Card socket. 0 = PC Card power change interrupts routed to IRQ2 (default). 1 = A CSC interrupt is generated on PC Card power changes.
25	SMISTATUS	R/W	SMI interrupt status. This socket dependent bit is set when bit 24 (SMIENB) is set and a write occurs to set the socket power. Writing a 1 to bit 25 clears the status. 0 = SMI interrupt is signaled. 1 = SMI interrupt is not signaled.
24	SMIENB	R/W	SMI interrupt mode enable. When bit 24 is set, the SMI interrupt signaling generates an interrupt when a write to the socket power control occurs. This bit is shared and defaults to 0 (disabled). 0 = SMI interrupt mode is disabled (default). 1 = SMI interrupt mode is enabled.
23	RSVD	R	Reserved. Bit 23 returns 0 when read.

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

BIT	SIGNAL	TYPE	FUNCTION
22	CBRSVD	R/W	CardBus reserved terminals signaling. When bit 22 is set, the RSVD CardBus terminals are driven low when a CardBus card is inserted. When bit 22 is low, as default, these signals are placed in a high-impedance state. 0 = Place the CardBus RSVD terminals in a high-impedance state 1 = Drive the Cardbus RSVD terminals low (default).
21	VCCPROT	R/W	V_{CC} protection enable. This bit is socket dependent. 0 = V_{CC} protection is enabled for 16-bit cards (default). 1 = V_{CC} protection is disabled for 16-bit cards.
20	REDUCEZV	R/W	Reduced zoomed video enable. When bit 20 is enabled, terminals A25–A22 of the card interface for PC Card 16 cards is placed in the high impedance state. This bit is encoded as: 0 = Reduced zoomed video is disabled (default). 1 = Reduced zoomed video is enabled.
19	CDREQEN	R/W	PC/PCI DMA card enable. When bit 19 is set, the PCI1451 allows 16-bit PC Cards to request PC/PCI DMA using the $\overline{DREQ}$ signaling. $\overline{DREQ}$ is selected through the socket DMA register 0 (PCI offset 94h, see Section 4.41). 0 = Ignore $\overline{DREQ}$ signaling from PC Cards (default). 1 = Signal DMA request on $\overline{DREQ}$.
18–16	CDMACHAN	R/W	PC/PCI DMA channel assignment. Bits 18–16 are encoded as: 0–3 = 8-bit DMA channels 4 = PCI master; not used (default) 5–7 = 16-bit DMA channels
15	MRBURSTDN	R/W	Memory read burst enable downstream. When bit 15 is set, memory read transactions are allowed to burst downstream. 0 = Downstream memory read burst is disabled. 1 = Downstream memory read burst is enabled (default).
14	MRBURSTUP	R/W	Memory read burst enable upstream. When bit 14 is set, the PCI1451 allows memory read transactions to burst upstream. 0 = Upstream memory read burst is disabled (default). 1 = Upstream memory read burst is enabled.
13	SOCACTIVE	R	Socket activity status. When set, bit 13 indicates access has been performed to or from a PC Card, and is cleared upon read of this status bit. This bit is socket dependent. 0 = No socket activity (default) 1 = Socket activity
12	RSVD	R	Reserved. This bit returns 1 when read. This is the clamping voltage bit in functions 0 and 1.
11	PWRSTREAM	R	Power stream in progress status bit. When set, bit 11 indicates that a power stream to the power switch is in progress and a powering change has been requested. This bit is cleared when the power stream is complete. 0 = Power stream is complete, delay has expired. 1 = Power stream is in progress.
10	DELAYUP	R	Power-up delay in progress status bit. When set, bit 10 indicates that a power-up stream has been sent to the power switch and proper power may not yet be stable. This bit is cleared when the power-up delay has expired. 0 = Power-up delay has expired. 1 = Power-up stream sent to switch. Power might not be stable.
9	DELAYDOWN	R	Power-down delay in progress status bit. When set, bit 9 indicates that a power-down stream has been sent to the power switch and proper power may not yet be stable. This bit is cleared when the power-down delay has expired. 0 = Power-down delay has expired. 1 = Power-down stream sent to switch. Power might not be stable.
8	INTERROGATE	R	Interrogation in progress. When set, bit 8 indicates an interrogation is in progress and clears when the interrogation completes. This bit is socket dependent. 0 = Interrogation not in progress (default) 1 = Interrogation in progress
7	RSVD	R	Reserved. Bit 7 returns 0 when read.

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

BIT	SIGNAL	TYPE	FUNCTION
6	PWRSAVINGS	R/W	Power savings mode enable. When bit 6 is set, the PCI1451 will consume less power with no performance loss. This bit is shared between the two PCI1451 functions. 0 = Power savings mode disabled 1 = Power savings mode enabled (default)
5	SUBSYSRW	R/W	Subsystem ID (see Section 4.27), subsystem vendor ID (see Section 4.26), and the ExCA identification and revision (see Section 5.1) registers read/write enable. Bit 5 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	CB_DPAR	R/W	CardBus data parity $\overline{\text{SERR}}$ signaling enable. 0 = CardBus data parity not signaled on PCI $\overline{\text{SERR}}$ signal (default) 1 = CardBus data parity signaled on PCI $\overline{\text{SERR}}$ signal
3	CDMA_EN	R/W	PC/PCI DMA enable. Enables PC/PCI DMA when set. When PC/PCI DMA is enabled, $\overline{\text{PCREQ}}$ and PCGNT should be routed to a multifunction routing terminal. See multifunction routing status register (PCI offset 8Ch, see Section 4.36) for options. 0 = Centralized DMA disabled (default) 1 = Centralized DMA enabled
2	RSVD	R	Reserved. Bit 2 returns 0 when read.
1	KEEPCLK	R/W	Keep clock. When bit 1 is set, the PCI1451 will always follow $\overline{\text{CLKRUN}}$ protocol to maintain the system PCLK and the CCLK (CardBus clock). This bit is global to the PCI1451 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 PCI1451, setting this bit maintains both the PCI clock and the CCLK.
0	RIMUX	R/W	$\overline{\text{PME}}$/$\overline{\text{RI\_OUT}}$ select bit. When bit 0 is 1, the $\overline{\text{PME}}$ signal is routed on to the $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ terminal. When this bit is 0 and bit 7 (RIENB) of the card control register (PCI offset 91h, see Section 4.38) is 1, the $\overline{\text{RI\_OUT}}$ signal is routed on to the $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ terminal. If this bit is 0 and bit 7 (RIENB) of the card control register is 0, then the output on the $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ terminal is placed in a high-impedance state. This terminal is encoded as: 0 = $\overline{\text{RI\_OUT}}$ signal is routed to the $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ terminal if bit 7 of the card control register is 1 (default). 1 = $\overline{\text{PME}}$ signal is routed on to the $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ terminal of the PCI1451 controller.

4.30 Multimedia Control Register

The multimedia control register provides port mapping for the PCI1451 zoomed video/data ports. See Section 3.4.3, *Zoomed Video Support*, for details on the PCI1451 zoomed video support. Access this register only through function 0.

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

Register: **Multimedia control**
Type: Read/Write
Offset: 84h (Functions 0, 1)
Default: 00h

Table 4–8. Multimedia Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	ZVOUTEN	R/W	ZV output enable. Bit 7 enables the output for the PCI1451 outsourcing ZV terminals. When this bit is reset, these terminals are in a high-impedance state. 0 = PCI1451 ZV output terminals disabled (default) 1 = PCI1451 ZV output terminals enabled
6	PORTSEL	R/W	ZV port select. Bit 6 controls the multiplexing control over which PC Card ZV port data is driven to the outsourcing PCI1451 ZV port. 0 = Output card 0 ZV if enabled (default) 1 = Output card 1 ZV if enabled
5	ZVAUTO	R/W	Zoomed video auto-detect. Bit 5 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	AUTODETECT	R/W	Auto-detect priority encoding. Bits 4–2 have meaning only if bit 5 (ZVAUTO) is enabled. If bit 5 is enabled, then bits 4–2 are encoded as follows: 000 = Slot A, Slot B, External Source 001 = Slot A, External Source, Slot B 010 = Slot B, Slot A, External Source 011 = Slot B, External Source, Slot A 100 = External Source, Slot A, Slot B 101 = External Source, Slot B, Slot A 110 = Reserved 111 = Reserved
1	ZVEN1	R/W	PC Card 1 ZV mode enable. Enables the zoomed video mode for socket 1. When bit 1 set, the PCI1451 inputs ZV data from the PC Card interface, and disables output drivers on ZV terminals. 0 = PC Card 1 ZV disabled (default) 1 = PC Card 1 ZV enabled
0	ZVEN0	R/W	PC Card 0 ZV mode enable. Enables the zoomed video mode for socket 0. When bit 0 set, the PCI1451 inputs ZV data from the PC Card interface, and disables output drivers on ZV terminals. 0 = PC Card 0 ZV disabled (default) 1 = PC Card 0 ZV enabled

4.31 General Status Register

The general status register provides the general device status information. The status of the serial EEPROM interface is provided through this register. Bits 2–0 are global in nature and are accessed only through function 0.

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

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

Table 4–9. General Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–3	RSVD	R	Reserved. Bits 7–3 return 0s when read.
2	EEDETECT	R	Serial EEPROM detect. When bit 2 is cleared, it indicates that the PCI1450 serial EEPROM circuitry has detected an EEPROM. A pullup resistor must be implemented on the SDA terminal for this bit to be set. This status bit is encoded as: 0 = EEPROM not detected (default) 1 = EEPROM detected
1	DATAERR	R/C	Serial EEPROM data error status. Bit 1 indicates when a data error occurs on the serial EEPROM interface. This bit will be set due to a missing acknowledge. This bit is cleared by a writeback of 1. 0 = No error detected. (default) 1 = Data error detected.
0	EEBUSY	R	Serial EEPROM busy status. Bit 0 indicates the status of the PCI1451 serial EEPROM circuitry. This bit is set during the loading of the subsystem ID value. 0 = Serial EEPROM circuitry is not busy (default). 1 = Serial EEPROM circuitry is busy.

4.32 General-Purpose Event Status Register

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

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

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

Table 4–10. General-Purpose Event Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	PWR_STS	RCU	Power change status. Bit 7 is set when software changes the V_{CC} or V_{PP} power state of either socket.
6	VPP12_STS	RCU	12V V_{PP} request status. Bit 6 is set when software has changed the requested V_{PP} level to or from 12 V for either socket.
5–4	RSVD	R	Reserved. Bits 5 and 4 return 0s when read.
3	GP3_STS	RCU	GPI3 status. Bit 3 is set on a change in status of the MFUNC3 terminal input level if configured as a general-purpose input, GPI3.
2	GP2_STS	RCU	GPI2 status. Bit 2 is set on a change in status of the MFUNC2 terminal input level if configured as a general-purpose input, GPI2.
1	GP1_STS	RCU	GPI1 status. Bit 1 is set on a change in status of the MFUNC1 terminal input level if configured as a general-purpose input, GPI1.
0	GP0_STS	RCU	GPI0 status. Bit 0 is set on a change in status of the MFUNC0 terminal input level if configured as a general-purpose input, GPI0.

4.33 General-Purpose Event Enable Register

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

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

Register: **General-purpose event enable**
Type: Read-only, Read/Write
Offset: 89h
Default: 00h

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

BIT	SIGNAL	TYPE	FUNCTION
7	PWR_EN	R/W	Power change $\overline{\text{GPE}}$ enable. When bit 7 is set, $\overline{\text{GPE}}$ is signaled on PWR_STS events.
6	VPP12_EN	R/W	12-Volt V_{PP} $\overline{\text{GPE}}$ enable. When bit 6 is set, $\overline{\text{GPE}}$ is signaled on VPP12_STS events.
5–4	RSVD	R	Reserved. Bits 5 and 4 return 0s when read.
3	GP3_EN	R/W	GPI3 $\overline{\text{GPE}}$ enable. When bit 3 is set, $\overline{\text{GPE}}$ is signaled on GP3_STS events.
2	GP2_EN	R/W	GPI2 $\overline{\text{GPE}}$ enable. When bit 2 is set, $\overline{\text{GPE}}$ is signaled on GP2_STS events.
1	GP1_EN	R/W	GPI1 $\overline{\text{GPE}}$ enable. When bit 1 is set, $\overline{\text{GPE}}$ is signaled on GP1_STS events.
0	GP0_EN	R/W	GPI0 $\overline{\text{GPE}}$ enable. When bit 0 is set, $\overline{\text{GPE}}$ is signaled on GP0_STS events.

4.34 General-Purpose Input Register

The general-purpose input register contains GPI terminal status.

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

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

Table 4–12. General-Purpose Input Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–4	RSVD	R	Reserved. Bits 7–4 return 0s when read.
3	GPI3_DATA	RU	GPI3 data input. Bit 3 represents the logical value of the data input from GPI3.
2	GPI2_DATA	RU	GPI2 data input. Bit 2 represents the logical value of the data input from GPI2.
1	GPI1_DATA	RU	GPI1 data input. Bit 1 represents the logical value of the data input from GPI1.
0	GPI0_DATA	RU	GPI0 data input. Bit 0 represents the logical value of the data input from GPI0.

4.35 General-Purpose Output Register

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

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

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

Table 4–13. General-Purpose Output Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–4	RSVD	R	Reserved. Bits 7–4 return 0s when read.
3	GPO3_DATA	R/W	Bit 3 represents the logical value of the data driven to GPO3.
2	GPO2_DATA	R/W	Bit 2 represents the logical value of the data driven to GPO2.
1	GPO1_DATA	R/W	Bit 1 represents the logical value of the data driven to GPO1.
0	GPO0_DATA	R/W	Bit 0 represents the logical value of the data driven to GPO0.

4.36 Multifunction Routing Status Register

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

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

Register: **Multifunction routing status**
Type: Read/Write
Offset: 8Ch
Default: 0000 0000h

Table 4–14. Multifunction Routing Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
31	RSVD	R	Reserved. Bit 31 returns 0 when read.
30–28	MFUNC7_SEL	R/W	MFUNC7 select. Bits 30–28 control the mapping of MFUNC7 as follows: 000 = IDSEL 001 = $\overline{\text{RI\_OUT}}$ 010 = RSVD 011 = $\overline{\text{PCREQ}}$ 100 = $\overline{\text{D3\_STAT}}$ 101 = LOCK 110 = RSVD 111 = RSVD
27	RSVD	R	Reserved. Bit 27 returns 0 when read.
26–24	MFUNC6_SEL	R/W	MFUNC6 select. Bits 26–24 control the mapping of MFUNC6 as follows: 000 = RSVD 001 = RSVD 010 = RSVD 011 = RSVD 100 = $\overline{\text{D3\_STAT}}$ 101 = RSVD 110 = CAUDPWM 111 = $\overline{\text{PCGNT}}$
23	RSVD	R	Reserved. Bit 23 returns 0 when read.
22–20	MFUNC5_SEL	R/W	MFUNC5 select. Bits 22–20 control the mapping of MFUNC5 as follows: 000 = RSVD 001 = RSVD 010 = RSVD 011 = RSVD 100 = RSVD 101 = $\overline{\text{GPE}}$ 110 = CAUDPWM 111 = $\overline{\text{PCREQ}}$
19	RSVD	R	Reserved. Bit 19 returns 0 when read.
18–16	MFUNC4_SEL	R/W	MFUNC4 select. Bits 18–16 control the mapping of MFUNC4 as follows: 000 = RSVD 001 = RSVD 010 = LEDA1 011 = $\overline{\text{PCREQ}}$ 100 = RSVD 101 = $\overline{\text{GPE}}$ 110 = RSVD 111 = ZV_STAT
15	RSVD	R	Reserved. Bit 15 returns 0 when read.
14–12	MFUNC3_SEL	R/W	MFUNC3 select. Bits 14–12 control the mapping of MFUNC3 as follows: 000 = GPI3 001 = GPO3 010 = LEDA2 011 = $\overline{\text{PCGNT}}$ 100 = RSVD 101 = LOCK 110 = RSVD 111 = C_ZVCLK
11	RSVD	R	Reserved. Bit 11 returns 0 when read.

Table 4–14. Multifunction Routing Status Register Description (continued)

BIT	SIGNAL	TYPE	FUNCTION
10–8	MFUNC2_SEL	R/W	MFUNC2 select. Bits 10–8 control the mapping of MFUNC2 as follows: 000 = GPI2 001 = GPO2 010 = INTC 011 = PCGNT 100 = D3_STAT 101 = RSVD 110 = RSVD 111 = C_ZVCLK
7	RSVD	R	Reserved. Bit 7 returns 0 when read.
6–4	MFUNC1_SEL	R/W	MFUNC1 select. Bits 6–4 control the mapping of MFUNC1 as follows: 000 = GPI1 001 = GPO1 010 = INTB 011 = TEST_MUX 100 = D3_STAT 101 = LOCK 110 = CAUDPWM 111 = ZV_STAT
3	RSVD	R	Reserved. Bit 3 returns 0 when read.
2–0	MFUNC0_SEL	R/W	MFUNC0 select. Bits 2–0 control the mapping of MFUNC0 as follows: 000 = GPI0 001 = GPO0 010 = INTA 011 = PCREQ 100 = RSVD 101 = GPE 110 = RSVD 111 = ZV_STAT

4.37 Retry Status Register

The retry status register enables the retry time-out counters and displays the retry expiration status. The flags are set when the PCI1451 retries a PCI or CardBus master request and the master does not return within 2^{15} PCI clock cycles. The flags are cleared by writing a 1 to the bit. These bits are expected to be incorporated into the command register (see Section 4.4), status register (see Section 4.5), and bridge control register (see Section 4.25) by the PCI SIG. Access this register only through function 0.

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

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

Table 4–15. Retry Status Register Description

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

4.38 Card Control Register

The card control register is provided for PCI1130 compatibility. The contents provide the PC Card function interrupt flag (IFG) and an alias for the ZVEN0 and ZVEN1 bits found in the PCI1451 multimedia control register (see Section 4.30). When this register is accessed by function 0, the ZVEN0 bit will alias with bit 6 (ZVENABLE). When this register is accessed by function 1, the ZVEN1 bit will alias with bit 6 (ZVENABLE). Setting bit 6 only places the PC Card socket interface ZV terminals in a high impedance state, but does not enable the PCI1451 to drive ZV data onto the ZV terminals.

The $\overline{\text{RI\_OUT}}$ signal is enabled through this register, and bit 7 (RIENB) is shared between functions 0 and 1.

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

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

Table 4–16. Card Control Register Description

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

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

4.39 Device Control Register

The device control register is provided for PCI1130 compatibility. It contains bits which are shared between functions 0 and 1. The interrupt mode select and the socket-capable force bits are programmed through this register. Bits 6 and 3–0 are global in nature and should be accessed only through function 0.

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

Register: **Device control**
Type: Read-only, Read/Write
Offset: 92h (Functions 0, 1)
Default: 66h

Table 4–17. Device Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	SKTPWR_LOCK	R/W	Socket power lock bit. When bit 7 is set to 1, software will not be able to power down the PC Card socket while in D3. This may be necessary to support wake on LAN or RING if the operating system is programmed to power down a socket when the CardBus controller is placed in the D3 state.
6	3VCAPABLE	R/W	3-V socket capable force bit. 0 = Not 3-V capable 1 = 3-V capable (default)
5	IO16R2	R/W	Diagnostic bit. Bit 5 defaults to 1.
4	RSVD	R	Reserved. Bit 4 returns 0 when read. A write has no effect.
3	TEST	R/W	TI test bit. Write only 0 to this bit. This bit can be set to shorten the interrogation counter.
2–1	INTMODE	R/W	Interrupt mode. Bits 2–1 select the interrupt signaling mode. The interrupt mode bits are encoded: 00 = Parallel PCI interrupts only 01 = Reserved 10 = IRQ serialized interrupts & parallel PCI interrupts $\overline{\text{INTA}}$ and $\overline{\text{INTB}}$ 11 = IRQ & PCI serialized interrupts (default)
0	RSVD	R/W	Reserved. NAND tree enable bit. There is a NAND tree diagnostic structure in the PCI1451, and it tests only the terminals that are inputs or I/Os. Any output only terminal on the PCI1451 is excluded from the NAND tree test.

4.40 Diagnostic Register

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

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

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

Table 4–18. Diagnostic Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	TRUE_VAL	R/W	This bit defaults to 0. This bit is encoded as: 0 = Reads true values in vendor ID (see Section 4.2) and device ID (see Section 4.3) registers (default). 1 = Reads all ones in reads to the PCI vendor ID and PCI device ID registers.
6	RSVD	R/W	Reserved.
5	CSC	R/W	CSC interrupt routing control 0 = CSC interrupts routed to PCI if ExCA 803 (see Section 5.4) bit 4 = 1. 1 = CSC Interrupts routed to PCI if ExCA 805 (see Section 5.6) bits 7–4 = 0000b. (Default) In this case, the setting of ExCA 803 bit 4 is a “don’t care.”
4	DIAG4	R/W	Diagnostic RETRY_DIS. Delayed transaction disable.
3	DIAG3	R/W	Diagnostic RETRY_EXT. Extends the latency from 16 to 64.
2	DIAG2	R/W	Diagnostic DISCARD_TIM_SEL_CB. Set = 2^{10}, Reset = 2^{15}
1	DIAG1	R/W	Diagnostic DISCARD_TIM_SEL_PCI. Set = 2^{10}, Reset = 2^{15}
0	ASYNC_CSC	R/W	Asynchronous interrupt generation. 0 = CSC interrupt not generated asynchronously 1 = CSC interrupt is generated asynchronously (default)

4.41 Socket DMA Register 0

Socket DMA register 0 provides control over the PC Card $\overline{\text{DREQ}}$ (DMA request) signaling.

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

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

Table 4–19. Socket DMA Register 0 Description

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

4.42 Socket DMA Register 1

Socket DMA register 1 provides control over the distributed DMA (DDMA) registers and the PCI portion of DMA transfers. The DMA base address locates the DDMA registers in a 16-byte region within the first 64K bytes of PCI I/O address space. Note that 32-bit transfers to the 16-bit PC Card interface are not supported; the maximum transfer possible to the PC Card interface is 16 bits. However, 32 bits of data are prefetched from the PCI bus, thus allowing back-to-back 16-bit transfers to the PC Card interface.

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

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

Table 4–20. Socket DMA Register 1 Description

BIT	SIGNAL	TYPE	FUNCTION
31–16	RSVD	R	Reserved. Bits 31–16 return 0s when read.
15–4	DMABASE	R/W	DMA base address. Locates the socket's DMA registers in PCI I/O space. This field represents a 16-bit PCI I/O address. The upper 16 bits of the address are hardwired to 0, forcing this window to within the lower 64K bytes of I/O address space. The lower 4 bits are hardwired to 0, and are included in the address decode. Thus, the window is aligned to a natural 16-byte boundary.
3	EXTMODE	R	Extended addressing. This feature is not supported by the PCI1451, and always returns a 0.
2–1	XFERSIZE	R/W	Transfer size. Bits 2 and 1 specify the width of the DMA transfer on the PC Card interface, and are encoded as: 00 = Transfers are 8 bits (default). 01 = Transfers are 16 bits. 10 = Reserved 11 = Reserved
0	DDMAEN	R/W	DDMA registers decode enable. Enables the decoding of the distributed DMA registers based upon the value of bits 15–4 (DMABASE field). 0 = Disabled (default) 1 = Enabled

4.43 Capability ID Register

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

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

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

4.44 Next Item Pointer Register

The contents of this register indicate the next item in the linked list of the PCI power management capabilities. Since the PCI1451 functions only include one capabilities item, this register returns 0s when read.

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

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

4.45 Power Management Capabilities Register

The power management capabilities register contains information on the capabilities of the PC Card function related to power management. Both PCI1451 CardBus bridge functions support D0, D1, D2, and D3 power states.

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

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

Table 4–21. Power Management Capabilities Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	PME_Support	R/W	$\overline{\text{PME}}$ support. This 5-bit field indicates the power states from which the PCI1451 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 a 1 indicating the $\overline{\text{PME}}$ signal can be asserted from the $D3_{cold}$ state. This bit is read/write because wake-up support from $D3_{cold}$ is contingent on the system providing an auxiliary power source to the V_{CC} terminals. If the system designer chooses not to provide an auxiliary power source to the V_{CC} terminals for $D3_{cold}$ wake-up support, then BIOS should write a 0 to this bit.
14–11	PME_Support	R	Bit 14 – contains the value 1 to indicate 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 to indicate that the $\overline{\text{PME}}$ signal can be asserted from the D0 state.
10	D2_Support	R	D2 support. Bit 10 returns a 1 when read, indicating that the CardBus function supports the D2 device power state.
9	D1_Support	R	D1 support. Bit 9 returns a 1 when read, indicating that the CardBus function supports the D1 device power state.
8–6	RSVD	R	Reserved. Bits 8–6 return 000b when read.
5	DSI	R	Device specific initialization. Bit 5 returns 0 when read.
4	AUX_PWR	R	Auxiliary power source. Bit 4 is meaningful only if bit 15 (PME_Support, $D3_{cold}$) is set. When bit 4 is set, it indicates that support for $\overline{\text{PME}}$ in $D3_{cold}$ requires auxiliary power supplied by the system by way of a proprietary delivery vehicle. When bit 4 is 0, it indicates that the function supplies its own auxiliary power source.
3	PMECLK	R	When bit 3 is 1, it indicates that the function relies on the presence of the PCI clock for $\overline{\text{PME}}$ operation. When bit 3 is 0, it indicates that no PCI clock is required for the function to generate $\overline{\text{PME}}$.
2–0	VERSION	R	Version. Bits 2–0 return 001b when read, indicating that there are 4 bytes of general-purpose power management (PM) registers as described in the *PCI Bus Power Management Interface Specification*.

4.46 Power Management Control/Status Register

The power management control/status register determines and changes the current power state of the PCI1451 CardBus function. The contents of this register are not affected by the internally generated reset caused by the transition from the $D3_{hot}$ to D0 state.

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

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

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

Table 4–22. Power Management Control/Status Register Description

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

4.47 Power Management Control/Status Register Bridge Support Extensions

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

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

Register: **Power management control/status register bridge support extensions**
Type: Read-only
Offset: A6h (Functions 0, 1)
Default: C0h

Table 4–23. Power Management Control/Status Register Bridge Support Extensions

BIT	SIGNAL	TYPE	FUNCTION
7	BPCC_EN	R	Bus power/clock control enable. This bit returns 1 when read. This bit is encoded as: 0 = Bus power/clock control is disabled. 1 = Bus power/clock control is enabled (default). A 0 indicates that the bus power/clock control policies defined in the *PCI Bus Power Management Interface Specification* are disabled. When the bus power/clock control enable mechanism is disabled, the bridge's power management control/status register power state field (see Section 4.46, bits 1–0) cannot be used by the system software to control the power or the clock of the bridge's secondary bus. A 1 indicates that the bus power/clock control mechanism is enabled.
6	B2_$\overline{B3}$	R	B2/B3 support for $D3_{hot}$. The state of this bit determines the action that is to occur as a direct result of programming the function to $D3_{hot}$. This bit is only meaningful if bit 7 (BPCC_EN) is a 1. This bit is encoded as: 0 = When the bridge is programmed to $D3_{hot}$, its secondary bus will have its power removed (B3). 1 = When the bridge function is programmed to $D3_{hot}$, its secondary bus's PCI clock is stopped (B2). (Default)
5–0	RSVD	R	Reserved. Bits 5–0 return 0s when read.

4.48 General-Purpose Event Control/Status Register

If the $\overline{\text{GPE}}$ (general-purpose event) function is programmed onto the MFUNC5 terminal by writing 101b to bits 22–20 of the multifunction routing status register (PCI offset 8Ch, see Section 4.36), then this register may be used to program which events will cause $\overline{\text{GPE}}$ to be asserted and report the status.

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

Table 4–24. $\overline{\text{GPE}}$ Control/Status Register Description

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

5 ExCA Compatibility Registers (Functions 0 and 1)

The ExCA (exchangeable card architecture) registers implemented in the PCI1451 are register-compatible with the Intel 82365SL-DF PCMCIA controller. ExCA registers are identified by an offset value, which is compatible with the legacy I/O index/data scheme used on the Intel 82365 ISA controller. The ExCA registers are accessed through this scheme by writing the register offset value into the index register (I/O base), and reading or writing the data register (I/O base + 1). The I/O base address used in the index/data scheme is programmed in the PC Card 16-bit I/F legacy mode base address register (see Section 4.28), which is shared by both card sockets. The offsets from this base address run contiguously from 00h to 3Fh for socket A, and from 40h to 7Fh for socket B. Refer to Figure 5–1 for an ExCA I/O mapping illustration. Table 5–1 identifies each ExCA register and its respective ExCA offset.

The TI PCI1451 also provides a memory-mapped alias of the ExCA registers by directly mapping them into PCI memory space. They are located through the CardBus socket/ExCA base address register (PCI register 10h, see Section 4.12) at memory offset 800h. Each socket has a separate base address programmable by function. Refer to Figure 5–2 for an ExCA memory mapping illustration. Note that memory offsets are 800h–844h for both functions 0 and 1. This illustration also identifies the CardBus socket register mapping, which is mapped into the same 4K window at memory offset 0h.

The interrupt registers, as defined by the 82365SL Specification, in the ExCA register set control such card functions as reset, type, interrupt routing, and interrupt enables. Special attention must be paid to the interrupt routing registers and the host interrupt signaling method selected for the PCI1451 to ensure that all possible PCI1451 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.

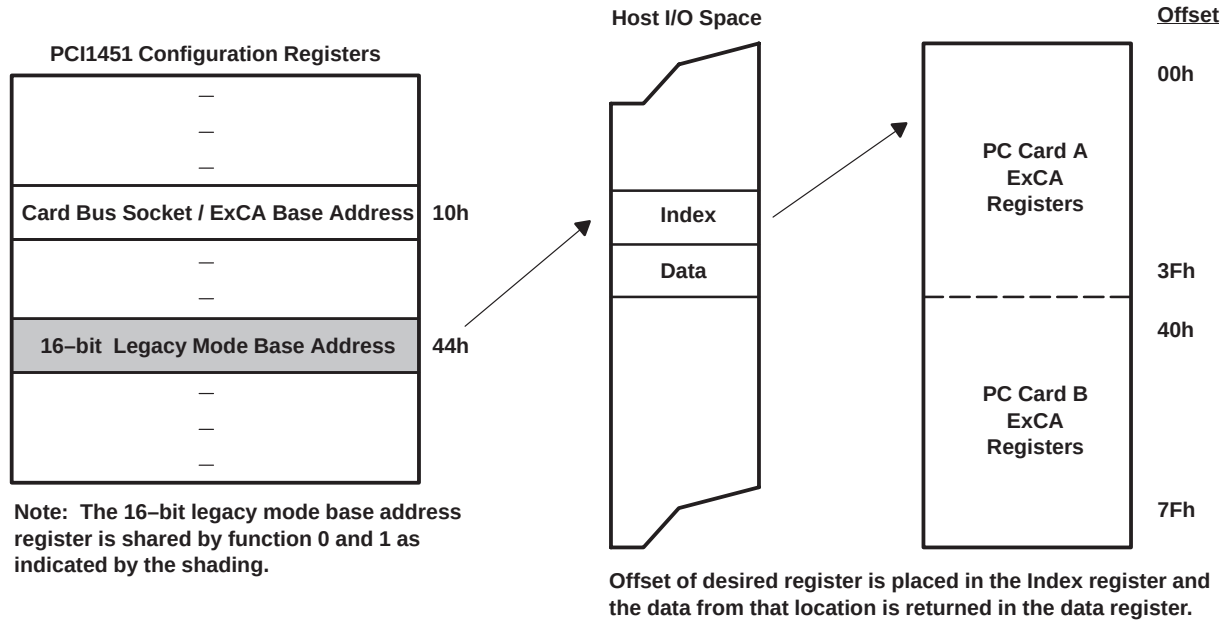


Figure 5–1. ExCA Register Access Through I/O

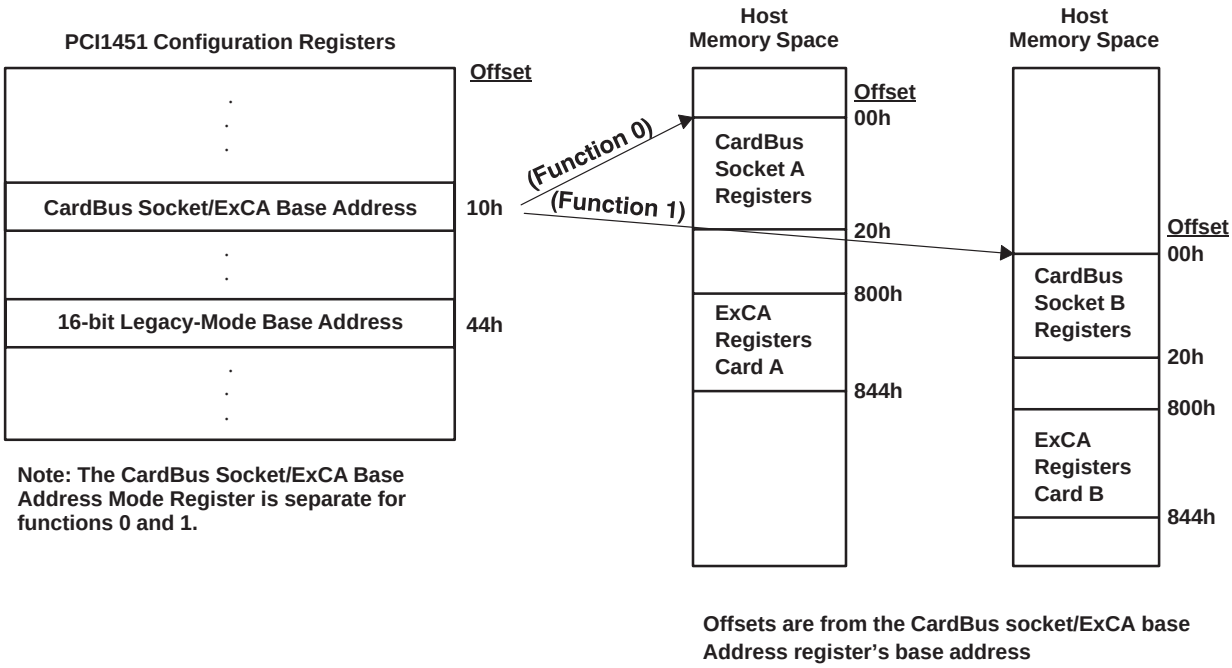


Figure 5–2. ExCA Register Access Through Memory

Table 5–1. ExCA Registers and Offsets

REGISTER NAME	PCI MEMORY ADDRESS OFFSET	EXCA OFFSET (CARD A)	EXCA OFFSET (CARD B)
Identification and revision	800	00	40
Interface status	801	01	41
Power control	802	02	42
Interrupt and general control	803	03	43
Card status change	804	04	44
Card status change interrupt configuration	805	05	45
Address window enable	806	06	46
I / O window control	807	07	47
I / O window 0 start-address low-byte	808	08	48
I / O window 0 start-address high-byte	809	09	49
I / O window 0 end-address low-byte	80A	0A	4A
I / O window 0 end-address high-byte	80B	0B	4B
I / O window 1 start-address low-byte	80C	0C	4C
I / O window 1 start-address high-byte	80D	0D	4D
I / O window 1 end-address low-byte	80E	0E	4E
I / O window 1 end-address high-byte	80F	0F	4F
Memory window 0 start-address low-byte	810	10	50
Memory window 0 start-address high-byte	811	11	51
Memory window 0 end-address low-byte	812	12	52
Memory window 0 end-address high-byte	813	13	53
Memory window 0 offset-address low-byte	814	14	54
Memory window 0 offset-address high-byte	815	15	55
Card detect and general control	816	16	56
Reserved	817	17	57
Memory window 1 start-address low-byte	818	18	58
Memory window 1 start-address high-byte	819	19	59
Memory window 1 end-address low-byte	81A	1A	5A
Memory window 1 end-address high-byte	81B	1B	5B
Memory window 1 offset-address low-byte	81C	1C	5C
Memory window 1 offset-address high-byte	81D	1D	5D
Global control	81E	1E	5E
Reserved	81F	1F	5F
Memory window 2 start-address low-byte	820	20	60
Memory window 2 start-address high-byte	821	21	61
Memory window 2 end-address low-byte	822	22	62
Memory window 2 end-address high-byte	823	23	63
Memory window 2 offset-address low-byte	824	24	64
Memory window 2 offset-address high-byte	825	25	65
Reserved	826	26	66
Reserved	827	27	67
Memory window 3 start-address low-byte	828	28	68
Memory window 3 start-address high-byte	829	29	69
Memory window 3 end-address low-byte	82A	2A	6A

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

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

5.1 ExCA Identification and Revision Register (Index 00h)

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

NOTE: If bit 5 (SUBSYRW) in the system control register (see Section 4.29) is 1, then this register is read-only.

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

Register: **ExCA identification and revision**
Type: Read/Write
Offset: CardBus Socket Address + 800h: Card A ExCA Offset 00h
Card B ExCA Offset 40h
Default: 84h

Table 5–2. ExCA Identification and Revision Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–6	IFTYPE	R/W	Interface type. These bits, which are hardwired as 10b, identify the 16-bit PC Card support provided by the PCI1451. The PCI1451 supports both I/O and memory 16-bit PC Cards.
5–4	RSVD	R/W	These bits can be used for 82365SL emulation.
3–0	365REV	R/W	82365SL revision. This field stores the 82365SL revision supported by the PCI1451. Host software may read this field to determine compatibility to the 82365SL register set. This field defaults to 0100b upon reset.

5.2 ExCA Interface Status Register (Index 01h)

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

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

Register: **ExCA interface status**
Type: Read-only
Offset: CardBus Socket Address + 801h: Card A ExCA Offset 01h
Card B ExCA Offset 41h
Default: 00XX XXXXb

Table 5–3. ExCA Interface Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	RSVD	R	This bit returns 0 when read. A write has no effect.
6	CARDPWR	R	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 (see Section 5.3). 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	READY	R	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	CARDWP	R	Card write protect. This bit indicates the current status of the WP signal at the PC Card interface. This signal reports to the PCI1451 whether or not the memory card is write protected. Further, write protection for an entire PCI1451 16-bit memory window is available by setting the appropriate bit in the ExCA memory window offset-address high byte register (see Section 5.18). 0 = WP signal is 0. PC Card is R/W. 1 = WP signal is 1. PC Card is read-only.
3	CDETECT2	R	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	CDETECT1	R	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	BVDSTAT	R	Battery voltage detect. When a 16-bit memory card is inserted, the field indicates the status of the battery voltage detect signals (BVD1, BVD2) at the PC Card interface, where bit 0 reflects the BVD1 status, and bit 1 reflects BVD2. 00 = Battery is dead. 01 = Battery is dead. 10 = Battery is low; warning. 11 = Battery is good. When a 16-bit I/O card is inserted, this field indicates the status of the $\overline{\text{SPKR}}$ (bit 1) signal and the $\overline{\text{STSCHG}}$ (bit 0) at the PC Card interface. In this case, the two bits in this field directly reflect the current state of these card outputs.

5.3 ExCA Power Control Register (Index 02h)

This register provides PC Card power control. Bit 7 of this register controls the 16-bit output enables on the socket interface, and can be used for power management in 16-bit PC Card applications.

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

Register: **ExCA power control**
Type: Read-only, Read/Write
Offset: CardBus Socket Address + 802h: Card A ExCA Offset 02h
Card B ExCA Offset 42h
Default: 00h

Table 5–4. ExCA Power Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	COE	R/W	Card output enable. This bit controls the state of all of the 16-bit outputs on the PCI1451. This bit is encoded as: 0 = 16-bit PC Card outputs are disabled (default). 1 = 16-bit PC Card outputs are enabled.
6–5	RSVD	R	These bits return 0s when read. Writes have no effect.
4–3	EXCAVCC	R/W	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	RSVD	R	This bit returns 0 when read. A write has no effect.
1–0	EXCAVPP	R/W	V_{PP}. These bits are used to request changes to card V_{PP}. The PCI1451 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

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

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

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

Register: **ExCA interrupt and general control**
Type: Read/Write
Offset: CardBus Socket Address + 803h: Card A ExCA Offset 03h
Card B ExCA Offset 43h
Default: 00h

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

BIT	SIGNAL	TYPE	FUNCTION
7	RINGEN	R/W	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	RESET	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	CARDTYPE	R/W	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	CSCROUTE	R/W	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 (see Section 5.6). 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	INTSELECT	R/W	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

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

This register reflects the status of PC Card CSC interrupt sources. The ExCA card status change interrupt configuration register (see Section 5.6) 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 (see Section 5.22).

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

Register: **ExCA card status-change**
Type: Read-only
Offset: CardBus Socket Address + 804h: Card A ExCA Offset 04h
Card B ExCA Offset 44h
Default: 00h

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

BIT	SIGNAL	TYPE	FUNCTION
7–4	RSVD	R	These bits return 0s when read. Writes have no effect.
3	CDCHANGE	R	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	READYCHANGE	R	Ready change. When a 16-bit memory is installed in the socket, this bit includes whether the source of a PCI1451 interrupt was due to a change on the READY signal at the PC Card interface indicating that PC Card is now ready to accept new data. A read of this bit or writing a 1 to this bit clears it. This bit is encoded as: 0 = No low-to-high transition detected on READY (default) 1 = Detected a low-to-high transition on READY When a 16-bit I/O card is installed, this bit is always 0.
1	BATWARN	R	Battery warning change. When a 16-bit memory card is installed in the socket, this bit indicates whether the source of a PCI1451 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	BATDEAD	R	Battery dead or status change. When a 16-bit memory card is installed in the socket, this bit indicates whether the source of a PCI1451 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 PCI1451 is configured for ring indicate operation this bit indicates the status of the RI pin.

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

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

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

Register: **ExCA card status-change interrupt configuration**
Type: Read/Write
Offset: CardBus Socket Address + 805h: Card A ExCA Offset 05h
Card B ExCA Offset 45h
Default: 00h

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

BIT	SIGNAL	TYPE	FUNCTION
7–4	CSCSELECT	R/W	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) (see Section 4.40) 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) (see Section 4.40) 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	CDEN	R/W	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 = Enables interrupts on CD1 or CD2 line changes
2	READYEN	R/W	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	BATWARNEN	R/W	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	BATDEADEN	R/W	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

5.7 ExCA Address Window Enable Register (Index 06h)

This register enables/disables the memory and I/O windows to the 16-bit PC Card. By default, all windows to the card are disabled. The PCI1451 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 (see Sections 5.9 through 5.20).

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

Register: **ExCA address window enable**
Type: Read-only, Read/Write
Offset: CardBus Socket Address + 806h: Card A ExCA Offset 06h
Card B ExCA Offset 46h
Default: 00h

Table 5–8. ExCA Address Window Enable Register Description

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

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

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

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

Register: **ExCA I/O window control**
Type: Read/Write
Offset: CardBus Socket Address + 807h: Card A ExCA Offset 07h
Card B ExCA Offset 47h
Default: 00h

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

BIT	SIGNAL	TYPE	FUNCTION
7	WAITSTATE1	R/W	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	ZEROWS1	R/W	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	IOSIS16W1	R/W	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 = Window data width determined by DATASIZE1, bit 4 (default) 1 = Window data width determined by IOIS16
4	DATASIZE1	R/W	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	WAITSTATE0	R/W	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	ZEROWS0	R/W	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	IOIS16W0	R/W	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 = Window data width determined by DATASIZE0, bit 0 (default) 1 = Window data width determined by IOIS16
0	DATASIZE0	R/W	I/O window 0 data size. This bit controls the I/O window 0 data size. This bit is ignored if the I/O window 0 IOIS16 Source bit (bit 1) is set. This bit is encoded as: 0 = Window data width is 8 bits (default) 1 = Window data width is 16 bits

5.9 ExCA I/O Windows 0 and 1 Start-Address Low-Byte Registers (Index 08h, 0Ch)

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

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

Register: **ExCA I/O window 0 start-address low-byte**
Offset: CardBus Socket Address + 808h: Card A ExCA Offset 08h
Card B ExCA Offset 48h
Register: **ExCA I/O window 1 start-address low-byte**
Offset: CardBus Socket Address + 80Ch: Card A ExCA Offset 0Ch
Card B ExCA Offset 4Ch
Type: Read/Write
Default: 00h
Size: One byte

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

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

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

Register: **ExCA I/O window 0 start-address high-byte**
Offset: CardBus Socket Address + 809h: Card A ExCA Offset 09h
Card B ExCA Offset 49h
Register: **ExCA I/O window 1 start-address high-byte**
Offset: CardBus Socket Address + 80Dh: Card A ExCA Offset 0Dh
Card B ExCA Offset 4Dh
Type: Read/Write
Default: 00h
Size: One byte

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

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

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

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

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

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

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

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

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

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

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

Register: **ExCA memory window 0 start-address low-byte**
Offset: CardBus Socket Address + 810h: Card A ExCA Offset 10h
Card B ExCA Offset 50h

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

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

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

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

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

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

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

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

Register: **ExCA memory window 0 start-address high-byte**
Offset: CardBus Socket Address + 811h: Card A ExCA Offset 11h
Card B ExCA Offset 51h

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

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

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

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

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

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

BIT	SIGNAL	TYPE	FUNCTION
7	DATASIZE	R/W	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	ZEROWAIT	R/W	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	SCRATCH	R/W	Scratch pad bits. These bits have no effect on memory window operation.
3–0	STAHN	R/W	Start address high-nibble. These bits represent the upper address bits A23–A20 of the memory window start address.

5.15 ExCA Memory Windows 0–4 End-Address Low-Byte Registers (Index 12h/1Ah/22h/2Ah/32h)

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

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

Register: **ExCA memory window 0 end-address low-byte**
Offset: CardBus Socket Address + 812h: Card A ExCA Offset 12h
Card B ExCA Offset 52h
Register: **ExCA memory window 1 end-address low-byte**
Offset: CardBus Socket Address + 81Ah: Card A ExCA Offset 1Ah
Card B ExCA Offset 5Ah
Register: **ExCA memory window 2 end-address low-byte**
Offset: CardBus Socket Address + 822h: Card A ExCA Offset 22h
Card B ExCA Offset 62h
Register: **ExCA memory window 3 end-address low-byte**
Offset: CardBus Socket Address + 82Ah: Card A ExCA Offset 2Ah
Card B ExCA Offset 6Ah
Register: **ExCA memory window 4 end-address low-byte**
Offset: CardBus Socket Address + 832h: Card A ExCA Offset 32h
Card B ExCA Offset 72h

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

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

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

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

Register: **ExCA memory window 0 end-address high-byte**
Offset: CardBus Socket Address + 813h: Card A ExCA Offset 13h
Card B ExCA Offset 53h
Register: **ExCA memory window 1 end-address high-byte**
Offset: CardBus Socket Address + 81Bh: Card A ExCA Offset 1Bh
Card B ExCA Offset 5Bh
Register: **ExCA memory window 2 end-address high-byte**
Offset: CardBus Socket Address + 823h: Card A ExCA Offset 23h
Card B ExCA Offset 63h
Register: **ExCA memory window 3 end-address high-byte**
Offset: CardBus Socket Address + 82Bh: Card A ExCA Offset 2Bh
Card B ExCA Offset 6Bh
Register: **ExCA Memory window 4 end-address high-byte**
Offset: CardBus Socket Address + 833h: Card A ExCA Offset 33h
Card B ExCA Offset 73h
Type: Read/Write, Read-only
Default: 00h
Size: One byte

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

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

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

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

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

Register: **ExCA memory window 0 offset-address low-byte**
Offset: CardBus Socket Address + 814h: Card A ExCA Offset 14h
Card B ExCA Offset 54h

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

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

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

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

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

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

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

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

Register: **ExCA memory window 0 offset-address high-byte**
Offset: CardBus Socket Address + 815h: Card A ExCA Offset 15h
Card B ExCA Offset 55h
Register: **ExCA memory window 1 offset-address high-byte**
Offset: CardBus Socket Address + 81Dh: Card A ExCA Offset 1Dh
Card B ExCA Offset 5Dh
Register: **ExCA memory window 2 offset-address high-byte**
Offset: CardBus Socket Address + 825h: Card A ExCA Offset 25h
Card B ExCA Offset 65h
Register: **ExCA memory window 3 offset-address high-byte**
Offset: CardBus Socket Address + 82Dh: Card A ExCA Offset 2Dh
Card B ExCA Offset 6Dh
Register: **ExCA memory window 4 offset-address high-byte**
Offset: CardBus Socket Address + 835h: Card A ExCA Offset 35h
Card B ExCA Offset 75h
Type: Read/Write
Default: 00h
Size: One byte

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

BIT	SIGNAL	TYPE	FUNCTION
7	WINWP	R/W	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	REG	R/W	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	OFFHB	R/W	Offset-address high byte. These bits represent the upper address bits A25–A20 of the memory-window offset address.

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

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

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

Register: **ExCA I/O window 0 offset-address low-byte**
Offset: CardBus Socket Address + 836h: Card A ExCA Offset 36h
Card B ExCA Offset 76h
Register: **ExCA I/O window 1 offset-address low-byte**
Offset: CardBus Socket Address + 838h: Card A ExCA Offset 38h
Card B ExCA Offset 78h
Type: Read/Write
Default: 00h
Size: One byte

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

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

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

Register: **ExCA I/O window 0 offset-address high-byte**
Offset: CardBus Socket Address + 837h: Card A ExCA Offset 37h
Card B ExCA Offset 77h
Register: **ExCA I/O window 1 offset-address high-byte**
Offset: CardBus Socket Address + 839h: Card A ExCA Offset 39h
Card B ExCA Offset 79h
Type: Read/Write
Default: 00h
Size: One byte

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

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

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

Register: **ExCA card detect and general control**
Type: Read-only, Write-only, Read/Write
Offset: CardBus Socket Address + 816h: Card A ExCA Offset 16h
Card B ExCA Offset 56h
Default: XX00 0000b

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

BIT	SIGNAL	TYPE	FUNCTION
7	VS2STAT	R	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	VS1STAT	R	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	SWCSC	W	Software card detect interrupt. If the card detect enable bit in the ExCA card status-change interrupt configuration register (see Section 5.6) 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 (see Section 5.6), 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.
4	CDRESUME	R/W	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 (see Section 5.5) 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	RSVD	R	These bits return 0s when read. Writes have no effect.
1	REGCONFIG	R/W	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	RSVD	R	This bit returns 0 when read. A write has no effect.

5.22 ExCA Global Control Register (Index 1Eh)

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

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

Register: **ExCA global control**
Type: Read-only, Read/Write
Offset: CardBus Socket Address + 81Eh: Card A ExCA Offset 1Eh
Card B ExCA Offset 5Eh
Default: 00h

Table 5–14. ExCA Global Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–5	RSVD	R	These bits return 0s when read. Writes have no effect.
4	INTMODEB	R/W	Level/edge interrupt mode select – card B. This bit selects the signaling mode for the PCI1451 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	INTMODEA	R/W	Level/edge interrupt mode select – card A. This bit selects the signaling mode for the PCI1451 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	IFCMODE	R/W	Interrupt flag clear mode select. This bit selects the interrupt flag clear mechanism for the flags in the ExCA card status change register (see Section 5.5). This bit is encoded as: 0 = Interrupt flags cleared by read of CSC register (default) 1 = Interrupt flags cleared by explicit write back of 1
1	CSCMODE	R/W	Card status change level/edge mode select. This bit selects the signaling mode for the PCI1451 host interrupt for card status changes. This bit is encoded as: 0 = Host interrupt is edge mode (default) 1 = Host interrupt is level mode
0	PWRDWN	R/W	PWRDWN mode select. When the bit is set to 1, the PCI1451 is in power-down mode. In power-down mode the PCI1451 card outputs are placed in a high-impedance state until an active cycle is executed on the card interface. Following an active cycle the outputs are again placed in a high-impedance state. The PCI1451 still receives DMA requests, functional interrupts and/or card status change interrupts; however, an actual card access is required to wake up the interface. This bit is encoded as: 0 = Power-down mode disabled (default) 1 = Power-down mode enabled

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

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

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

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

6 CardBus Socket Registers (Functions 0 and 1)

The PCMCIA CardBus Specification requires a CardBus socket controller to provide five 32-bit registers which report and control the socket-specific functions. The PCI1451 provides the CardBus socket/ExCA base address register (PCI offset 10h) to locate these CardBus socket registers in PCI memory address space. Each socket has a separate base address register for accessing the CardBus socket registers, see Figure 6–1 below. Table 6–1 illustrates the location of the socket registers in relation to the CardBus socket/ExCA base address.

Table 6–1. CardBus Socket Registers

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

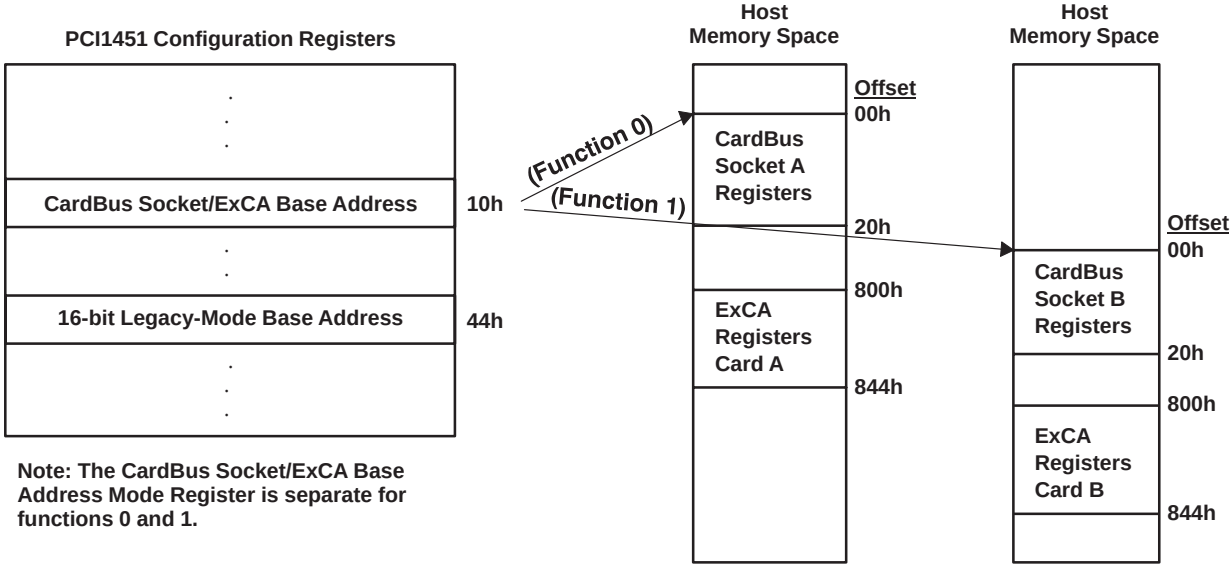


Figure 6–1. Accessing CardBus Socket Registers Through PCI Memory

6.1 Socket Event Register

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

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

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

Table 6–2. Socket Event Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–4	RSVD	R	These bits return 0s when read.
3	PWREVENT	R/WC	Power cycle. This bit is set when the PCI1451 detects that the PWRCYCLE bit in the socket present state register (see Section 6.3) has changed. This bit is cleared by writing a 1.
2	CD2EVENT	R/WC	$\overline{\text{CCD2}}$. This bit is set when the PCI1451 detects that the CDETECT2 field in the socket present state register (see Section 6.3) has changed. This bit is cleared by writing a 1.
1	CD1EVENT	R/WC	$\overline{\text{CCD1}}$. This bit is set when the PCI1451 detects that the CDETECT1 field in the socket present state register (see Section 6.3) has changed. This bit is cleared by writing a 1.
0	CSTSEVENT. CSTSCHG	R/WC	This bit is set when the CARDSTS field in the socket present state register (see Section 6.3) 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.

6.2 Socket Mask Register

This register allows software to control the CardBus card events which generate a status change interrupt. Table 6–3 below describes each bit in this register. The state of these mask bits does not prevent the corresponding bits from reacting in the socket event register (see Section 6.1).

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

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

Table 6–3. Socket Mask Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–4	RSVD	R	These bits return 0s when read.
3	PWRMASK	R/W	Power cycle. This bit masks the PWRCYCLE bit in the socket present state register (see Section 6.3) 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	CDMASK	R/W	Card detect mask. These bits mask the CDETECT1 and CDETECT2 bits in the socket present state register (see Section 6.3) 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	CSTSMASK	R/W	CSTSCHG mask. This bit masks the CARDSTS field in the socket present state register (see Section 6.3) from causing a CSC interrupt. 0 = CARDSTS event will not cause CSC interrupt (default) 1 = CARDSTS event will cause CSC interrupt

6.3 Socket Present State Register

This register reports information about the socket interface. Writes to the socket force event register (see Section 6.4) are reflected here as well as general socket interface status. Information about PC Card V_{CC} support and card type is only updated at each insertion. Also note that the PCI1451 uses the $\overline{CCD1}$ and $\overline{CCD2}$ signals during card identification, and changes on these signals during this operation are not reflected in this register.

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

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

Table 6–4. Socket Present State Register Description

BIT	SIGNAL	TYPE	FUNCTION
31	YVSOCKET	R	YV socket. This bit indicates whether or not the socket can supply V_{CC} = Y.YV to PC Cards. The PCI1451 does not support Y.YV V_{CC}; therefore, this bit is hardwired to 0.
30	XVSOCKET	R	XV socket. This bit indicates whether or not the socket can supply V_{CC} = X.XV to PC Cards. The PCI1451 does not support X.XV V_{CC}; therefore, this bit is hardwired to 0.
29	3VSOCKET	R	3-V socket. This bit indicates whether or not the socket can supply V_{CC} = 3.3 Vdc to PC Cards. The PCI1451 does support 3.3 V V_{CC}; therefore, this bit is always set unless overridden by the socket force event register (see Section 6.4).
28	5VSOCKET	R	5-V socket. This bit indicates whether or not the socket can supply V_{CC} = 5.0 Vdc to PC Cards. The PCI1451 does support 5.0 V V_{CC}; therefore, this bit is always 1 unless overridden by the device control register (bit 6) (see Section 4.39).
27–14	RSVD	R	These bits return 0s when read.
13	YVCARD	R	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 (see Section 6.4).
12	XVCARD	R	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 (see Section 6.4).
11	3VCARD	R	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 F3VCARD bit in the socket force event register (see Section 6.4).
10	5VCARD	R	5-V card. This bit indicates whether or not the PC Card inserted in the socket supports V_{CC} = 5.0 Vdc. This bit can be set by writing to the F5VCARD bit in the socket force event register (see Section 6.4).
9	BADVCCREQ	R	Bad V_{CC} request. This bit indicates that the host software has requested that the socket be powered at an invalid voltage. 0 = Normal operation (default) 1 = Invalid V_{CC} request by host software
8	DATALOST	R	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 PCI1451. 0 = Normal operation (default) 1 = Potential data loss due to card removal
7	NOTACARD	R	Not a card. This bit indicates that an unrecognizable PC Card has been inserted in the socket. This bit is not updated until a valid PC Card is inserted into the socket. 0 = Normal operation (default) 1 = Unrecognizable PC Card detected

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

BIT	SIGNAL	TYPE	FUNCTION

6	IREQCINT	R	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	CBCARD	R	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	16BITCARD	R	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	PWRCYCLE	R	Power cycle. This bit indicates the status of the card power request. This bit is encoded as: 0 = Socket is powered down (default) 1 = Socket is powered up
2	CDETECT2	R	$\overline{\text{CCD2}}$. 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	CDETECT1	R	$\overline{\text{CCD1}}$. 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	CARDSTS. CSTSCHG	R	This bit reflects the current status of the CSTSCHG signal at the PC Card interface. 0 = CSTSCHG is low 1 = CSTSCHG is high

6.4 Socket Force Event Register

This register is used to force changes to the socket event register (see Section 6.1) and the socket present state register (see Section 6.3). The CVSTEST bit in this register must be written when forcing changes that require card interrogation.

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

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

Table 6–5. Socket Force Event Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–15	RSVD	R	These bits return 0s when read.
14	CVSTEST	W	Card VS test. When this bit is set, the PCI1451 reinterrogates the PC Card, updates the socket present state register (see Section 6.3), and re-enables the socket power control.
13	FYVCARD	W	Force YV card. Writes to this bit cause the YVCARD bit in the socket present state register (see Section 6.3) to be written. When set, this bit disables the socket power control.
12	FXVCARD	W	Force XV card. Writes to this bit cause the XVCARD bit in the socket present state register (see Section 6.3) to be written. When set, this bit disables the socket power control.
11	F3VCARD	W	Force 3-V card. Writes to this bit cause the 3VCARD bit in the socket present state register (see Section 6.3) to be written. When set, this bit disables the socket power control.
10	F5VCARD	W	Force 5-V card. Writes to this bit cause the 5VCARD bit in the socket present state register (see Section 6.3) to be written. When set, this bit disables the socket power control.
9	FBADVCCREQ	W	Force BadVccReq. Changes to the BADVCCREQ bit in the socket present state register (see Section 6.3) can be made by writing this bit.
8	FDATALOST	W	Force data lost. Writes to this bit cause the DATALOST bit in the socket present state register (see Section 6.3) to be written.
7	FNOTACARD	W	Force not a card. Writes to this bit cause the NOTACARD bit in the socket present state register (see Section 6.3) to be written.
6	RSVD	R	This bit returns 0 when read.
5	FCBCARD	W	Force CardBus card. Writes to this bit cause the CBCARD bit in the socket present state register (see Section 6.3) to be written.
4	F16BITCARD	W	Force 16-bit card. Writes to this bit cause the 16BITCARD bit in the socket present state register (see Section 6.3) to be written.
3	FPWRCYCLE	W	Force power cycle. Writes to this bit cause the PWREVENT bit in the socket event register (see Section 6.1) to be written, and the PWRCYCLE bit in the socket present state register (see Section 6.3) is unaffected.
2	FCDETECT2	W	Force $\overline{\text{CCD2}}$. Writes to this bit cause the CD2EVENT bit in the socket event register (see Section 6.1) to be written, and the CDETECT2 bit in the socket present state register (see Section 6.3) is unaffected.
1	FCDETECT1	W	Force $\overline{\text{CCD1}}$. Writes to this bit cause the CD1EVENT bit in the socket event register (see Section 6.1) to be written, and the CDETECT1 bit in the socket present state register (see Section 6.3) is unaffected.
0	FCARDSTS	W	Force CSTSCHG. Writes to this bit cause the CSTSEVENT bit in the socket event register (see Section 6.1) to be written. The CARDSTS bit in the socket present state register (see Section 6.3) is unaffected.

6.5 Socket Control Register

This register provides control of the voltages applied to the socket's V_{PP} and V_{CC}. The PCI1451 ensures that the socket is powered up only at acceptable voltages when a CardBus card is inserted.

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

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

Table 6–6. Socket Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–8	RSVD	R	These bits return 0s when read.
7	STOPCLK	R/W	This bit controls how the CardBus clock run state machine decides when to stop the CardBus clock to the CardBus Card: 0 = The PCI1451 Clock run master will request 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 PCI1451 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 if the CardBus interface is sampled idle for 8 clocks.
6–4	VCCCTRL	R/W	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	RSVD	R	This bit returns 0 when read.
2–0	VPPCTRL	R/W	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

6.6 Socket Power Management Register

This register provides power management control over the socket through a mechanism for slowing or stopping the clock on the card interface when the card is idle.

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

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

Table 6–7. Socket Power Management Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–26	RSVD	R	These bits return 0s when read.
25	SKTACCES	R	Socket access status. This bit provides information on when a socket access has occurred. This bit is cleared by a read access. 0 = No PC Card access has occurred (default) 1 = PC Card has been accessed
24	SKTMODE	R	Socket mode status. This bit provides clock mode information. 0 = Normal clock operation 1 = Clock frequency has changed
23–17	RSVD	R	These bits return 0s when read.
16	CLKCTRLEN	R/W	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	RSVD	R	These bits return 0s when read.
0	CLKCTRL	R/W	CardBus clock control. This bit determines whether the CardBus $\overline{\text{CLKRUN}}$ protocol will attempt to stop or slow the CardBus clock during idle states. The CLKCTRLEN bit enables this bit. 0 = Allows the CardBus $\overline{\text{CLKRUN}}$ protocol to attempt to stop the CardBus clock (default). 1 = Allows the CardBus $\overline{\text{CLKRUN}}$ protocol to attempt to slow the CardBus clock by a factor of 16.

7 Distributed DMA (DDMA) Registers

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

These PCI1451 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 PCI1451 will implement these obsolete register bits as nonfunctional, read-only bits. The reserved registers shown in Table 7–1 are implemented as read-only, and return 0s when read. Writes to reserved registers have no effect.

Table 7–1. Distributed DMA Registers

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

7.1 DMA Current Address/Base Address Register

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

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

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

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

Register: **DMA current address/base address**
Type: Read/Write
Offset: DMA Base Address + 00h
Default: 0000h
Size: Two bytes

7.2 DMA Page Register

This register is used to set the upper byte of the address of a DMA transfer. Details of the address represented by this register are explained in the DMA current address/base address register (see Section 7.1).

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

Register: **DMA page**
Type: Read/Write
Offset: DMA Base Address + 02h
Default: 00h
Size: One byte

7.3 DMA Current Count/Base Count Register

This register is used to set the total transfer count, in bytes, of a direct memory transfer. Reads from this register indicate the current count of a direct memory transfer. In the 8-bit transfer mode, the count is decremented by 1 after each transfer. Likewise, the count is decremented by 2 in 16-bit transfer mode.

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

Register: **DMA current count/base count**
Type: Read/Write
Offset: DMA Base Address + 04h
Default: 0000h
Size: Two bytes

7.4 DMA Command Register

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

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

Register: **DMA command**
Type: Read-only, Read/Write
Offset: DMA Base Address + 08h
Default: 00h
Size: One byte

Table 7–2. DDMA Command Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–3	RSVD	R	These bits return 0s when read.
2	DMAEN	R/W	DMA controller enable. This bit enables and disables the distributed DMA slave controller in the PCI1451, and defaults to the enabled state. 0 = DMA controller enabled (default) 1 = DMA controller disabled
1–0	RSVD	R	These bits return 0s when read.

7.5 DMA Status Register

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

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

Register: **DMA status**
Type: Read-only
Offset: DMA Base Address + 08h
Default: 00h
Size: One byte

Table 7–3. DMA Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–4	DREQSTAT	R	Channel request. In the 8237, these bits indicate the status of the $\overline{\text{DREQ}}$ signal of each DMA channel. In the PCI1451, 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 (see Section 7.9) has no effect on these bits.
3–0	TC	R/C	Channel terminal count. The 8327 uses these bits to indicate the TC status of each of its four DMA channels. In the PCI1451, these bits report information about just a single DMA channel; therefore, all four of these register bits indicate the TC status of the single socket being serviced by this register. All four bits are set when the terminal count (TC) is reached by the DMA channel. These bits are reset when read or when the DMA channel is reset.

7.6 DMA Request Register

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

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

Register: **DMA request**
Type: Write-only
Offset: DMA Base Address + 09h
Default: 00h
Size: One byte

7.7 DMA Mode Register

This register is used to set the DMA transfer mode.

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

Register: **DMA mode**
Type: Read-only, Read/Write
Offset: DMA Base Address + 0Bh
Default: 00h
Size: One byte

Table 7–4. DDMA Mode Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–6	DMAMODE	R/W	Mode select bits. The PCI1451 uses these bits to determine the transfer mode. 00 = Demand mode select (default) 01 = Single mode select 10 = Block mode select 11 = Reserved
5	INCDEC	R/W	Address increment/decrement. The PCI1451 uses this register bit to select the memory address in the DMA current address/base address register (see Section 7.1) to increment or decrement after each data transfer. This is in accordance with the 8237 use of this register bit, and is encoded as follows: 0 = Addresses increment (default) 1 = Addresses decrement
4	AUTOINIT	R/W	Auto-initialization bit. 0 = Auto-initialization disabled (default) 1 = Auto-initialization enabled
3–2	XFERTYPE	R/W	Transfer type. These bits select the type of direct memory transfer to be performed. A memory write transfer moves data from the PCI1451 PC Card interface to memory, and a memory read transfer moves data from memory to the PCI1451 PC Card interface. The field is encoded as: 00 = No transfer selected (default) 01 = Write transfer 10 = Read transfer 11 = Reserved
1–0	RSVD	R	These bits return 0s when read.

7.8 DMA Master Clear Register

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

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

Register: **DMA master clear**
Type: Write-only
Offset: DMA Base Address + 0Dh
Default: 00h
Size: One byte

7.9 DMA Multichannel Mask Register

The PCI1451 uses only the least significant bit of this register to mask the PC Card DMA channel. The PCI1451 sets the mask bit when the PC Card is removed. Host software is responsible for either resetting the socket DMA controller or re-enabling the mask bit.

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

Register: **DMA multichannel mask**
Type: Read-only, Read/Write
Offset: DMA Base Address + 0Fh
Default: 00h
Size: One byte

Table 7–5. DDMA Multichannel Mask Register Description

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

8 Electrical Characteristics

8.1 Absolute Maximum Ratings Over Operating Temperature Ranges†

Supply voltage range, V_{CC} . . . . . –0.5 V to 4.6 V
Supply voltage range, V_{CCP}, V_{CCA}, V_{CCB} . . . . . –0.5 V to 6 V
Input voltage range, V_I: PCI . . . . . –0.5 V to V_{CCP} + 0.5 V
Card A . . . . . –0.5 V to V_{CCA} + 0.5 V
Card B . . . . . –0.5 V to V_{CCB} + 0.5 V
Miscellaneous . . . . . –0.5 V to V_{CC} + 0.5 V
Output voltage range, V_O: PCI . . . . . –0.5 V to V_{CC} + 0.5 V
Card A . . . . . –0.5 V to V_{CCA} + 0.5 V
Card B . . . . . –0.5 V to V_{CCB} + 0.5 V
Miscellaneous . . . . . –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 Misc 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}. The limit specified applies for a dc condition.

2. Applies for external output and bidirectional buffers. V_O > V_{CC} does not apply to Misc 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}. The limit specified applies for a dc condition.

8.2 Recommended Operating Conditions (see Note 3)

		OPERATION	MIN	NOM	MAX	UNIT
Core voltage, V_{CC}	Commercial	3.3 V	3	3.3	3.6	V
PCI I/O voltage, V_{CCP}	Commercial	3.3 V	3	3.3	3.6	V
		5 V	4.75	5	5.25	
PC Card I/O voltage, $V_{CC(A/B)}$	Commercial	3.3 V	3	3.3	3.6	V
		5 V	4.75	5	5.25	
High-level Input voltage, V_{IH}†	PCI	3.3 V	0.5 V_{CCP}		V_{CCP}	V
		5 V	2		V_{CCP}	
	PC Card	3.3 V	0.475 $V_{CC(A/B)}$		$V_{CC(A/B)}$	
		5 V	2.4		$V_{CC(A/B)}$	
	VS	3.3 V	2		V_{CC}	
	CD	3.3 V	2.4		V_{CC}	
	Miscellaneous‡		2		V_{CC}	
Low-level input voltage, V_{IL}†	PCI	3.3 V	0		0.3 V_{CCP}	V
		5 V	0		0.8	
	PC Card	3.3 V	0		0.325 $V_{CC(A/B)}$	
		5 V	0		0.8	
	Miscellaneous‡		0		0.8	
Input voltage, V_I	PCI		0		V_{CCP}	V
	PC Card		0		$V_{CC(A/B)}$	
	Miscellaneous‡		0		V_{CC}	
Output voltage, V_O¶	PCI		0		V_{CCP}	V
	PC Card		0		$V_{CC(A/B)}$	
	Miscellaneous‡		0		V_{CC}	
Input transition times (t_r and t_f), t_t	PCI and PC Card		1		4	ns
	Miscellaneous‡		0		6	
Operating ambient temperature range, T_A			0	25	70	°C
Virtual junction temperature, T_J§			0	25	115	°C

† Applies for external inputs and bidirectional buffers without hysteresis

‡ Miscellaneous terminals are $\overline{\text{RI\_OUT}}$, CLOCK, DATA, LATCH, $\overline{\text{SPKROUT}}$, SCL, SDA, $\overline{\text{SUSPEND}}$, MFUNC terminals, VS terminals, CD terminals, and ZV terminals.

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

¶ Applies for external output buffers

NOTE 3: Unused or floating terminals (input or I/O) must be held high or low.

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

PARAMETER		TERMINALS	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		
		Miscellaneous§		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	
		Miscellaneous§		I_{OL} = 4 mA		0.5	
I_{OZL}	3-state-output, high-impedance-state current (see Note 4)	Output-only terminals	3.6 V	V_I = GND		–1	µA
			5.25 V	V_I = GND		–1	
I_{OZH}	3-state-output, high-impedance-state current	Output-only terminals	3.6 V	V_I = V_{CC}¶		10	µA
			5.25 V	V_I = V_{CC}¶		25	
I_{IL}	Low-level input current	Input-only terminals		V_I = GND		–1	µA
		I/O terminals†		V_I = GND		–10	
		Pullup terminals‡		V_I = GND		–190	
I_{IH}	High-level input current (see Note 5)	Input-only terminals	3.6 V	V_I = V_{CC}¶		10	µA
			5.25 V	V_I = V_{CC}¶		20	
		I/O terminals†	3.6 V	V_I = V_{CC}¶		10	
			5.25 V	V_I = V_{CC}¶		25	

† For I/O terminals, input leakage (I_{IL} and I_{IH}) includes I_{OZ} leakage of the disabled output.
‡ Pullup terminals: A_CPERR, A_CIRDY, A_CBLOCK, A_CSTOP, A_CDEVSEL, A_CTRDY, A_CSTSCHG, A_CAUDIO, A_CCD1, A_CCD2, A_CREQ, A_CINT, A_CRST, A_CVS1, A_CVS2, A_CSERR, B_CPERR, B_CIRDY, B_CBLOCK, B_CSTOP, B_CDEVSEL, B_CTRDY, B_CSTSCHG, B_CAUDIO, B_CCD1, B_CCD2, B_CREQ, B_CINT, B_CRST, B_CVS1, B_CVS2, B_CSERR, MFUNC5, MFUNC6, and LATCH.
§ Miscellaneous terminals are RI_OUT, CLOCK, DATA, LATCH, SPKROUT, SCL, SDA, SUSPEND, MFUNC terminals, VS terminals, CD terminals, and ZV terminals.
¶ For PCI terminals, V_I = V_{CCP}. For PC Card terminals, V_I = $V_{CC(A/B)}$.
NOTES: 4. V_{OH} and I_{OL} are not tested on RI_OUTZ (pin P12) because they are open-drain outputs.
5. I_{IH} is not tested on pullup terminals because they are pulled up with an internal resistor.

8.4 PCI Clock/Reset Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature (see Figure 8–2 and Figure 8–3)

PARAMETER		ALTERNATE SYMBOL	TEST CONDITIONS	MIN	MAX	UNIT
t_c	Cycle time, PCLK	t_{cyc}		30		ns
t_{wH}	Pulse duration, PCLK high	t_{high}		11		ns
t_{wL}	Pulse duration, PCLK low	t_{low}		11		ns
$\Delta v/\Delta t$	Slew rate, PCLK	t_r, t_f		1	4	V/ns
t_w	Pulse duration, RSTIN	t_{rst}		1		ms
t_{su}	Setup time, PCLK active at end of $\overline{RSTIN}$	$t_{rst\text{-}clk}$		100		µs

8.5 PCI Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature (see Note 7, Figure 8–1 and Figure 8–4)

PARAMETER			ALTERNATE SYMBOL	TEST CONDITIONS	MIN	MAX	UNIT
t_{pd}	Propagation delay time, See Note 6	PCLK-to-shared signal valid delay time	t_{val}	C_L = 50 pF, See Note 7		11	ns
		PCLK-to-shared signal invalid delay time	t_{inv}		2		
t_{en}	Enable time, high impedance-to-active delay time from PCLK		t_{on}		2		ns
t_{dis}	Disable time, active-to-high impedance delay time from PCLK		t_{off}			28	ns
t_{su}	Setup time before PCLK valid		t_{su}		7		ns
t_h	Hold time after PCLK high		t_h		0		ns

NOTES: 6. PCI shared signals are AD31–AD0, C/$\overline{BE3}$–C/$\overline{BE0}$, $\overline{FRAME}$, $\overline{TRDY}$, $\overline{IRDY}$, $\overline{STOP}$, IDSEL, $\overline{DEVSEL}$, 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.

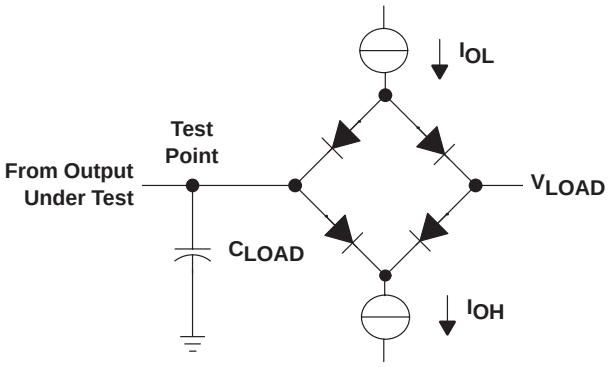
8.6 Parameter Measurement Information

LOAD CIRCUIT PARAMETERS

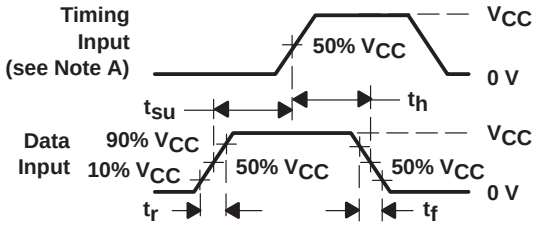
TIMING PARAMETER		C_{LOAD}† (pF)	I_{OL} (mA)	I_{OH} (mA)	V_{LOAD}‡ (V)
t_{en}	t_{PZH}	50	8	–8	0
	t_{PZL}				3
t_{dis}	t_{PHZ}	50	8	–8	1.5
	t_{PLZ}				
t_{pd}		50	8	–8	‡

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

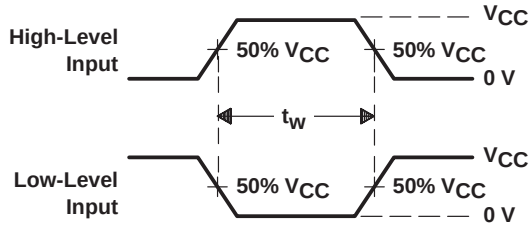
‡ $\frac{V_{LOAD} - V_{OL}}{I_{OL}}$ = 50 Ω, where V_{OL} = 0.6 V, I_{OL} = 8 mA



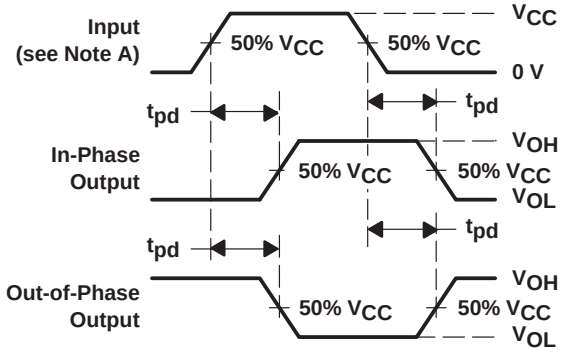
LOAD CIRCUIT



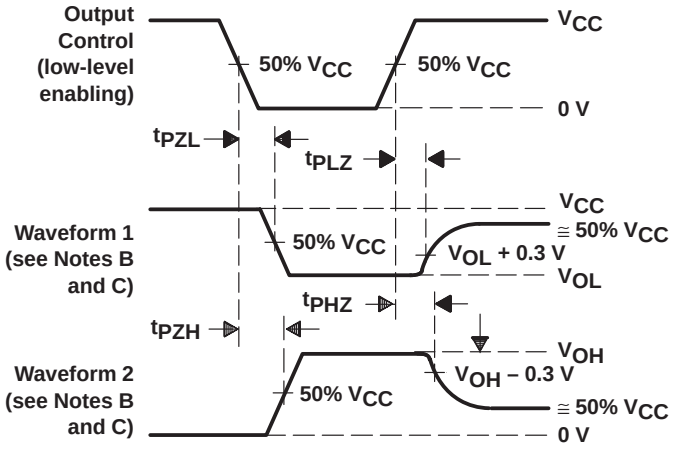
VOLTAGE WAVEFORMS
SETUP AND HOLD TIMES
INPUT RISE AND FALL TIMES



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 8–1. Load Circuit and Voltage Waveforms

8.7 PCI Bus Parameter Measurement Information

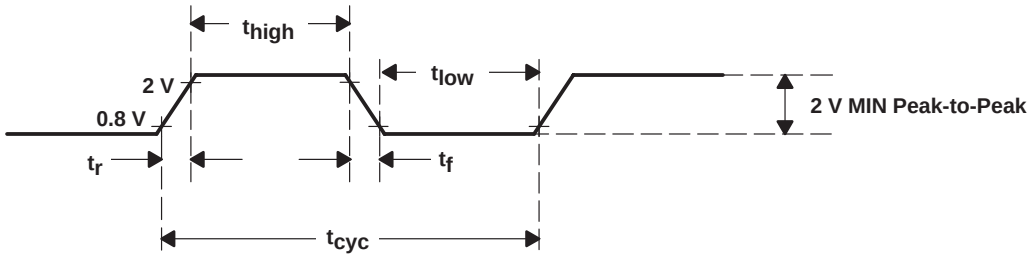


Figure 8–2. PCLK Timing Waveform

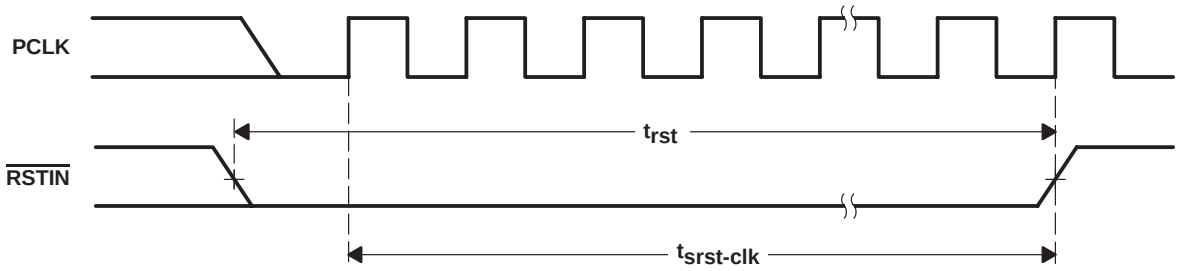


Figure 8–3. $\overline{RSTIN}$ Timing Waveforms

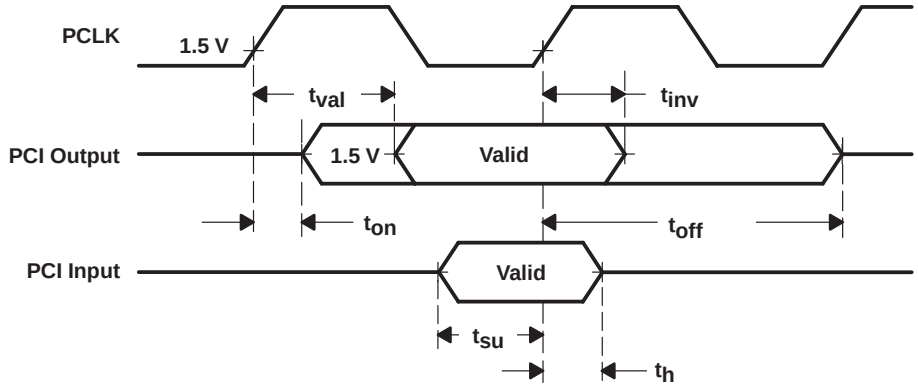


Figure 8–4. Shared Signals Timing Waveforms

8.8 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 8–1 shows address setup time in PCLK cycles and nanoseconds for I/O and memory cycles. Table 8–2 and Table 8–3 show command active time in PCLK cycles and nanoseconds for I/O and memory cycles. Table 8–4 shows address hold time in PCLK cycles and nanoseconds for I/O and memory cycles.

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

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

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

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

Table 8–3. PC Card Command Active Time, $t_{c(A)}$, 16-Bit PCI Cycles

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

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

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

8.9 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 8–5)

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

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

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

(see Figure 8–6)

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

8.11 Switching Characteristics Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature, Miscellaneous (see Figure 8–7)

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

8.12 PC Card Parameter Measurement Information

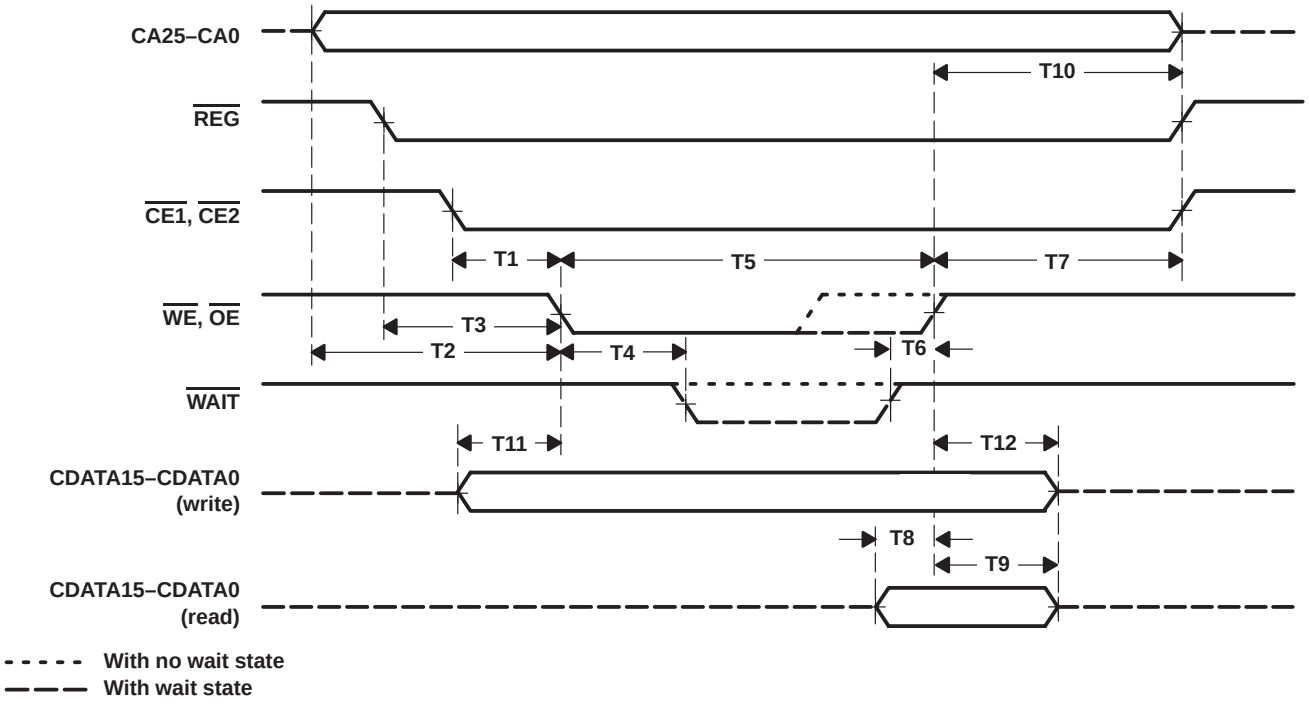


Figure 8–5. PC Card Memory Cycle

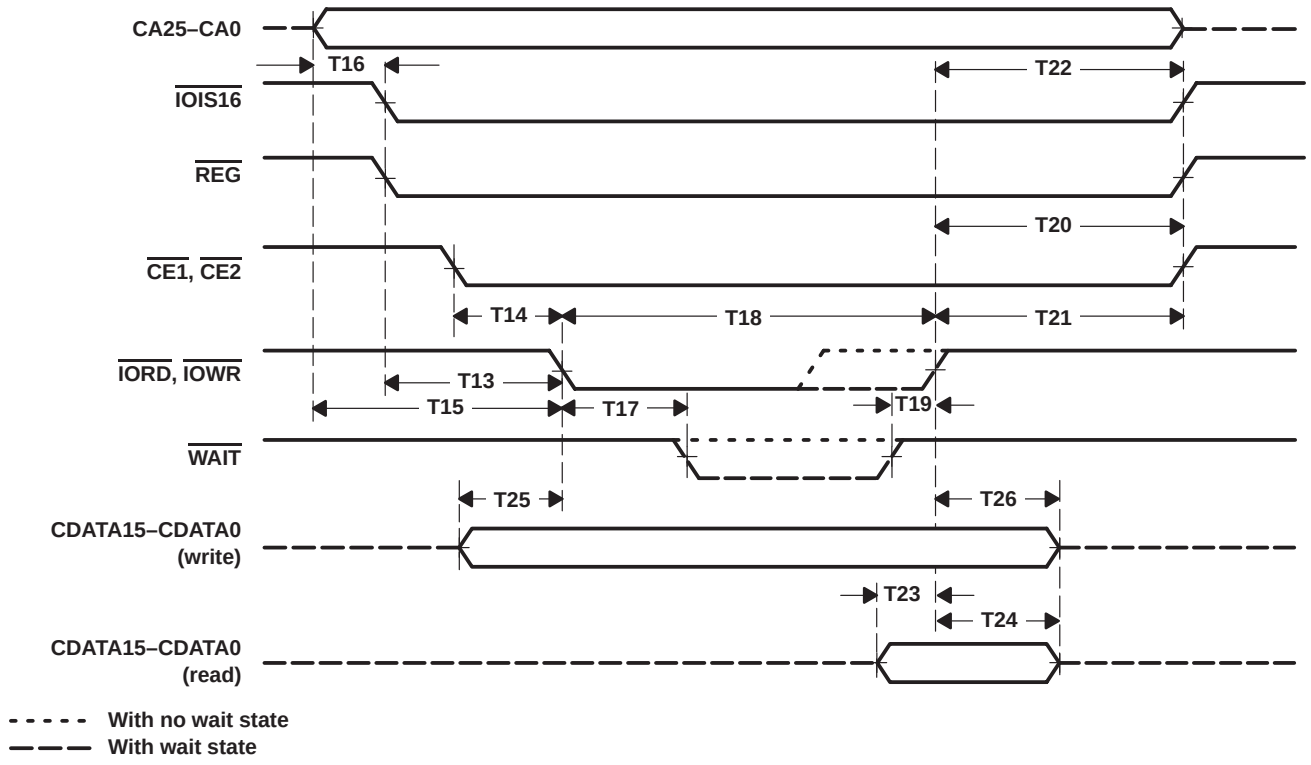


Figure 8–6. PC Card I/O Cycle

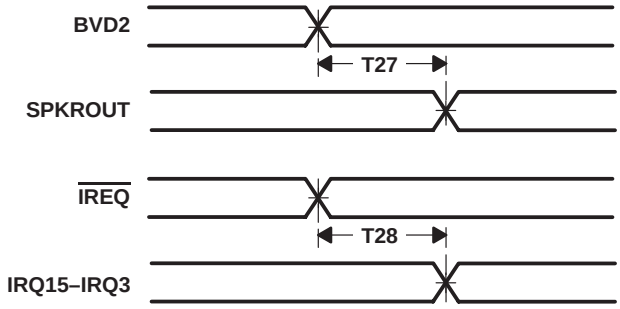
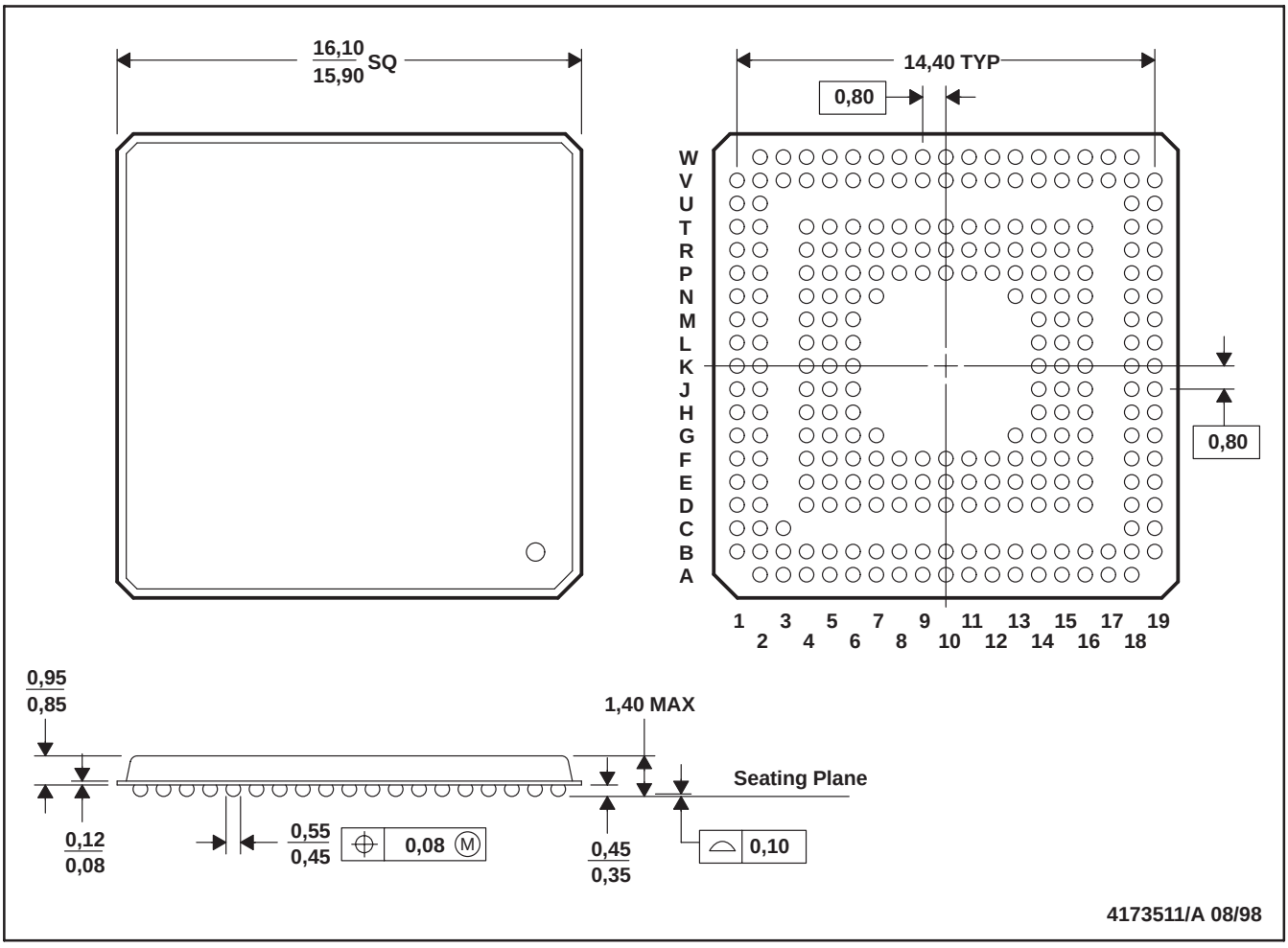


Figure 8–7. Miscellaneous PC Card Delay Times

9 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

MicroStar is a trademark of Texas Instruments Incorporated.

IMPORTANT NOTICE

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

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

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

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

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