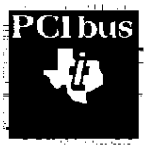
PCI1251A GFN/GJG
PC CARD CONTROLLER

- (PCI) Power Management 1.0 Compliant
- ACPI 1.0 Compliant
- Packaged in GFN 256-Pin BGA or GJG MicroStar BGA
- PCI Local Bus Specification Revision 2.2 Compliant
- 1997 PC Card™ Standard Compliant
- PC 99 Compliant
- 3.3-V Core Logic With Universal PCI Interfaces Compatible With 3.3-V and 5-V PCI Signaling Environments
- Mix-and-Match 5-V/3.3-V PC Card16 Cards and 3.3-V CardBus Cards
- Supports Two PC Card or CardBus Slots With Hot Insertion and Removal
- Uses Serial Interface to TI™ TPS2206 Dual Power Switch
- Supports Burst Transfers to Maximize Data Throughput on the PCI Bus and the CardBus Bus
- Supports Serialized Interrupt request (IRQ) With PCI Interrupts
- 8-Way Legacy IRQ Multiplexing
- System Interrupts Can Be Programmed as PCI Style or Industry Standard Architecture (ISA-IRQ) Style
- ISA-IRQ Interrupts Can Be Serialized Onto a Single IRQ Serial (IRQSER) Pin
- EEPROM Interface for Loading Subsystem ID and Subsystem Vendor ID
- Pipelined Architecture Allows Greater Than 130M-Bytes-Per-Second Throughput From CardBus to PCI and From PCI to CardBus
- Supports Zoom Video With Internal Buffering
- Four General Purpose I/Os
- Multifunction PCI Device With Separate Configuration Space for Each Socket
- Five PCI Memory Windows and Two I/O Windows Available for Each PC Card16 Socket
- Two I/O Windows and Two Memory Windows Available to Each CardBus Socket
- Exchangeable Card Architecture (ExCA) Compatible Registers Are Mappable in Memory and I/O Space
- Supports Distributed DMA (DDMA) and PC/PCI DMA
- Intel™ 82365SL-DF Register Compatible
- Supports 16-Bit DMA on Both PC Card Sockets
- Supports Ring Indicate, $\overline{\text{SUSPEND}}$, PCI $\overline{\text{CLKRUN}}$, and CardBus $\overline{\text{CCLKRUN}}$
- Advanced Submicron, Low-Power CMOS Technology
- Provides VGA/Palette Memory and I/O and Subtractive Decoding Options
- LED Activity Pins
- Supports PCI Bus Lock ($\overline{\text{LOCK}}$)

Table of Contents

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



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

description

The TI PCI1251A 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. The PCI1251A provides a rich feature set that makes it the best choice for bridging between PCI and PC Cards in both notebook and desktop computers. The 1997 PC Card Standard retains the 16-bit PC Card specification defined in PCMCIA Release 2.2, and defines the new 32-bit PC Card, CardBus, capable of full 32-bit data transfers at 33 MHz. The PCI1251A supports any combination of 16-bit and CardBus PC Cards in the two sockets, powered at 5 V or 3.3 V, as required.

The PCI1251A is compliant with the latest *PCI Bus Power Management Specification*. It is also compliant with the PCI Local Bus Specification 2.2, 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 direct memory access (DMA) transfers or CardBus PC Card bridging transactions.

Multiple system-interrupt signaling options are provided and they include:

- Parallel PCI interrupts
- Parallel ISA interrupts
- Serialized ISA interrupts
- Serialized ISA and PCI interrupts

Additionally, general-purpose inputs and outputs are provided for the board designer to implement sideband functions.

All card signals are internally buffered to allow hot insertion and removal without external buffering. The PCI1251A is register compatible with the Intel 82365SL-DF ExCA controller. The PCI1251A 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 PCI1251A can also be programmed to accept fast posted writes to improve system-bus utilization.

The PCI1251A provides an internally buffered zoom video (ZV) path. This reduces the design effort of PC board manufacturers to add a ZV-compatible solution and guarantees compliance with the CardBus loading specifications. Many other features, such as socket activity light-emitting diode (LED) outputs, are designed into the PCI1251A. These features are discussed in detail throughout the design specification.

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

Unused PCI1251A inputs must be pulled up using a 43 kΩ resistor.

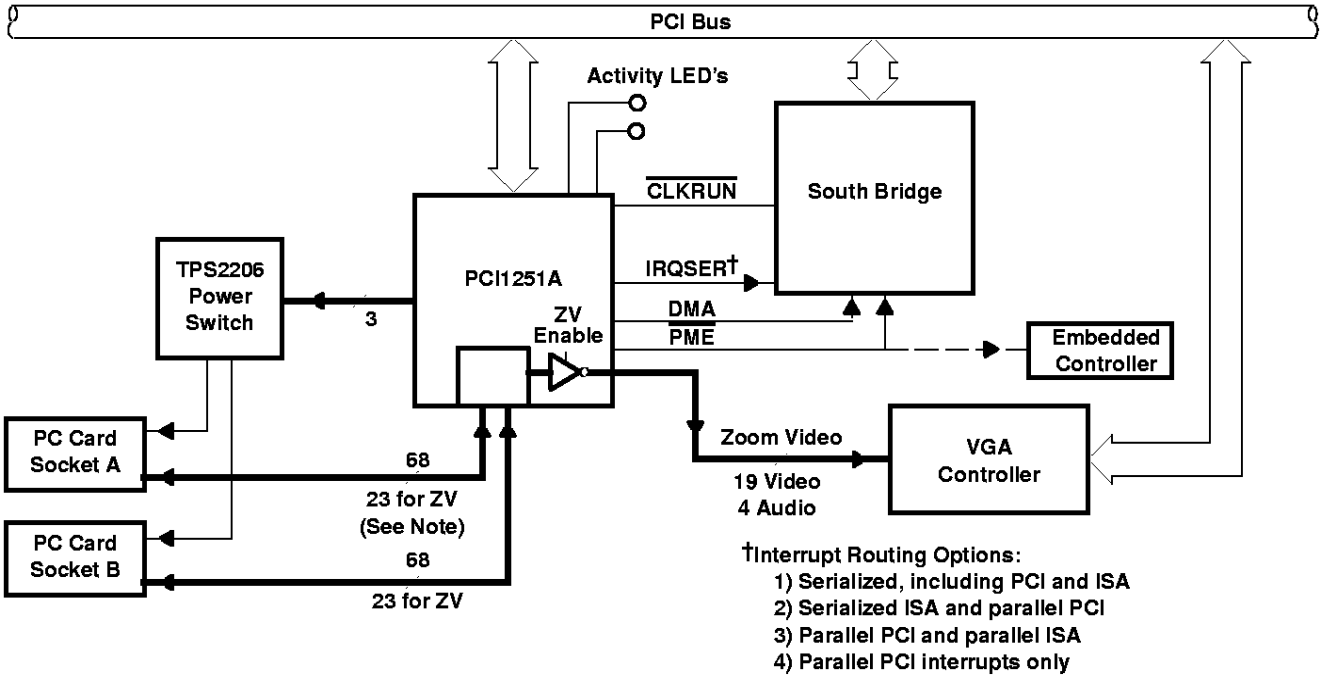
system block diagram

A simplified system block diagram using the PCI1251A is provided below. The zoomed video (ZV) capability can be used to route the ZV data directly to the VGA controller.

The PCI interface includes all address/data and control signals for PCI protocol. The 68-pin PC Card interface includes all address/data and control signals for CardBus and 16-bit (R2) protocols. When zoomed video (ZV) is enabled (in 16-bit PC Card mode), 23 of the 68 signals are redefined to support the ZV protocol.

The interrupt interface includes terminals for parallel PCI, parallel ISA, and serialized PCI and ISA signaling. The ring indicate terminal is included in the interrupt interface because its function is to perform system wake up on incoming PC Card modem rings. Other miscellaneous system interface terminals available on the PCI1251A include:

- Programmable general purpose multifunction terminals
- $\overline{\text{SUSPEND}}$, $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ (power management control signal)
- SPKROUT



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.

signal names and terminal assignments

Signal names and their terminal assignments are shown in Tables 1 through 4 sorted alpha-numerically by the assigned terminal.

Table 1. GFN Terminals Sorted Alpha-Numerically for CardBus and 16-Bit Signals

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

signal names and terminal assignments (continued)

Table 1. GFN Terminals Sorted Alpha-Numerically for CardBus and 16-Bit Signals (Continued)

GFN	SIGNAL NAME	GFN	SIGNAL NAME	GFN	SIGNAL NAME
N4	GND	U9	IRQMUX1	W6	A_$\overline{CCD2}$//A_$\overline{CD2}$
N17	GND	U10	V_{CC}	W7	A_CAD29//A_D1
N18	AD22	U11	$\overline{PCGNT}$/IRQMUX6	W8	A_CAD31//A_D10
N19	AD23	U12	CLOCK	W9	IRQMUX3
N20	IDSEL	U13	GND	W10	IRQMUX5
P1	A_$\overline{CBLOCK}$//A_A19	U14	AD6	W11	GPIO1/LEDA2
P2	A_$\overline{CPERR}$//A_A14	U15	V_{CC}	W12	LATCH
P3	A_$\overline{CGNT}$//A_$\overline{WE}$	U16	AD11	W13	IRQSER/$\overline{INTB}$
P4	A_$\overline{CTRDY}$//A_A22	U17	GND	W14	AD1
P17	AD17	U18	$\overline{PERR}$	W15	AD4
P18	V_{CCP}	U19	$\overline{SERR}$	W16	AD7
P19	AD20	U20	$\overline{TRDY}$	W17	AD9
P20	AD21	V1	A_CAD18//A_A7	W18	AD13
R1	A_$\overline{CSTOP}$//A_A20	V2	A_CAD20//A_A6	W19	C/$\overline{BE1}$
R2	A_$\overline{CDEVSEL}$//A_A21	V3	A_CAD21//A_A5	W20	V_{CCP}
R3	V_{CCA}	V4	A_CAD25//A_A1	Y1	A_$\overline{CREQ}$//A_$\overline{INPACK}$
R4	V_{CC}	V5	A_$\overline{CSERR}$//A_$\overline{WAIT}$	Y2	A_CC/$\overline{BE3}$//A_$\overline{REG}$
R17	V_{CC}	V6	A_CSTSCHG//A_BVD1($\overline{STSCHG}$/$\overline{RI}$)	Y3	A_CVS1//A_$\overline{VS1}$
R18	AD16	V7	A_CAD28//A_D8	Y4	A_$\overline{CINT}$//A_READY($\overline{IREQ}$)
R19	AD18	V8	A_RSVD//A_D2	Y5	A_CAUDIO//A_BVD2($\overline{SPKR}$)
R20	AD19	V9	IRQMUX2	Y6	A_CAD27//A_D0
T1	A_CCLK//A_A16	V10	V_{CCI}	Y7	A_CAD30//A_D9
T2	A_$\overline{CIRDY}$//A_A15	V11	GPIO0/LEDA1	Y8	IRQMUX0
T3	A_CC/$\overline{BE2}$//A_A12	V12	DATA	Y9	IRQMUX4
T4	A_CAD19//A_A25	V13	GPIO3/$\overline{INTA}$	Y10	SPKROUT
T17	$\overline{STOP}$	V14	AD3	Y11	$\overline{SUSPEND}$
T18	$\overline{IRDY}$	V15	V_{CCP}	Y12	$\overline{PCREQ}$/IRQMUX7
T19	$\overline{FRAME}$	V16	AD8	Y13	$\overline{RI\_OUT}$/$\overline{PME}$
T20	C/$\overline{BE2}$	V17	AD12	Y14	AD0
U1	A_$\overline{CFRAME}$//A_23	V18	AD15	Y15	AD2
U2	A_CAD17//A_A24	V19	GPIO2/$\overline{LOCK}$	Y16	AD5
U3	A_CVS2//A_$\overline{VS2}$	V20	$\overline{DEVSEL}$	Y17	C/$\overline{BE0}$
U4	GND	W1	A_$\overline{CRST}$//A_RESET	Y18	AD10
U5	A_CAD26//A_A0	W2	A_CAD22//A_A4	Y19	AD14
U6	V_{CC}	W3	A_CAD23//A_A3	Y20	PAR
U7	A_$\overline{CCLKRUN}$//A_WP($\overline{IOIS16}$)	W4	A_CAD24//A_A2		
U8	GND	W5	V_{CCA}		

signal names and terminal assignments (continued)

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

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

signal names and terminal assignments (continued)

Table 2. GFN Terminals Sorted Alpha-Numerically for CardBus and 16-bit Signals. (Continued)

NO.	SIGNAL NAME	NO.	SIGNAL NAME	NO.	SIGNAL NAME
M16	$\overline{\text{CBE3}}$	R6	A_CSTSCHG//A_BVD1($\overline{\text{STSCHG}}$/RI)	V3	A_CAD25//A_A1
M18	IDSEL	R7	A_CAD28//A_D8	V4	A_CVS1//A_$\overline{\text{VS1}}$
M19	AD23	R8	IRQMUX2	V5	A_CAUDIO//A_BVD2($\overline{\text{SPKR}}$)
N1	A_$\overline{\text{CDEVSEL}}$//A_A21	R9	IRQMUX5	V6	GND
N2	GND	R10	$\overline{\text{PCGNT}}$/IRQMUX6	V7	A_CAD29//A_D1
N4	A_CCLK//A_A16	R11	CLOCK	V8	IRQMUX0
N5	A_$\overline{\text{CTRDY}}$//A_A22	R12	AD0	V9	GND
N6	V_{CCA}	R13	GND	V10	GPI01/LEDA2
N7	A_$\overline{\text{CGNT}}$//A_$\overline{\text{WE}}$	R14	$\overline{\text{C/BE0}}$	V11	LATCH
N13	AD1	R15	V_{CC}	V12	V_{CC}
N14	GND	R16	$\overline{\text{TRDY}}$	V13	AD3
N15	AD19	R18	$\overline{\text{FRAME}}$	V14	V_{CCP}
N16	AD21	R19	$\overline{\text{IRDY}}$	V15	AD10
N18	V_{CCP}	T1	A_CAD20//A_A6	V16	AD13
N19	AD20	T2	A_$\overline{\text{CRST}}$//A_RESET	V17	C/$\overline{\text{BE1}}$
P1	A_$\overline{\text{CIRDY}}$//A_A15	T4	A_CAD21//A_A5	V18	V_{CCP}
P2	A_$\overline{\text{CFRAME}}$//A_23	T5	A_$\overline{\text{CINT}}$//A_READY($\overline{\text{IREQ}}$)	V19	GPI02/$\overline{\text{LOCK}}$
P4	A_CC/$\overline{\text{BE2}}$//A_A12	T6	A_$\overline{\text{CCLKRUN}}$//A_WP($\overline{\text{IOIS16}}$)	W2	A_CC/$\overline{\text{BE3}}$//A_$\overline{\text{REG}}$
P5	V_{CC}	T7	A_RSVD//A_D2	W3	A_CAD24//A_A2
P6	A_CAD17//A_A24	T8	IRQMUX1	W4	A_CAD26//A_A0
P7	A_CAD27//A_D0	T9	IRQMUX3	W5	V_{CCA}
P8	V_{CC}	T10	GPIO0/LEDA1	W6	A_$\overline{\text{CCD2}}$//A_$\overline{\text{CD2}}$
P9	V_{CCI}	T11	DATA	W7	A_CAD30//A_D9
P10	$\overline{\text{SPKROUT}}$	T12	GPI03/$\overline{\text{INTA}}$	W8	A_CAD31//A_D10
P11	$\overline{\text{PCREQ}}$/IRQMUX7	T13	AD4	W9	IRQMUX4
P12	$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	T14	AD7	W10	$\overline{\text{SUSPEND}}$
P13	AD5	T15	AD11	W11	GND
P14	AD8	T16	AD15	W12	IRQSER/$\overline{\text{INTB}}$
P15	C/$\overline{\text{BE2}}$	T18	$\overline{\text{DEVSEL}}$	W13	AD2
P16	AD16	T19	$\overline{\text{STOP}}$	W14	AD6
P18	AD18	U1	GND	W15	AD9
P19	AD17	U2	A_CAD22//A_A4	W16	AD12
R1	A_CAD18//A_A7	U18	$\overline{\text{PERR}}$	W17	AD14
R2	A_CAD19//A_A25	U19	$\overline{\text{SERR}}$	W18	PAR
R4	A_CVS2//A_$\overline{\text{VS2}}$	V1	A_$\overline{\text{CREQ}}$//A_$\overline{\text{INPACK}}$		
R5	A_$\overline{\text{CSERR}}$//A_$\overline{\text{WAIT}}$	V2	A_CAD23//A_A3		

signal names and terminal assignments (continued)

Table 3. CardBus Signal Names Sorted Alpha-Numerically for GFN and GJG Pins

SIGNAL NAME	GFN	GJG	SIGNAL NAME	GFN	GJG	SIGNAL NAME	GFN	GJG
A_CAD0	G2	F2	A_CCLK	T1	N4	B_CAD20	A14	A13
A_CAD1	H3	G5	A_$\overline{\text{CCLKRUN}}$	U7	T6	B_CAD21	B13	A12
A_CAD2	G1	G6	A_$\overline{\text{CDEVSEL}}$	R2	N1	B_CAD22	A13	B12
A_CAD3	H1	G4	A_$\overline{\text{FRAME}}$	U1	P2	B_CAD23	C12	E12
A_CAD4	H2	G1	A_$\overline{\text{CGNT}}$	P3	N7	B_CAD24	A12	A11
A_CAD5	J3	H1	A_$\overline{\text{CINT}}$	Y4	T5	B_CAD25	B11	B11
A_CAD6	J4	H6	A_$\overline{\text{CIRDY}}$	T2	P1	B_CAD26	C11	E11
A_CAD7	J1	H4	A_$\overline{\text{CPAR}}$	N3	M1	B_CAD27	D9	D9
A_CAD8	K3	J1	A_$\overline{\text{CPERR}}$	P2	M4	B_CAD28	A8	E9
A_CAD9	L1	J5	A_$\overline{\text{CREQ}}$	Y1	V1	B_CAD29	B8	F8
A_CAD10	L2	K6	A_$\overline{\text{CRST}}$	W1	T2	B_CAD30	C8	B8
A_CAD11	L3	K1	A_$\overline{\text{CSERR}}$	V5	R5	B_CAD31	B7	E8
A_CAD12	M1	K4	A_$\overline{\text{CSTOP}}$	R1	M5	B_CAUDIO	D10	D10
A_CAD13	L4	K2	A_CSTSCHG	V6	R6	B_$\overline{\text{CBLOCK}}$	B18	A18
A_CAD14	M3	L1	A_$\overline{\text{CTRDY}}$	P4	N5	B_$\overline{\text{CCBE0}}$	D20	F16
A_CAD15	M2	L6	A_CVS1	Y3	V4	B_$\overline{\text{CCBE1}}$	B19	C18
A_CAD16	M4	L2	A_CVS2	U3	R4	B_$\overline{\text{CCBE2}}$	D14	D14
A_CAD17	U2	P6	A_RSVD	J2	H2	B_$\overline{\text{CCBE3}}$	B12	D11
A_CAD18	V1	R1	A_RSVD	N2	L5	B_CCD1	H20	H14
A_CAD19	T4	R2	A_RSVD	V8	T7	B_CCD2	C9	B9
A_CAD20	V2	T1	B_CAD00	H19	H19	B_CCLK	A17	D15
A_CAD21	V3	T4	B_CAD01	G20	H16	B_$\overline{\text{CCLKRUN}}$	B9	A9
A_CAD22	W2	U2	B_CAD02	H18	H18	B_$\overline{\text{CDEVSEL}}$	A18	B16
A_CAD23	W3	V2	B_CAD03	F20	G13	B_$\overline{\text{CFRAME}}$	C15	E14
A_CAD24	W4	W3	B_CAD04	G19	H15	B_$\overline{\text{CGNT}}$	D16	A16
A_CAD25	V4	V3	B_CAD05	F19	G18	B_$\overline{\text{CINT}}$	A10	F10
A_CAD26	U5	W4	B_CAD06	G18	G19	B_$\overline{\text{CIRDY}}$	A16	B15
A_CAD27	Y6	P7	B_CAD07	G17	G14	B_CPAR	A19	B18
A_CAD28	V7	R7	B_CAD08	E19	F18	B_$\overline{\text{CPERR}}$	B17	A17
A_CAD29	W7	V7	B_CAD09	E18	F15	B_$\overline{\text{CREQ}}$	D12	D12
A_CAD30	Y7	W7	B_CAD10	D19	E19	B_$\overline{\text{CRST}}$	C13	D13
A_CAD31	W8	W8	B_CAD11	C20	E18	B_$\overline{\text{CSERR}}$	B10	A10
A_CAUDIO	Y5	V5	B_CAD12	D18	D19	B_$\overline{\text{CSTOP}}$	C17	B17
A_$\overline{\text{CBLOCK}}$	P1	M2	B_CAD13	E17	E16	B_CSTSCHG	A9	E10
A_CC/$\overline{\text{BE0}}$	K1	J2	B_CAD14	B20	C19	B_$\overline{\text{CTRDY}}$	C16	E15
A_CC/$\overline{\text{BE1}}$	N1	L4	B_CAD15	C19	D18	B_CVS1	A11	F11
A_CC/$\overline{\text{BE2}}$	T3	P4	B_CAD16	C18	D16	B_CVS2	B14	B13
A_CC/$\overline{\text{BE3}}$	Y2	W2	B_CAD17	B15	A14	B_RSVD	A20	B19
A_CCD1	G3	F1	B_CAD18	A15	F13	B_RSVD	A7	D8
A_CCD2	W6	W6	B_CAD19	C14	E13	B_RSVD	E20	G15

signal names and terminal assignments (continued)

Table 4. 16 Bit Signal Names Sorted Alpha-Numerically for GFN and GJG Pins

SIGNAL NAME	GFN	GJG	SIGNAL NAME	GFN	GJG	SIGNAL NAME	GFN	GJG
A_A00	U5	W4	A_D08	V7	R7	B_A20	C17	B17
A_A01	V4	V3	A_D09	Y7	W7	B_A21	A18	B16
A_A02	W4	W3	A_D10	W8	W8	B_A22	C16	E15
A_A03	W3	V2	A_D11	G1	G6	B_A23	C15	E14
A_A04	W2	U2	A_D12	H2	G1	B_A24	B15	A14
A_A05	V3	T4	A_D13	J4	H6	B_A25	C14	E13
A_A06	V2	T1	A_D14	J2	H2	B_BVD1	A9	E10
A_A07	V1	R1	A_D15	K3	J1	B_BVD2	D10	D10
A_A08	N1	L4	A_INPACK	Y1	V1	B_$\overline{\text{CD1}}$	H20	H14
A_A09	M3	L1	A_$\overline{\text{IORD}}$	L4	K2	B_$\overline{\text{CD2}}$	C9	B9
A_A10	L1	J5	A_$\overline{\text{IOWR}}$	M2	L6	B_$\overline{\text{CE1}}$	D20	F16
A_A11	M1	K4	A_$\overline{\text{OE}}$	L3	K1	B_$\overline{\text{CE2}}$	D19	E19
A_A12	T3	P4	A_RDY	Y4	T5	B_D00	D9	D9
A_A13	N3	M1	A_$\overline{\text{REG}}$	Y2	W2	B_D01	B8	F8
A_A14	P2	M4	A_RESET	W1	T2	B_D02	A7	D8
A_A15	T2	P1	A_$\overline{\text{VS1}}$	Y3	V4	B_D03	H19	H19
A_A16	T1	N4	A_$\overline{\text{VS2}}$	U3	R4	B_D04	G20	H16
A_A17	M4	L2	A_$\overline{\text{WAIT}}$	V5	R5	B_D05	F20	G13
A_A18	N2	L5	A_$\overline{\text{WE}}$	P3	N7	B_D06	F19	G18
A_A19	P1	M2	A_WP	U7	T6	B_D07	G17	G14
A_A20	R1	M5	B_A00	C11	E11	B_D08	A8	E9
A_A21	R2	N1	B_A01	B11	B11	B_D09	C8	B8
A_A22	P4	N5	B_A02	A12	A11	B_D10	B7	E8
A_A23	U1	P2	B_A03	C12	E12	B_D11	H18	H18
A_A24	U2	P6	B_A04	A13	B12	B_D12	G19	H15
A_A25	T4	R2	B_A05	B13	A12	B_D13	G18	G19
A_BVD1	V6	R6	B_A06	A14	A13	B_D14	E20	G15
A_BVD2	Y5	V5	B_A07	A15	F13	B_D15	E19	F18
A_$\overline{\text{CD1}}$	G3	F1	B_A08	B19	C18	B_INPACK	D12	D12
A_$\overline{\text{CD2}}$	W6	W6	B_A09	B20	C19	B_$\overline{\text{IORD}}$	E17	E16
A_$\overline{\text{CE1}}$	K1	J2	B_A10	E18	F15	B_$\overline{\text{IOWR}}$	C19	D18
A_$\overline{\text{CE2}}$	L2	K6	B_A11	D18	D19	B_$\overline{\text{OE}}$	C20	E18
A_D00	Y6	P7	B_A12	D14	D14	B_RDY	A10	F10
A_D01	W7	V7	B_A13	A19	B18	B_$\overline{\text{REG}}$	B12	D11
A_D02	V8	T7	B_A14	B17	A17	B_RESET	C13	D13
A_D03	G2	F2	B_A15	A16	B15	B_$\overline{\text{VS1}}$	A11	F11
A_D04	H3	G5	B_A16	A17	D15	B_$\overline{\text{VS2}}$	B14	B13
A_D05	H1	G4	B_A17	C18	D16	B_$\overline{\text{WAIT}}$	B10	A10
A_D06	J3	H1	B_A18	A20	B19	B_$\overline{\text{WE}}$	D16	A16
A_D07	J1	H4	B_A19	B18	A18	B_WP	B9	A9

Terminal Functions

The terminals are grouped in tables by functionality, such as PCI system function, power-supply function, etc. The terminal numbers are also listed for convenient reference.

Table 5. Power Supply

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

Table 6. PC Card power switch

TERMINAL			I/O TYPE	FUNCTION
NAME	GFN NO.	GJG NO.		
CLOCK	U12	R11	I/O	3-line power switch clock. Information on the DATA line is sampled at the rising edge of CLOCK. CLOCK defaults to an input, but can be changed to a PCI1251A output by using the P2CCLK bit in the system control register. The TPS2206 defines the maximum frequency of this signal to be 2 MHz. If a system design defines this terminal as an output, CLOCK requires an external pulldown resistor. The frequency of the PCI1251A output CLOCK is derived by dividing the PCI CLK by 36.
DATA	V12	T11	O	3-line power switch data. DATA is used to serially communicate socket power-control information to the power switch.
LATCH	W12	V11	O	3-line power switch latch. LATCH is asserted by the PCI1251A to indicate to the PC Card power switch that the data on the DATA line is valid.

Terminal Functions (Continued)

Table 7. PCI system

TERMINAL NAME	TERMINAL GFN NO.	TERMINAL GJG NO.	I/O TYPE	FUNCTION
$\overline{\text{CLKRUN}}$	J18	J16	I/O	PCI clock run. $\overline{\text{CLKRUN}}$ is used by the central resource to request permission to stop the PCI clock or to slow it down, and the PCI1251A responds accordingly. If $\overline{\text{CLKRUN}}$ is not implemented, this pin should be tied low. $\overline{\text{CLKRUN}}$ is enabled by default by bit 1 (KEEPCLK) in the system control register.
PCLK	J17	J15	I	PCI bus clock. PCLK provides timing for all transactions on the PCI bus. All PCI signals are sampled at the rising edge of PCLK.
$\overline{\text{PRST}}$	J19	J18	I	PCI reset. When the PCI bus reset is asserted, $\overline{\text{PRST}}$ causes the PCI1251A 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 PCI1251A is in its default state. When the $\overline{\text{SUSPEND}}$ mode is enabled, the device is protected from the $\overline{\text{PRST}}$, and the internal registers are preserved. All outputs are placed in a high-impedance state, but the contents of the registers are preserved.

Terminal Functions (Continued)

Table 8. PCI Address and Data

TERMINAL NAME	TERMINAL GFN NO.	TERMINAL GJG NO.	I/O TYPE	FUNCTION
AD31	K18	K16	I/O	PCI address/data bus. These signals make up the multiplexed PCI address and data bus on the primary interface. During the address phase of a primary bus PCI cycle, AD31–AD0 contain a 32-bit address or other destination information. During the data phase, AD31–AD0 contain data.
AD30	K19	K18		
AD29	L20	L14		
AD28	L18	L15		
AD27	L19	L18		
AD26	M20	L19		
AD25	M19	L16		
AD24	M18	M15		
AD23	N19	M19		
AD22	N18	M14		
AD21	P20	N16		
AD20	P19	N19		
AD19	R20	N15		
AD18	R19	P18		
AD17	P17	P19		
AD16	R18	P16		
AD15	V18	T16		
AD14	Y19	W17		
AD13	W18	V16		
AD12	V17	W16		
AD11	U16	T15		
AD10	Y18	V15		
AD9	W17	W15		
AD8	V16	P14		
AD7	W16	T14		
AD6	U14	W14		
AD5	Y16	P13		
AD4	W15	T13		
AD3	V14	V13		
AD2	Y15	W13		
AD1	W14	N13		
AD0	Y14	R12		
C/$\overline{BE3}$	M17	M16	I/O	PCI bus commands and byte enables. These signals are multiplexed on the same PCI terminals. During the address phase of a primary bus PCI cycle, C/$\overline{BE3}$–C/$\overline{BE0}$ define the bus command. During the data phase, this 4-bit bus is used as byte enables. The byte enables determine which byte paths of the full 32-bit data bus carry meaningful data. C/$\overline{BE0}$ applies to byte 0 (AD7–AD0), C/$\overline{BE1}$ applies to byte 1 (AD15–AD8), C/$\overline{BE2}$ applies to byte 2 (AD23–AD16), and C/$\overline{BE3}$ applies to byte 3 (AD31–AD24).
C/$\overline{BE2}$	T20	P15		
C/$\overline{BE1}$	W19	V17		
C/$\overline{BE0}$	Y17	R14		
PAR	Y20	W18	I/O	PCI bus parity. In all PCI bus read and write cycles, the PCI1251A calculates even parity across the AD31–AD0 and C/$\overline{BE3}$–C/$\overline{BE0}$ buses. As an initiator during PCI cycles, the PCI1251A outputs this parity indicator with a one-PCLK delay. As a target during PCI cycles, the calculated parity is compared to the initiator's parity indicator. A compare error results in the assertion of a parity error ($\overline{PERR}$).

Terminal Functions (Continued)

Table 9. PCI Interface Control

TERMINAL NAME	GFN NO.	GJG NO.	I/O TYPE	FUNCTION
$\overline{\text{DEVSEL}}$	V20	T18	I/O	PCI device select. The PCI1251A asserts $\overline{\text{DEVSEL}}$ to claim a PCI cycle as the target device. As a PCI initiator on the bus, the PCI1251A monitors $\overline{\text{DEVSEL}}$ until a target responds. If no target responds before timeout occurs, the PCI1251A terminates the cycle with an initiator abort.
$\overline{\text{FRAME}}$	T19	R18	I/O	PCI cycle frame. $\overline{\text{FRAME}}$ is driven by the initiator of a bus cycle. $\overline{\text{FRAME}}$ is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When $\overline{\text{FRAME}}$ is deasserted, the PCI bus transaction is in the final data phase.
$\overline{\text{GNT}}$	J20	J14	I	PCI bus grant. $\overline{\text{GNT}}$ is driven by the PCI bus arbiter to grant the PCI1251A access to the PCI bus after the current data transaction has completed. $\overline{\text{GNT}}$ may or may not follow a PCI bus request, depending on the PCI bus parking algorithm.
GPIO2/$\overline{\text{LOCK}}$	V19	V19	I/O	PCI bus general-purpose I/O pins or PCI bus lock. GPIO2/$\overline{\text{LOCK}}$ can be configured as PCI $\overline{\text{LOCK}}$ and used to gain exclusive access downstream. Since this functionality is not typically used, a general-purpose I/O may be accessed through this terminal. GPIO2/$\overline{\text{LOCK}}$ defaults to a general-purpose input and can be configured through the GPIO2 control register.
IDSEL	N20	M18	I	Initialization device select. IDSEL selects the PCI1251A during configuration space accesses. IDSEL can be connected to one of the upper 24 PCI address lines on the PCI bus.
$\overline{\text{IRDY}}$	T18	R19	I/O	PCI initiator ready. $\overline{\text{IRDY}}$ indicates the PCI bus initiator's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCLK where both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. Until $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are both sampled asserted, wait states are inserted.
$\overline{\text{PERR}}$	U18	U18	I/O	PCI parity error indicator. $\overline{\text{PERR}}$ is driven by a PCI device to indicate that calculated parity does not match PAR when PERR is enabled through bit 6 of the command register.
$\overline{\text{REQ}}$	K17	K15	O	PCI bus request. $\overline{\text{REQ}}$ is asserted by the PCI1251A to request access to the PCI bus as an initiator.
$\overline{\text{SERR}}$	U19	U19	O	PCI system error. $\overline{\text{SERR}}$ is an output that is pulsed from the PCI1251A when enabled through the command register, indicating a system error has occurred. The PCI1251A need not be the target of the PCI cycle to assert this signal. When $\overline{\text{SERR}}$ is enabled in the bridge control register, this signal also pulses, indicating that an address parity error has occurred on a CardBus interface.
$\overline{\text{STOP}}$	T17	T19	I/O	PCI cycle stop signal. $\overline{\text{STOP}}$ is driven by a PCI target to request the initiator to stop the current PCI bus transaction. $\overline{\text{STOP}}$ is used for target disconnects and is commonly asserted by target devices that do not support burst data transfers.
$\overline{\text{TRDY}}$	U20	R16	I/O	PCI target ready. $\overline{\text{TRDY}}$ indicates the primary bus target's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCLK when both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. Until both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted, wait states are inserted.

Terminal Functions (Continued)

Table 10. System Interrupt

TERMINAL NAME	TERMINAL GFN NO.	TERMINAL GJG NO.	I/O TYPE	FUNCTION
GPIO3/$\overline{\text{INTA}}$	V13	T12	I/O	Parallel PCI interrupt. $\overline{\text{INTA}}$ can be optionally mapped to GPI03 when parallel PCI interrupts are used. See *interrupt subsystem* on page 32 for details on interrupt signaling. GPIO3/$\overline{\text{INTA}}$ defaults to a general-purpose input.
IRQSER/$\overline{\text{INTB}}$	W13	W12	I/O	Serial interrupt signal. IRQSER provides the IRQSER-style serial interrupting scheme. Serialized PCI interrupts can also be sent in the IRQSER stream. See *interrupt subsystem* on page 32 for details on interrupt signaling. This terminal can be used to signal PCI $\overline{\text{INTB}}$ when one of the parallel interrupt modes is selected in the device control register.
IRQMUX7 IRQMUX6 IRQMUX5 IRQMUX4 IRQMUX3 IRQMUX2 IRQMUX1 IRQMUX0	Y12 U11 W10 Y9 W9 V9 U9 Y8	P11 R10 R9 W9 T9 R8 T8 V8	O	Interrupt request/secondary functions multiplexed. The primary function of these terminals is to provide the ISA-type IRQ signaling supported by the PCI1251A. These interrupt multiplexer outputs can be mapped to any of 15 IRQs. The device control register must be programmed for the ISA IRQ interrupt mode and the IRQMUX routing register must have the IRQ routing programmed before these terminals are enabled. All of these terminals have secondary functions, such as PCI $\overline{\text{INTB}}$, PC/PCI DMA request/grant, ring indicate output, and zoom video status, that can be selected with the appropriate programming of this register. When the secondary functions are enabled, the respective terminals are not available for IRQ routing. See the IRQMUX routing register for programming options.
$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	Y13	P12	O	Ring Indicate Out and Power Management Event Output. Terminal provides an output for ring-indicate or $\overline{\text{PME}}$ signals.

Table 11. PC/PCI DMA

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

Terminal Functions (Continued)

Table 12. Zoom Video

TERMINAL NAME	TERMINAL GFN NO.	TERMINAL GJG NO.	I/O AND MEMORY INTERFACE SIGNAL	I/O TYPE	FUNCTION
ZV_HREF	A6	G7	A10	O	Horizontal sync to the zoom video port
ZV_VSYNC	C7	A7	A11	O	Vertical sync to the zoom video port
ZV_Y7 ZV_Y6 ZV_Y5 ZV_Y4 ZV_Y3 ZV_Y2 ZV_Y1 ZV_Y0	A3 B4 C5 B5 C6 D7 A5 B6	B5 A5 E6 B6 A6 F7 E7 B7	A20 A14 A19 A13 A18 A8 A17 A9	O	Video data to the zoom video port in YV:4:2:2 format
ZV_UV7 ZV_UV6 ZV_UV5 ZV_UV4 ZV_UV3 ZV_UV2 ZV_UV1 ZV_UV0	D2 C3 B1 B2 A2 C4 B3 D5	C1 A2 B2 A3 B3 A4 B4 D5	A25 A12 A24 A15 A23 A16 A22 A21	O	Video data to the zoom video port in YV:4:2:2 format
ZV_SCLK	C2	B1	A7	O	Audio SCLK PCM
ZV_MCLK	D3	C2	A6	O	Audio MCLK PCM
ZV_PCLK	E1	E2	$\overline{\text{IOIS16}}$	O	Pixel clock to the zoom video port
ZV_LRCLK	E3	E5	$\overline{\text{INPACK}}$	O	Audio LRCLK PCM
ZV_SDATA	E2	E1	$\overline{\text{SPKR}}$	O	Audio SDATA PCM
NC	F1	F4		O	Reserved. No connection.
ZV_RSVD ZV_RSVD	C1 E4	D2 D1	A5 A4	O	Reserved. No connection in the PC Card. ZV_RSVD1 and ZV_RSVD0 are put into the high-impedance state by host adapter.

Terminal Functions (Continued)

Table 13. Miscellaneous

TERMINAL NAME	TERMINAL GFN NO.	TERMINAL GJG NO.	I/O TYPE	FUNCTION
GPIO0/LEDA1	V11	T10	I/O	GPIO0/socket activity LED indicator 1. When GPIO0/LEDA1 is configured as LEDA1, it provides an output indicating PC Card socket 0 activity. Otherwise, GPIO0/LEDA1 can be configured as a general-purpose input and output, GPIO0. The zoom video enable signal (ZV_STAT) can also be routed to this signal through the GPIO0 control register. GPIO0/LEDA1 defaults to a general-purpose input.
GPIO1/LEDA2	W11	V10	I/O	GPIO1/socket activity LED indicator 2. When GPIO1/LEDA2 is configured as LEDA2, it provides an output indicating PC Card socket 1 activity. Otherwise, GPIO1/LEDA2 can be configured as a general-purpose input and output, GPIO1. A CSC interrupt can be generated on a GPDATA change, and this input can be used for power switch overcurrent (OC) sensing. See *GPIO1 control register* for programming details. GPIO1/LEDA2 defaults to a general-purpose input.
SPKROUT	Y10	P10	O	Speaker output. SPKROUT is the output to the host system that can carry $\overline{\text{SPKR}}$ or CAUDIO through the PCI1251A from the PC Card interface. SPKROUT is driven as the exclusive-OR combination of card $\overline{\text{SPKR}}$//CAUDIO inputs.
$\overline{\text{SUSPEND}}$	Y11	W10	I	Suspend. $\overline{\text{SUSPEND}}$ is used to protect the internal registers from clearing when $\overline{\text{PRST}}$ is asserted. See *$\overline{\text{SUSPEND}}$ mode* for details.

Terminal Functions (Continued)

Table 14. 16-bit PC Card Address and Data (slots A and B)

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

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

Terminal Functions (Continued)

Table 15. 16-bit PC Card Interface Control (slots A and B)

NAME	TERMINAL GFN NO. SLOT A†	TERMINAL GFN NO. SLOT B‡	TERMINAL GJG NO. SLOT A†	TERMINAL GJG NO. SLOT B‡	I/O TYPE	FUNCTION
BVD1 (STSCHG/RI)	V6	A9	R6	E10	I	Battery voltage detect 1. BVD1 is generated by 16-bit memory PC Cards that include batteries. BVD1 is used with BVD2 as an indication of the condition of the batteries on a memory PC Card. Both BVD1 and BVD2 are kept high when the battery is good. When BVD2 is low and BVD1 is high, the battery is weak and should be replaced. When BVD1 is low, the battery is no longer serviceable and the data in the memory PC Card is lost. See ExCA card status-change interrupt configuration register on page 84 for enable bits. See ExCA card status-change register and the ExCA interface status register on page 80 for the status bits for this signal. Status change. STSCHG is used to alert the system to a change in the READY, write protect, or battery voltage dead condition of a 16-bit I/O PC Card. Ring indicate. RI is used by 16-bit modem cards to indicate a ring detection.
BVD2 (SPKR)	Y5	D10	V5	D10	I	Battery voltage detect 2. BVD2 is generated by 16-bit memory PC Cards that include batteries. BVD2 is used with BVD1 as an indication of the condition of the batteries on a memory PC Card. Both BVD1 and BVD2 are high when the battery is good. When BVD2 is low and BVD1 is high, the battery is weak and should be replaced. When BVD1 is low, the battery is no longer serviceable and the data in the memory PC Card is lost. See ExCA card status-change interrupt configuration register on page 84 for enable bits. See ExCA card status-change register on page 83 and the ExCA interface status register on page 80 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 PCI1251A and are output on SPKROUT. DMA request. BVD2 can be used as the DMA request signal during DMA operations to a 16-bit PC Card that supports DMA. The PC Card asserts BVD2 to indicate a request for a DMA operation.
CD1 CD2	G3 W6	H20 C9	F1 W6	H14 B9	I	PC Card detect 1 and PC Card detect 2. CD1 and CD2 are internally connected to ground on the PC Card. When a PC Card is inserted into a socket, CD1 and CD2 are pulled low. For signal status, see ExCA interface status register on page 80.
CE1 CE2	K1 L2	D20 D19	J2 K6	F16 E19	O	Card enable 1 and card enable 2. CE1 and CE2 enable even- and odd-numbered address bytes. CE1 enables even-numbered address bytes, and CE2 enables odd-numbered address bytes.
INPACK	Y1	D12	V1	D12	I	Input acknowledge. INPACK is asserted by the PC Card when it can respond to an I/O read cycle at the current address. DMA request. INPACK can be used as the DMA request signal during DMA operations from a 16-bit PC Card that supports DMA. If used as a strobe, the PC Card asserts this signal to indicate a request for a DMA operation.
IORD	L4	E17	K2	E16	O	I/O read. IORD is asserted by the PCI1251A 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 PCI1251A asserts IORD during DMA transfers from the PC Card to host memory.
IOWR	M2	C19	L6	D18	O	I/O write. IOWR is driven low by the PCI1251A 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 PCI1251A asserts IOWR during transfers from host memory to the PC Card.

Terminal Functions (Continued)

Table 15. 16-bit PC Card Interface Control (slots A and B) (continued)

NAME	TERMINAL GFN NO. SLOT A†	TERMINAL GFN NO. SLOT B‡	TERMINAL GJG NO. SLOT A†	TERMINAL GJG NO. SLOT B‡	I/O TYPE	FUNCTION
$\overline{OE}$	L3	C20	K1	E18	O	Output enable. $\overline{OE}$ is driven low by the PCI1251A to enable 16-bit memory PC Card data output during host memory read cycles. DMA terminal count. $\overline{OE}$ is used as terminal count (TC) during DMA operations to a 16-bit PC Card that supports DMA. The PCI1251A asserts $\overline{OE}$ to indicate TC for a DMA write operation.
READY ($\overline{IREQ}$)	Y4	A10	T5	F10	I	Ready. The ready function is provided by READY when the 16-bit PC Card and the host socket are configured for the memory-only interface. READY is driven low by the 16-bit memory PC Cards to indicate that the memory card circuits are busy processing a previous write command. READY is driven high when the 16-bit memory PC Card is ready to accept a new data transfer command. Interrupt request. $\overline{IREQ}$ is asserted by a 16-bit I/O PC Card to indicate to the host that a device on the 16-bit I /O PC Card requires service by the host software. $\overline{IREQ}$ is high (deasserted) when no interrupt is requested.
$\overline{REG}$	Y2	B12	W2	D11	O	Attribute memory select. $\overline{REG}$ remains high for all common memory accesses. When $\overline{REG}$ is asserted, access is limited to attribute memory ($\overline{OE}$ or $\overline{WE}$ active) and to the I/O space ($\overline{IORD}$ or $\overline{IOWR}$ active). Attribute memory is a separately accessed section of card memory and is generally used to record card capacity and other configuration and attribute information. DMA acknowledge. $\overline{REG}$ is used as a DMA acknowledge ($\overline{DACK}$) during DMA operations to a 16-bit PC Card that supports DMA. The PCI1251A asserts $\overline{REG}$ to indicate a DMA operation. $\overline{REG}$ is used in conjunction with the DMA read ($\overline{IOWR}$) or DMA write ($\overline{IORD}$) strobes to transfer data.
RESET	W1	C13	T2	D13	O	PC Card reset. RESET forces a hard reset to a 16-bit PC Card.
$\overline{WAIT}$	V5	B10	R5	A10	I	Bus cycle wait. $\overline{WAIT}$ is driven by a 16-bit PC Card to delay the completion of (i.e., extend) the memory or I/O cycle in progress.
$\overline{WE}$	P3	D16	N7	A16	O	Write enable. $\overline{WE}$ is used to strobe memory write data into 16-bit memory PC Cards. $\overline{WE}$ is also used for memory PC Cards that employ programmable memory technologies. DMA terminal count. $\overline{WE}$ is used as TC during DMA operations to a 16-bit PC Card that supports DMA. The PCI1251A asserts $\overline{WE}$ to indicate TC for a DMA read operation.
WP ($\overline{IOIS16}$)	U7	B9	T6	A9	I	Write protect. WP applies to 16-bit memory PC Cards. WP reflects the status of the write-protect switch on 16-bit memory PC Cards. For 16-bit I/O cards, WP is used for the 16-bit port ($\overline{IOIS16}$) function. I/O is 16 bits. $\overline{IOIS16}$ applies to 16-bit I/O PC Cards. $\overline{IOIS16}$ is asserted by the 16-bit PC Card when the address on the bus corresponds to an address to which the 16-bit PC Card responds, and the I/O port that is addressed is capable of 16-bit accesses. DMA request. WP can be used as the DMA request signal during DMA operations to a 16-bit PC Card that supports DMA. If used, the PC Card asserts WP to indicate a request for a DMA operation.
$\overline{VS1}$ $\overline{VS2}$	Y3 U3	A11 B14	V4 R4	F11 B13	I/O	Voltage sense 1 and voltage sense 2. $\overline{VS1}$ and $\overline{VS2}$, when used in conjunction with each other, determine the operating voltage of the 16-bit PC Card.

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

Terminal Functions (Continued)

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

NAME	TERMINAL GFN NO. SLOT A†	TERMINAL GFN NO. SLOT B‡	TERMINAL GJG NO. SLOT A†	TERMINAL GJG NO. SLOT B‡	I/O TYPE	FUNCTION
CCLK	T1	A17	N4	D15	O	CardBus PC Card clock. CCLK provides synchronous timing for all transactions on the CardBus interface. All signals except $\overline{\text{CRST}}$, $\overline{\text{CCLKRUN}}$, $\overline{\text{CINT}}$, CSTSCHG, CAUDIO, $\overline{\text{CCD2:1}}$, and CVS2–CVS1 are sampled on the rising edge of CCLK, and all timing parameters are defined with the rising edge of this signal. CCLK operates at the PCI bus clock frequency, but it can be stopped in the low state or slowed down for power savings.
$\overline{\text{CCLKRUN}}$	U7	B9	T6	A9	O	CardBus PC Card clock run. $\overline{\text{CCLKRUN}}$ is used by a CardBus PC Card to request an increase in the CCLK frequency, and by the PCI1251A to indicate that the CCLK frequency is decreased. CardBus clock run ($\overline{\text{CCLKRUN}}$) follows the PCI clock run ($\overline{\text{CLKRUN}}$)
$\overline{\text{CRST}}$	W1	C13	T2	D13	I/O	CardBus PC Card reset. $\overline{\text{CRST}}$ is used to bring CardBus PC Card-specific registers, sequencers, and signals to a known state. When $\overline{\text{CRST}}$ is asserted, all CardBus PC Card signals must be 3-stated, and the PCI1251A drives these signals to a valid logic level. Assertion can be asynchronous to CCLK, but deassertion must be synchronous to CCLK.

† Terminal name for slot A is preceded with A_. For example, the full name for terminal T1 is A_CCLK.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminal A17 is B_CCLK.

Terminal Functions (Continued)

Table 17. CardBus PC Card Address and Data (slots A and B)

NAME	TERMINAL GFN NO. SLOT A†	TERMINAL GFN NO. SLOT B‡	TERMINAL GJG NO. SLOT A†	TERMINAL GJG NO. SLOT B‡	I/O TYPE	FUNCTION
CAD31	W8	B7	W8	E8	I/O	PC Card address and data. These signals make up the multiplexed CardBus address and data bus on the CardBus interface. During the address phase of a CardBus cycle, CAD31–CAD0 contain a 32-bit address. During the data phase of a CardBus cycle, CAD31–CAD0 contain data. CAD31 is the most-significant bit.
CAD30	Y7	C8	W7	B8		
CAD29	W7	B8	V7	F8		
CAD28	V7	A8	R7	E9		
CAD27	Y6	D9	P7	D9		
CAD26	U5	C11	W4	E11		
CAD25	V4	B11	V3	B11		
CAD24	W4	A12	W3	A11		
CAD23	W3	C12	V2	E12		
CAD22	W2	A13	U2	B12		
CAD21	V3	B13	T4	A12		
CAD20	V2	A14	T1	A13		
CAD19	T4	C14	R2	E13		
CAD18	V1	A15	R1	F13		
CAD17	U2	B15	P6	A14		
CAD16	M4	C18	L2	D16		
CAD15	M2	C19	L6	D18		
CAD14	M3	B20	L1	C19		
CAD13	L4	E17	K2	E16		
CAD12	M1	D18	K4	D19		
CAD11	L3	C20	K1	E18		
CAD10	L2	D19	K6	E19		
CAD9	L1	E18	J5	F15		
CAD8	K3	E19	J1	F18		
CAD7	J1	G17	H4	G14		
CAD6	J4	G18	H6	G19		
CAD5	J3	F19	H1	G18		
CAD4	H2	G19	G1	H15		
CAD3	H1	F20	G4	G13		
CAD2	G1	H18	G6	H18		
CAD1	H3	G20	G5	H16		
CAD0	G2	H19	F2	H19		
CC/$\overline{\text{BE3}}$	Y2	B12	W2	D11	I/O	CardBus bus commands and byte enables. CC/$\overline{\text{BE3}}$–CC/$\overline{\text{BE0}}$ are multiplexed on the same CardBus terminals. During the address phase of a CardBus cycle, CC/$\overline{\text{BE3}}$–CC/$\overline{\text{BE0}}$ defines the bus command. During the data phase, this 4-bit bus is used as byte enables. The byte enables determine which byte paths of the full 32-bit data bus carry meaningful data. CC/$\overline{\text{BE0}}$ applies to byte 0 (CAD7–CAD0), CC/$\overline{\text{BE1}}$ applies to byte 1 (CAD15–CAD8), CC/$\overline{\text{BE2}}$ applies to byte 2 (CAD23–CAD8), and CC/$\overline{\text{BE3}}$ applies to byte 3 (CAD31–CAD24).
CC/$\overline{\text{BE2}}$	T3	D14	P4	D14		
CC/$\overline{\text{BE1}}$	N1	B19	L4	C18		
CC/$\overline{\text{BE0}}$	K1	D20	J2	F16		
CPAR	N3	A19	M1	B18	I/O	CardBus parity. In all CardBus read and write cycles, the PCI1251A calculates even parity across the CAD and CC/$\overline{\text{BE}}$ buses. As an initiator during CardBus cycles, the PCI1251A outputs CPAR with a one-CCLK delay. As a target during CardBus cycles, the calculated parity is compared to the initiator's parity indicator; a compare error results in a parity error assertion.

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

Terminal Functions (Continued)

Table 18. CardBus PC Card Interface Control (slots A and B)

NAME	TERMINAL GFN NO. SLOT A†	TERMINAL GFN NO. SLOT B‡	TERMINAL GJG NO. SLOT A†	TERMINAL GJG NO. SLOT B‡	I/O TYPE	FUNCTION
CAUDIO	Y5	D10	V5	D10	I	CardBus audio. CAUDIO is a digital input signal from a PC Card to the system speaker. The PCI1251A supports the binary audio mode and outputs a binary signal from the card to SPKROUT.
$\overline{\text{CBLOCK}}$	P1	B18	M2	A18	I/O	CardBus lock. $\overline{\text{CBLOCK}}$ is used to gain exclusive access to a target.
$\overline{\text{CCD1}}$ $\overline{\text{CCD2}}$	G3 W6	H20 C9	F1 W6	H14 B9	I	CardBus detect 1 and CardBus detect 2. $\overline{\text{CCD1}}$ and $\overline{\text{CCD2}}$ are used in conjunction with CVS1 and CVS2 to identify card insertion and interrogate cards to determine the operating voltage and card type.
$\overline{\text{CDEVSEL}}$	R2	A18	N1	B16	I/O	CardBus device select. The PCI1251A asserts $\overline{\text{CDEVSEL}}$ to claim a CardBus cycle as the target device. As a CardBus initiator on the bus, the PCI1251A monitors $\overline{\text{CDEVSEL}}$ until a target responds. If no target responds before timeout occurs, the PCI1251A terminates the cycle with an initiator abort.
$\overline{\text{CFRAME}}$	U1	C15	P2	E14	I/O	CardBus cycle frame. $\overline{\text{CFRAME}}$ is driven by the initiator of a CardBus bus cycle. $\overline{\text{CFRAME}}$ is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When $\overline{\text{CFRAME}}$ is deasserted, the CardBus bus transaction is in the final data phase.
$\overline{\text{CGNT}}$	P3	D16	N7	A16	I	CardBus bus grant. $\overline{\text{CGNT}}$ is driven by the PCI1251A to grant a CardBus PC Card access to the CardBus bus after the current data transaction has been completed.
$\overline{\text{CINT}}$	Y4	A10	T5	F10	I	CardBus interrupt. $\overline{\text{CINT}}$ is asserted low by a CardBus PC Card to request interrupt servicing from the host.
$\overline{\text{CIRDY}}$	T2	A16	P1	B15	I/O	CardBus initiator ready. $\overline{\text{CIRDY}}$ indicates the CardBus initiator's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of CCLK when both $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are asserted. Until $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are both sampled asserted, wait states are inserted.
$\overline{\text{CPERR}}$	P2	B17	M4	A17	I/O	CardBus parity error. $\overline{\text{CPERR}}$ is used to report parity errors during CardBus transactions, except during special cycles. It is driven low by a target two clocks following that data when a parity error is detected.
$\overline{\text{CREQ}}$	Y1	D12	V1	D12	I	CardBus request. $\overline{\text{CREQ}}$ indicates to the arbiter that the CardBus PC Card desires use of the CardBus bus as an initiator.
$\overline{\text{CSERR}}$	V5	B10	R5	A10	I	CardBus system error. $\overline{\text{CSERR}}$ reports address parity errors and other system errors that could lead to catastrophic results. $\overline{\text{CSERR}}$ is driven by the card synchronous to CCLK, but deasserted by a weak pullup, and may take several CCLK periods. The PCI1251A can report $\overline{\text{CSERR}}$ to the system by assertion of $\overline{\text{SERR}}$ on the PCI interface.
$\overline{\text{CSTOP}}$	R1	C17	M5	B17	I/O	CardBus stop. $\overline{\text{CSTOP}}$ is driven by a CardBus target to request the initiator to stop the current CardBus transaction. $\overline{\text{CSTOP}}$ is used for target disconnects, and is commonly asserted by target devices that do not support burst data transfers.
CSTSCHG	V6	A9	R6	E10	I	CardBus status change. CSTSCHG is used to alert the system to a change in the card's status and is used as a wake-up mechanism.
$\overline{\text{CTRDY}}$	P4	C16	N5	E15	I/O	CardBus target ready. $\overline{\text{CTRDY}}$ indicates the CardBus target's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of CCLK, when both $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are asserted; until this time, wait states are inserted.
CVS1 CVS2	Y3 U3	A11 B14	V4 R4	F11 B13	I/O	CardBus voltage sense 1 and CardBus voltage sense 2. CVS1 and CVS2 are used in conjunction with $\overline{\text{CCD1}}$ and $\overline{\text{CCD2}}$ to identify card insertion and interrogate cards to determine the operating voltage and card type.

† Terminal name for slot A is preceded with A_. For example, the full name for terminal Y5 is A_CAUDIO.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminal D10 is B_CAUDIO.

power supply sequencing

The PCI1251A contains 3.3-V I/O buffers with 5-V tolerance requiring a core power supply and clamp voltage. The core power supply is always 3.3 V. The clamp voltage can be either 3.3 V or 5 V, depending on the interface. The following power-up and power-down sequences are recommended.

The power-up sequence is:

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

The power-down sequence is:

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

I/O characteristics

Figure 1 shows a 3-state bidirectional buffer. The *recommended operating conditions* table, on page 109, provides the electrical characteristics of the inputs and outputs.

NOTE:
The PCI1251A meets the ac specifications of the 1997 PC Card Standard and PCI Local Bus Specification Rev. 2.2.

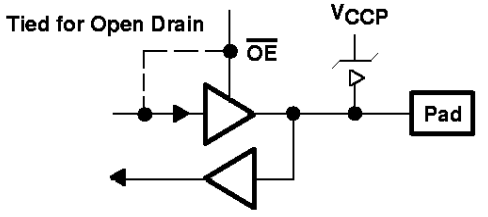


Figure 1. 3-State Bidirectional Buffer

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

clamping rail voltages

The clamping voltages are set to match whatever external environment the PCI1251A will be working with: 3.3 V or 5 V. The I/O sites can be pulled through a clamping diode to a voltage that protects the core from external signals. The core power supply is always 3.3 V and is independent of the clamping voltages. For example, PCI signaling can be either 3.3 V or 5 V, and the PCI1251A must reliably accommodate both voltage levels. This is accomplished by using a 3.3-V I/O buffer that is 5-V tolerant, with the applicable clamping voltage applied. If a system designer desires a 5-V PCI bus, V_{CCP} can be connected to a 5-V power supply.

The PCI1251A requires five separate clamping voltages because it supports a wide range of features. The five rails are listed and defined in the *recommended operating conditions*, on page 109.

PCI interface

This section describes the PCI interface of the PCI1251A and how the device responds and participates in PCI bus cycles. The PCI1251A provides all required signals for PCI master/slave devices, and can operate in either 5-V or 3.3-V PCI signaling environments by connecting the V_{CCP} terminals to the desired signaling level.

PCI bus lock (LOCK)

The bus-locking protocol defined in the PCI specification is not highly recommended, but is provided on the PCI1251A 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). Note that the use of $\overline{\text{LOCK}}$ is only supported by PCI-to-CardBus bridges in the downstream direction (away from the processor).

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

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

loading the subsystem identification (EEPROM interface)

The subsystem vendor ID register and subsystem ID register make up a doubleword of PCI configuration space located at offset 40h for functions 0 and 1. This doubleword register is used for system and option card (mobile dock) identification purposes and is required by some operating systems. Implementation of this unique identifier register is a PC '95 requirement.

The PCI1251A offers two mechanisms to load a read-only value into the subsystem registers. The first mechanism relies upon the system BIOS providing the subsystem ID value. The default access mode to the subsystem registers is read only, but can be made read/write by clearing the SUBSYSRW bit in the system control register (bit 5, offset 80h). Once this bit is cleared (0), the BIOS can write a subsystem identification value into the registers at offset 40h. The BIOS must set the SUBSYSRW bit such that the subsystem vendor ID register and subsystem ID register is 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 PCI1251A loads the doubleword of data from the serial EEPROM after a reset of the primary bus. Note that the $\overline{\text{SUSPEND}}$ input gates the PCI reset from the entire PCI1251A core, including the serial EEPROM state machine (see *suspend mode*, on page 37, for details on using $\overline{\text{SUSPEND}}$). The PCI1251A provides a two-line serial bus interface to a serial EEPROM.

The system designer must implement a pulldown resistor on the PCI1251A LATCH terminal to indicate the serial EEPROM mode. Only when this pulldown resistor is present will the PCI1251A attempt to load data through the serial EEPROM interface. The serial EEPROM interface is a two-pin interface with one data signal (SDA) and one clock signal (SCL). SDA is mapped to the PCI1251A IRQMUX6 terminal and SCL is mapped to the PCI1251A IRQMUX7 terminal. A typical PCI1251A application using the serial EEPROM interface is shown in Figure 2.

loading the subsystem identification (EEPROM interface) (continued)

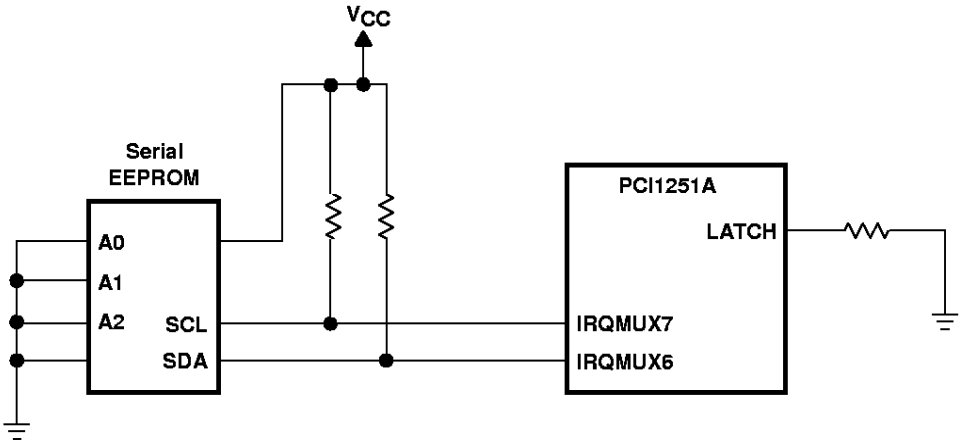


Figure 2. Serial EEPROM Application

When the PCI1251A is reset, the subsystem data is read automatically from the EEPROM. The PCI1251A masters the serial EEPROM bus and reads four bytes, as shown in Figure 3.

The EEPROM is addressed at word address 00h, as shown in Figure 3, and the address auto increments after each byte transfer 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.

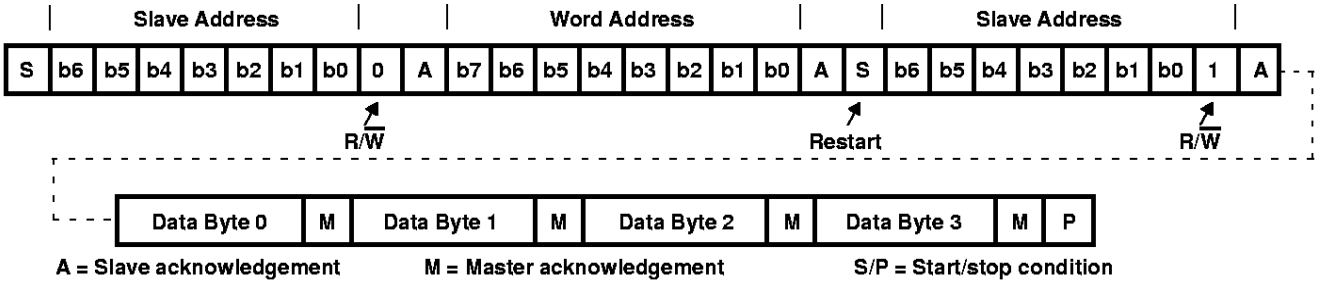


Figure 3. EEPROM Interface Subsystem Data Collection

The serial EEPROM is addressed at slave address 1010000b by the PCI1251A. 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 2) 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, and the protocol is defined for the bidirectional transfers. Both SCL and SDA are 3-stated and pulled high when the bus is not active. When the SDA line transitions 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 are interpreted as a control signal. Data is valid and stable during the clock high period. This protocol is shown in Figure 4.

loading the subsystem identification (EEPROM interface) (continued)

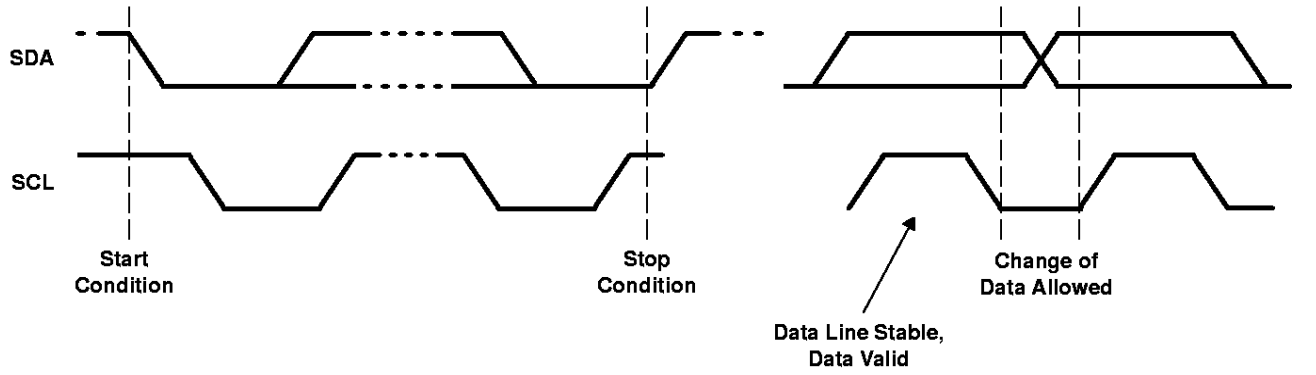


Figure 4. Serial EEPROM Start/Stop Conditions and Bit Transfers

Each address byte and data transfer is followed by an acknowledge bit, as shown in Figure 3. When the PCI1251A transmits the addresses, it returns SDA to the high state and 3-states the line. The PCI1251A then generates an SCL clock cycle and expects the EEPROM to pull down SDA during the acknowledge pulse. This procedure is referred to as a slave acknowledge with the PCI1251A transmitter and EEPROM receiver. General acknowledges are shown in Figure 5.

During the data byte transfers from the serial EEPROM to the PCI1251A, the EEPROM clocks the SCL signal. After the EEPROM transmits the data to the PCI1251A, it returns the SDA signal to the high state and 3-states the line. The EEPROM then generates an SCL clock cycle and expects the PCI1251A to pull down SDA during the acknowledge pulse. This procedure is referred to as a master acknowledge with the EEPROM transmitter and PCI1251A receiver. General acknowledges are shown in Figure 5.

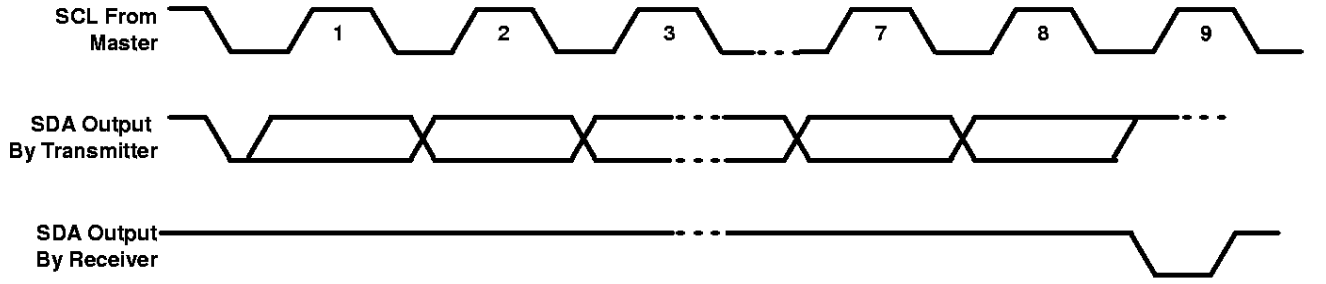


Figure 5. Serial EEPROM Protocol – Acknowledge

EEPROM interface status information is communicated through the general status register located at PCI offset 85h. The EEDETECT bit in this register indicates whether or not the PCI1251A serial EEPROM circuitry detects the pulldown resistor on LATCH. An error condition, such as a missing acknowledge, results in the DATAERR bit being set. The EEBUSY bit is set while the subsystem ID register is loading (serial EEPROM interface is busy).

PC Card applications

This section describes the following PC Card interfaces: PC Card recognition (which details the card interrogation procedure), card-powering procedure (including the protocol of the P^2C power-switch interface), internal zoom video (ZV) buffering provided by the PCI1251A and programming model, standard PC Card register models, and a brief discussion of the PC Card software protocol layers.

PC Card insertion/removal and recognition

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

The scheme uses the $\overline{CD1}$, $\overline{CD2}$, $\overline{VS1}$, and $\overline{VS2}$ signals ($\overline{CCD1}$, $\overline{CCD2}$, CVS1, and CVS2 for CardBus). A PC Card designer connects these four terminals in prescribed configuration determined by the type of card and the supply voltage. The encoding scheme for this is defined in the 1997 PC Card Standard and is shown in Table 19.

Table 19. PC Card Card-Detect and Voltage-Sense Connections

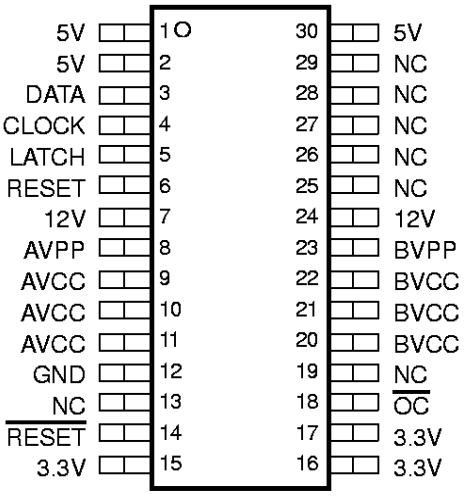
$\overline{CD2}$//$\overline{CCD2}$	$\overline{CD1}$//$\overline{CCD1}$	$\overline{VS2}$//CVS2	$\overline{VS1}$//CVS1	KEY	INTERFACE	VOLTAGE
Ground	Ground	Open	Open	5 V	16-bit PC Card	5 V
Ground	Ground	Open	Ground	5 V	16-bit PC Card	5 V and 3.3 V
Ground	Ground	Ground	Ground	5 V	16-bit PC Card	5 V, 3.3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

P^2C power-switch interface (TPS2206)

A power switch with a PCMCIA-to-peripheral control (P^2C) interface is required for the PC Card powering interface. The TI TPS2206 dual-slot PC Card power-interface switch provides the P^2C interface to the CLOCK, DATA, and LATCH terminals of the PCI1251A. Figure 6 shows the terminal assignments of the TPS2206 and Figure 7 illustrates a typical application where the PCI1251A represents the PCMCIA controller.

The CLOCK terminal on the PCI1251A can be an input or an output depending on whether bit 27 of the system control register is a 0 or a 1. The default is for the CLOCK terminal to be an input to control the serial interface and the PCI1251A internal state machine. The P2CCLK bit in the system control register can be set by the system BIOS to enable the PCI1251A to internally generate and drive the CLOCK from the PCI clock. When the system design implements CLOCK as an output from the PCI1251A, an external pulldown resistor is required since the CLOCK terminal defaults to an input.

P²C power-switch interface (TPS2206) (continued)



NC – No internal connection

Figure 6. TPS2206 Terminal Assignments

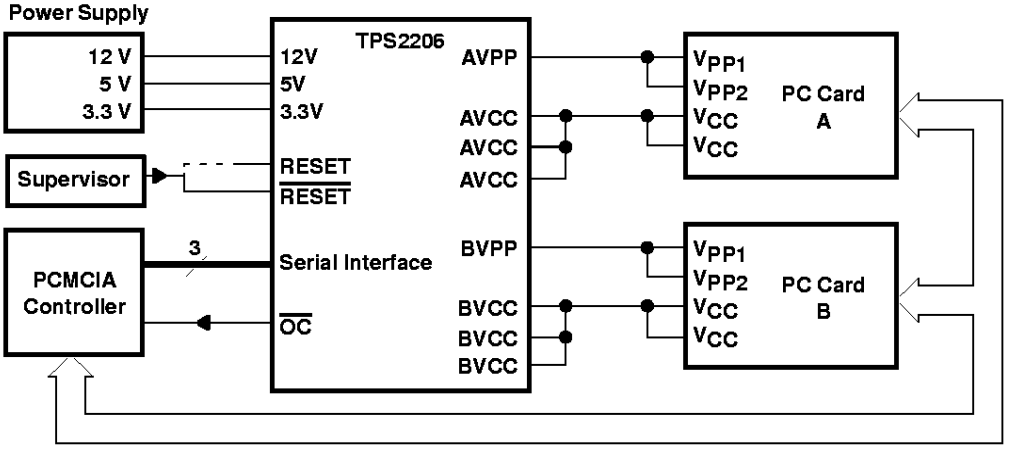
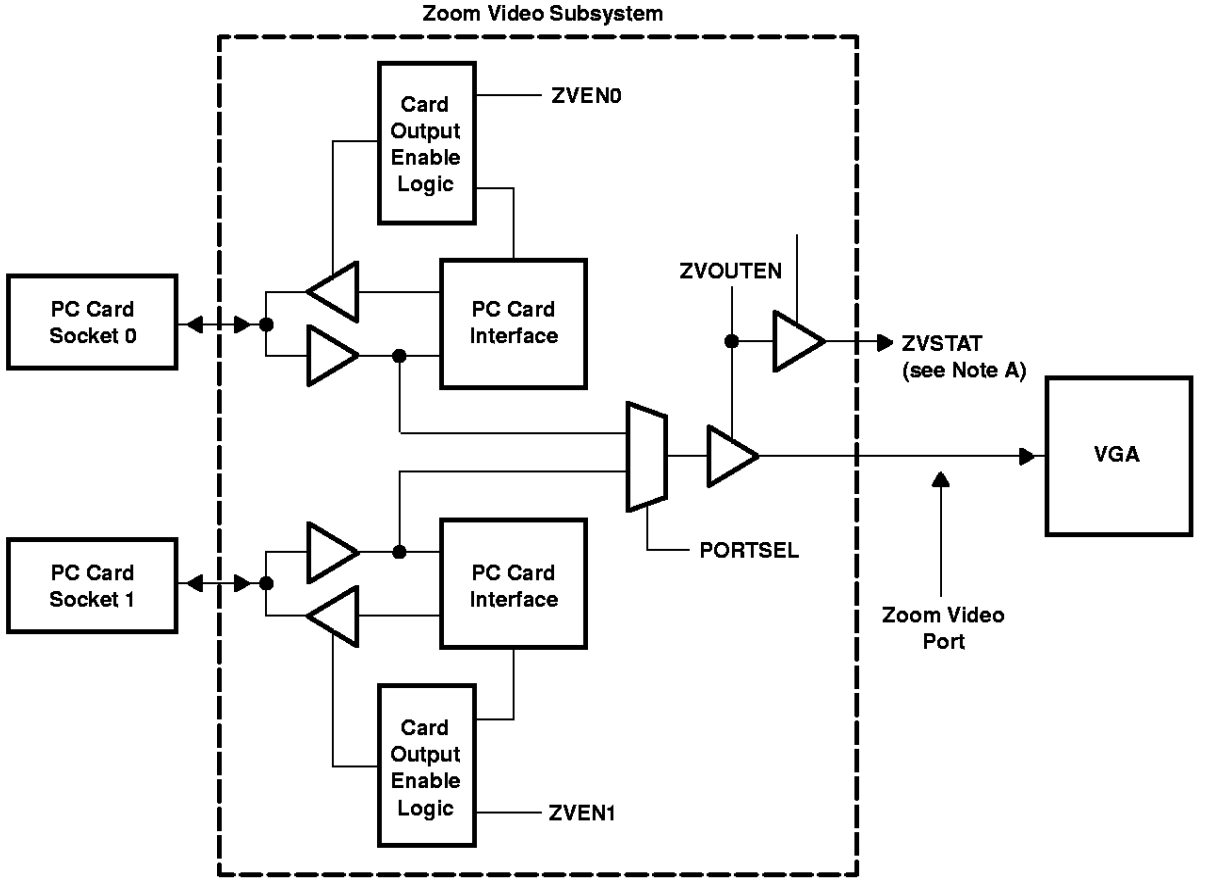


Figure 7. TPS2206 Typical Application

zoom video support

The ZV port on the PCI1251A provides an internally buffered 16-bit ZV PC Card data path. This internal routing is programmed through the multimedia control register. Figure 8 shows the zoom video subsystem implemented in the PCI1251A and details the bit functions found in the multimedia control register.

An output port (PORTSEL) is always selected. The PCI1251A defaults to socket 0 (see the multimedia control register on page 59). When ZVOUTEN is asserted, the zoom video output terminals are enabled to allow the PCI1251A to route the zoom video data. However, no data is transmitted unless either ZVEN0 or ZVEN1 is enabled in the multimedia control register. When the PORTSEL maps to a card port that is disabled (ZVEN0 or ZVEN1) then the zoom video port is driven low; that is, no data is transmitted.



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

Figure 8. Zoom Video Subsystem

SPKROUT usage

SPKROUT carries the digital audio signal from the PC Card to the system. When a 16-bit PC Card is configured for I/O mode, the BVD2 pin becomes $\overline{\text{SPKR}}$. This terminal, referred to as CAUDIO, is also used in CardBus applications. $\overline{\text{SPKR}}$ passes a TTL-level digital audio signal to the PCI1251A. The CardBus CAUDIO signal also can pass a single-amplitude binary waveform. The binary audio signals from the two PC Card sockets are XORed in the PCI1251A to produce SPKROUT. Figure 9 shows the SPKROUT connection.

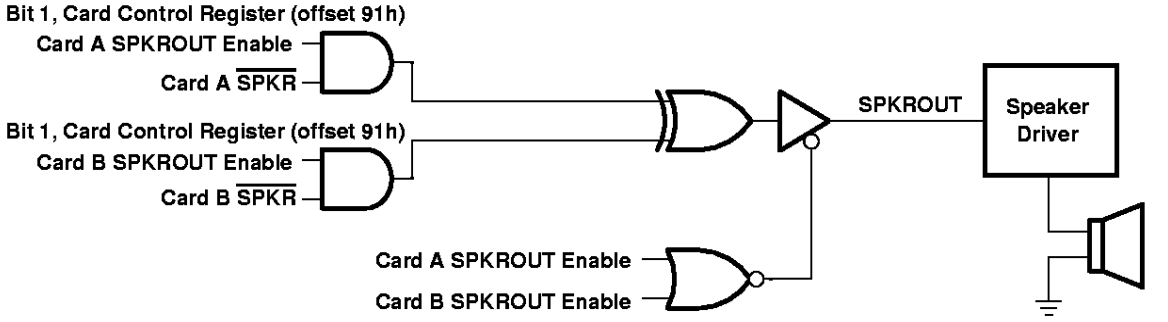


Figure 9. SPKROUT Connection to Speaker Driver

The SPKROUT signal is typically driven only by modem PC Cards. To verify SPKROUT on the PCI1251A, a sample circuit was constructed, and this simplified schematic is shown in Figure 10.

NOTE:
Earlier versions of the PC Card controller multiplexed $\overline{\text{SUSPEND}}$/SPKROUT on the same pin, which meant that a pullup resistor was needed to differentiate the signals. Because the PCI1251A does not multiplex this or any other function on SPKROUT, this terminal does not require a pullup resistor.

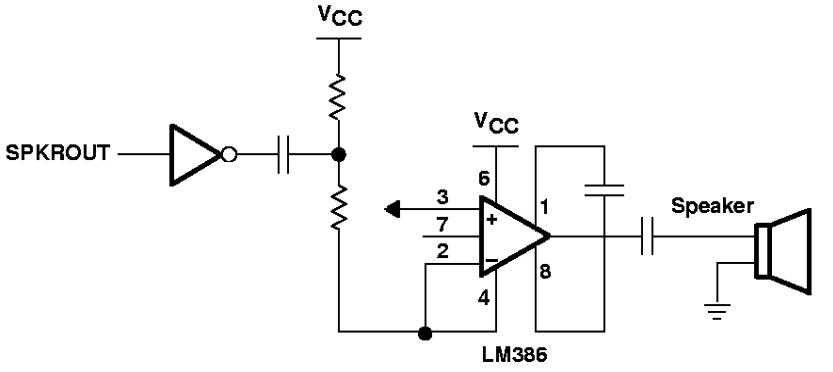


Figure 10. Simplified Test Schematic

LED socket activity indicators

Socket activity LEDs provided indicate when a PC Card is being accessed. The LED signals are multiplexed with general-purpose inputs and outputs (GPIOs), and the default for these terminals is GPI. When configured for LED outputs, these terminals output an active high signal to indicate socket activity. LEDA1 indicates socket 0 (card A) activity, and LEDA2 indicates socket 1 (card B) activity.

The LED signal is active high and is driven for 64-ms periods. When the LED is not being driven high, it is driven to a low state. Either of the two circuits shown in Figure 11 can be implemented to provide LED signaling, and the board designer can implement the circuit that best fits the application.

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 $\overline{\text{SUSPEND}}$ is asserted or when the PCI clock is stopped per the $\overline{\text{CLKRUN}}$ protocol.

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

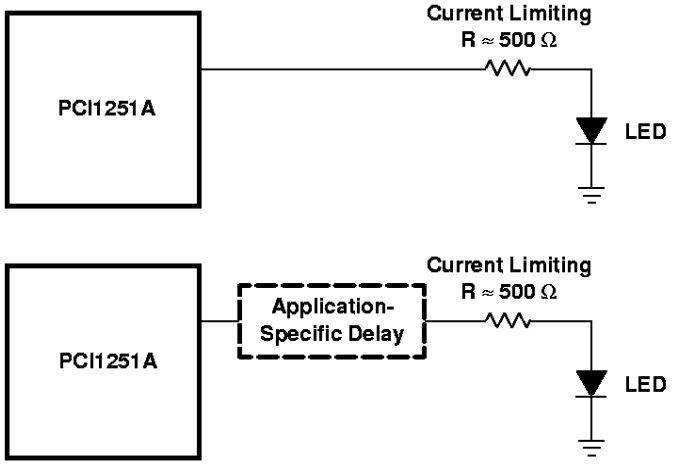


Figure 11. Two Sample LED Circuits

PC Card16 DMA support

The PCI1251A 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. The DDMA register configuration is provided in Table 20.

Table 20. Distributed DMA Registers

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

CardBus socket registers

The PCI1251A has socket registers for compatibility with the latest PCI-to-PCMCIA CardBus bridge specification. These CardBus socket registers are listed in Table 21 below.

Table 21. CardBus Socket Registers

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

programmable interrupt subsystem

Interrupts provide a way for I/O devices to let the microprocessor know that they require servicing. The dynamic nature of PC Cards, and the abundance of PC Card I/O applications require substantial interrupt support from the PCI1251A. The PCI1251A provides several interrupt signaling schemes to accommodate the needs of a variety of platforms. The different mechanisms for dealing with interrupts in this device are based on various specifications and industry standards. The ExCA register set provides interrupt control for some 16-bit PC Card functions, and the CardBus socket register set provides interrupt control for the CardBus PC Card functions. The PCI1251A is, therefore, backward compatible with existing interrupt control register definitions, and new registers have been defined where required.

The PCI1251A 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 PCI1251A, PC Card interrupts are classified as either card status change (CSC) or as functional interrupts.

The method by which any type of PCI1251A interrupt is communicated to the host interrupt controller varies from system to system. The PCI1251A offers system designers the choice of using parallel PCI interrupt signaling, parallel ISA-type IRQ interrupt signaling, or the IRQSER serialized ISA and/or PCI interrupt protocol. Traditional ISA IRQ signaling is provided through eight IRQMUX terminals. It is possible to use the parallel PCI interrupts in combination with either parallel IRQs or serialized IRQs, as detailed in the sections that follow.

PC Card functional and CSC interrupts

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

Card status change-type interrupts are defined as events at the PC Card interface that are detected by the PCI1251A and may warrant notification of host card and socket services software for service. CSC events include both card insertion and removal from PC Card sockets, as well as transitions of certain PC Card signals.

Table 22 summarizes the sources of PC Card interrupts and the type of card associated with them. CSC and functional interrupt sources are dependent on the type of card inserted in the PC Card socket. The three types of cards that can be inserted into any PC Card socket are 16-bit memory card, 16-bit I/O card, and CardBus cards. Note that 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.

PC Card functional and CSC interrupts (continued)

Table 22. PC Card Interrupt Events and Description

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

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

The PC Card standard describes the power-up sequence that must be followed by the PCI1251A 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 PCI1251A interrupt scheme can be used to notify the host system (see Table 23), denoted by the power cycle complete event. This interrupt source is considered a PCI1251A internal event, because it depends on the completion of applying power to the socket.

interrupt masks and flags

Host software may individually mask (or disable) most of the potential interrupt sources listed in Table 23 by setting the appropriate bits in the PCI1251A. By individually masking the interrupt sources listed, software can control those events that cause a PCI1251A interrupt. Host software has some control over the system interrupt the PCI1251A asserts by programming the appropriate routing registers. The PCI1251A allows host software to route PC Card CSC and PC Card functional interrupts to separate system interrupts. A discussion of interrupt routing is somewhat specific to the interrupt signaling method used, and is discussed in more detail in the following sections.

interrupt masks and flags (continued)

When an interrupt is signaled by the PCI1251A, the interrupt service routine must determine which of the events in Table 23 caused the interrupt. Internal registers in the PCI1251A provide flags that report which interrupt source was the cause of the interrupt. By reading these status bits, the interrupt service routine can determine the action to be taken.

Table 23 details the registers and bits associated with masking and reporting potential interrupts. All interrupts, except the functional PC Card interrupts, can be masked. An interrupt status flag is available for all types of interrupts.

Table 23. Interrupt Mask and Flag 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 bit 0	Socket event bit 0
	Interrupt request ($\overline{\text{CINT}}$)	Always enabled	PCI configuration offset 91h bit 0
	Power cycle complete	Socket mask bit 3	Socket event bit 3
	Card insertion or removal	Socket mask bits 2 and 1	Socket event bits 2 and 1

Notice that there is not a mask bit to stop the PCI1251A from passing PC Card functional interrupts through to the appropriate interrupt scheme. These interrupts are not valid until the card is properly powered, and there should never be a card interrupt that does not require service after proper initialization.

Various methods of clearing the interrupt flag bits are listed in Table 23. The flag bits in the ExCA registers (16-bit PC Card-related interrupt flags) can be cleared by two different methods. One method is an explicit write of 1 to the flag bit to clear, and the other is by reading the flag bit register. The selection of flag bit clearing is made by bit 2 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 be cleared by an explicit write of 1 to the interrupt flag in the socket event register. Although some of the functionality is shared between the CardBus registers and the ExCA registers, software should not program the chip through both register sets when a CardBus card is functioning.

legacy interrupt multiplexer

The IRQ multiplexer implemented in the PCI1251A provides a mechanism to route the IRQMUX signals to any of the 15 legacy IRQ signals. IRQMUX7–IRQMUX6 share the PC/PCI DMA terminals and take precedence when routed. The other six IRQMUX signals (IRQMUX5–IRQMUX0) are available in all platforms. To use the IRQMUX interrupt signaling, software must program the device control register (PCI offset 92h) to select the legacy IRQ signaling scheme.

The IRQMUX functionality describing PCREQ/IRQMUX7 is shown in Figure 12.

legacy interrupt multiplexer (continued)

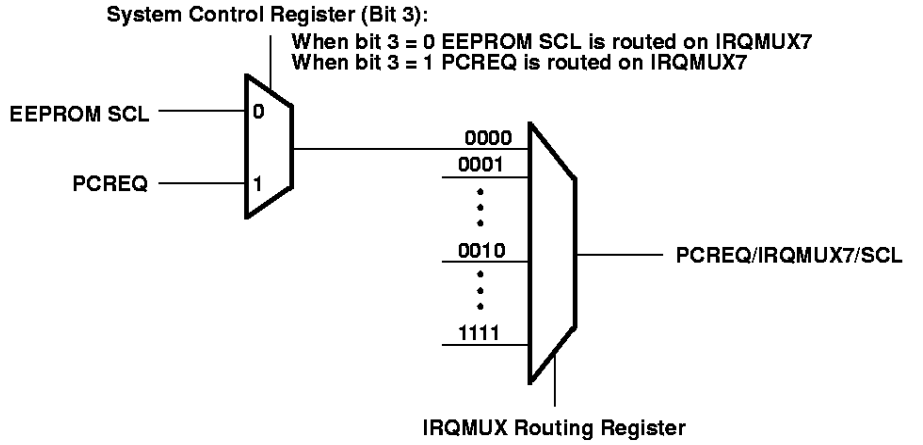


Figure 12. Interrupt Mux Functionality – Example of IRQMUX7 Routing

If parallel ISA IRQs are selected in the device control register, then the IRQMUX routing register (PCI offset 8Ch) must be programmed with the associated ISA IRQ connections. The PCI1251A supports up to eight parallel ISA IRQ signal connections (IRQMUX7–IRQMUX0). Figure 13 is an example of PCI1251A IRQ implementation that provides eight ISA interrupts. The system In this example cannot support PC/PCI DMA because all eight ISA IRQs are used. In this example, IRQMUX7 and IRQMUX6 are used to signal ISA IRQs and are not available for PC/PCI DMA. For systems not using all eight IRQs, PC/PCI DMA can be implemented and can coexist with ISA IRQs by using IRQMUX6 and IRQMUX7 for PC/PCI DMA; that is, legacy IRQs and PC/PCI DMA implementation are not mutually exclusive. However, if the IRQMUX registers are programmed to use IRQMUX7–6, they override PC/PCI DMA.

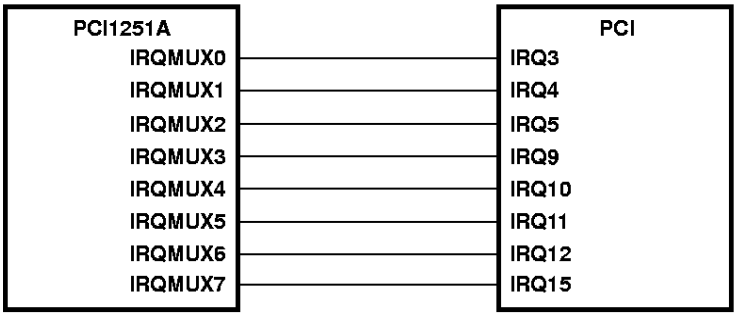


Figure 13. IRQ Implementation

Software is responsible for programming the IRQMUX routing register to reflect the IRQ configuration shown in Figure 13. In this example, programming is accomplished by writing a doubleword of data (FCBA9543h) to the PCI1251A IRQMUX routing register, PCI offset 8Ch. In this example (FCBA9543h), F corresponds to IRQ15, C to IRQ12, B to IRQ11, A to IRQ10, 9 to IRQ9, 5 to IRQ5, 4 to IRQ4, and 3 to IRQ3.

The IRQMUX routing register is shared between the two PCI1251A functions, and only one write to function 0 or function 1 is necessary to configure the IRQMUX signals.

using parallel PCI interrupts

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

using parallel PCI interrupts (continued)

PCI $\overline{\text{INTB}}$ is also available on the IRQMUX0 terminal by programming bits 3–0 to 0001b. See the IRQMUX routing register (8Ch). PCI $\overline{\text{INTA}}$ is available on the GPIO3 terminal by programming bits 7–6 in the GPIO3 control register (8Bh) to 00b.

The value read from the interrupt pin register is card slot dependent. The value read also depends on the interrupt INTRTIE bit in the system control register and the signaling mode selected through the device control register. When the INTRTIE bit is set, this register reads 0x01 (INTA) for both functions. The PCI1251A defaults to signaling PCI and IRQ interrupts through IRQSER serial interrupt terminal. Refer to Table 24 for a complete description of the register contents.

Table 24. 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 and parallel PCI interrupts	0	0x01 ($\overline{\text{INTA}}$)	0x02 ($\overline{\text{INTB}}$)
IRQ serialized (IRQSER) and parallel PCI interrupts	0	0x01 ($\overline{\text{INTA}}$)	0x02 ($\overline{\text{INTB}}$)
IRQ and 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 and parallel PCI interrupts	1	0x01 ($\overline{\text{INTA}}$)	0x01 ($\overline{\text{INTA}}$)
IRQ serialized (IRQSER) and parallel PCI interrupts	1	0x01 ($\overline{\text{INTA}}$)	0x01 ($\overline{\text{INTA}}$)
IRQ and PCI serialized (IRQSER) interrupts (default)	1	0x01 ($\overline{\text{INTA}}$)	0x01 ($\overline{\text{INTA}}$)

power management overview

TI has expended great effort to provide a high-performance device with low power consumption. In addition to the low-power CMOS technology process used for the PCI1251A, various features are designed into the device to allow implementation of popular power-saving techniques. These features and techniques are discussed in this section.

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

The PCI $\overline{\text{CLKRUN}}$ feature is the primary method of power management on the PCI bus side of the PCI1251A. 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 the CLKRUN terminal should be tied low. $\overline{\text{CLKRUN}}$ is enabled by default using bit 1 (KEEPCLK) in the system control register (80h).

CardBus PC Card power management

The PCI1251A implements its own card power-management engine that can be used to turn off the CCLK at 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 clock run protocol is followed on the CardBus interface to control this clock management.

PCI power management (PCIPM)

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

PCI power management (PCIPM) (continued)

The four power-management states of PCI functions are:

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

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

For the operating system (OS) to power manage the device power states on the PCI bus, the PCI function should support four power-management operations. These are capabilities reporting, power status reporting, setting the power state, and system wake up. The OS identifies the capabilities of the PCI function by traversing the new capabilities list. The presence of new capabilities is indicated by a 1 in the capabilities list bit in the PCI Status register (bit 4) and providing access to a capabilities list.

The capabilities pointer provides access to the first item in the linked list of capabilities. For the PCI1251A, a CardBus bridge with PCI configuration space header type 2, the capabilities pointer is mapped to an offset of 14h. The first byte of each capability register block is required to be a unique ID of that capability. PCI power management has been assigned an ID of 01h. The next byte is a pointer to the next pointer item in the list of capabilities. If there are no more items in the list, the next item pointer should be set to 0. The registers following the next item pointer are specific to the function's capability. The PCIPM capability implement the register block outlined in Table 25.

Table 25. Power-Management Registers

REGISTER NAME				OFFSET
Power-management capabilities		Next item pointer	Capability ID	0
Data	PMCSR bridge support extensions	Power-management control status (CSR)		4

The 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 state-dependent power measurements, such as power consumption or heat dissipation.

suspend mode

The $\overline{\text{SUSPEND}}$ signal is provided for backward compatibility, and gates the PCI reset ($\overline{\text{PRST}}$) signal from the PCI1251A. However, additional functionality has been defined for $\overline{\text{SUSPEND}}$ to provide additional power-management options.

$\overline{\text{SUSPEND}}$ provides a mechanism to gate the PCLK from the PCI1251A, as well as gate $\overline{\text{PRST}}$. This can potentially save power while in an idle state; however, it requires substantial design effort to implement. Some issues to consider are:

- What if cards are present in the sockets?
- What if the cards in the sockets are powered?
- How to pass CSC (insertion/removal) events?

Even without the PCI clock to the PCI1251A core, there are asynchronous-type functions (such as $\overline{\text{RI\_OUT}}$) that can pass CSC events, wake-up events, etc., back to the system. If a system designer chooses to not pass card removal events through to the system, then the PCI1251A would not be able to power down the empty socket without the power switch clock (CLOCK) generated externally. Refer to the P^2C power switch interface on page 27 for details. Figure 14 is a functional implementation diagram.

suspend mode (continued)

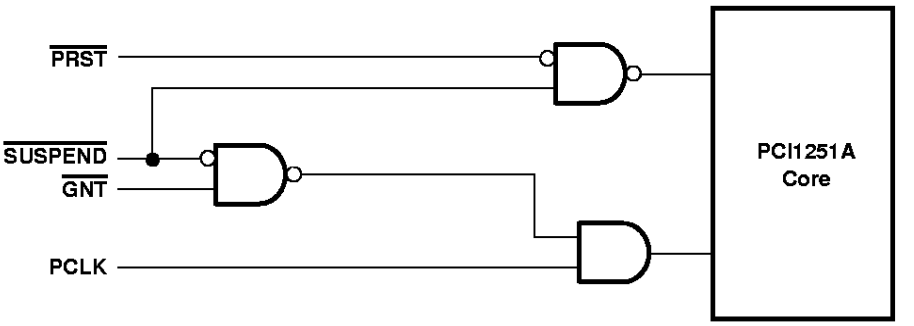


Figure 14. $\overline{\text{SUSPEND}}$ Functional Implementation

requirements for suspend mode

The PCI bus must not be parked on the PCI1251A when $\overline{\text{SUSPEND}}$ is asserted. The PCI1251A responds to $\overline{\text{SUSPEND}}$ being asserted by placing the $\overline{\text{REQ}}$ pin in a high impedance state and gates the internal clock and reset.

The GPIOs, IRQMUX signals, and $\overline{\text{RI\_OUT}}$ signals are active during $\overline{\text{SUSPEND}}$ unless they are disabled in the appropriate PCI1251A registers.

ring indicate

The $\overline{\text{RI\_OUT}}$ output is an important feature in power management and is basically used so that a system can go into a suspended mode and wake up on modem rings and other card events. TI has designed in flexibility to this signal to fit wide platform requirements. $\overline{\text{RI\_OUT}}$ on the PCI1251A can be asserted under any of the following conditions:

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

CSTSCHG 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. Figure 15 shows various enable bits for the PCI1251A $\overline{\text{RI\_OUT}}$ function; however, it does not show the masking of CSC events (see *interrupt masks and flags*, on page 33, for a detailed description of CSC interrupt masks and flags).

$\overline{\text{RI\_OUT}}$ is multiplexed with $\overline{\text{PME}}$ on the same terminal. The default is for $\overline{\text{RI\_OUT}}$ to be signaled on this terminal. In PCI power management systems the $\overline{\text{PME}}$ signal should be enabled by setting bit 8 (PME_EN) in the power-management control/status register (A4h) and setting RIMUX 80h to 1.

In addition to bit 0 in the system control register, the RIENB bit (bit 7) in the card control register (91h) must be set to enable $\overline{\text{RI\_OUT}}$.

ring indicate (continued)

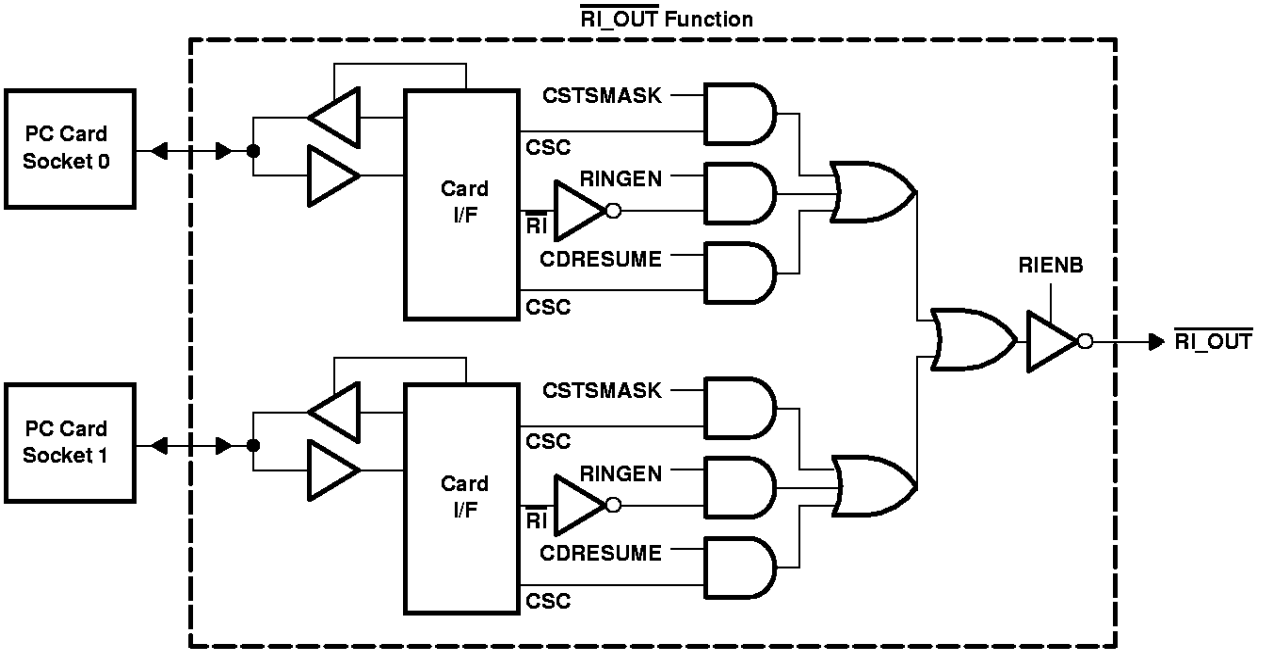


Figure 15. $\overline{\text{RI\_OUT}}$ Functional Diagram

Routing of CSC events to $\overline{\text{RI\_OUT}}$ is enabled on a per-socket basis, and is programmed by the RIENB bit in the card control register on page 66. This bit is socket dependent (not shared), as shown in Figure 15.

Ring indicate (RI) from the 16-bit PC Card interface is masked by the ExCA control bit RINGEN in the interrupt and general control register on page 82. 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 (page 100) in the CardBus socket registers.

PC Card controller programming model

This section describes the PCI1251A PCI configuration registers that make up the 256-byte PCI configuration header for each PCI1251A function. As noted, some bits are global in nature and should be accessed only through function 0.

PCI configuration registers (functions 0 and 1)

The PCI1251A is a multifunction PCI device, and the PC Card controller is integrated as PCI functions 0 and 1. The configuration header is compliant with the PCI specification as a CardBus bridge header, and is PC99 compliant as well. Table 26 shows the PCI configuration header, which includes both the predefined portion of the configuration space and the user-definable registers.

Table 26. PCI Configuration Registers (Functions 0 and 1)

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
Memory base register 0				1Ch
Memory limit register 0				20h
Memory base register 1				24h
Memory limit register 1				28h
I/O base register 0				2Ch
I/O limit register 0				30h
I/O base register 1				34h
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
GPIO3 control	GPIO2 control	GPIO1 control	GPIO0 control	88h
IRQMUX routing				8Ch
Diagnostic	Device control	Card control	Retry status	90h
Socket DMA register 0				94h
Socket DMA register 1				98h
Reserved				9Ah–9Fh
Power-management capabilities		Next-item pointer	Capability ID	A0h
Data (reserved)	PMCSR Bridge Support Extensions	Power-management control/status		A4h

vendor ID register

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

Register: **Vendor ID**
Type: Read only
Offset: 00h (functions 0, 1)
Default: 104Ch
Description: This 16-bit read-only register contains a value allocated by the PCI SIG (special interest group) and identifies the manufacturer of the PCI device. The vendor ID assigned to TI is 104Ch.

device ID register

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

Register: **Device ID**
Type: Read only
Offset: 02h (functions 0, 1)
Default: AC1Dh
Description: This 16-bit read-only register contains a value assigned to the PCI1251A by TI. The device identification for the PCI1251A is AC1D.

command register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Command															
Type	R	R	R	R	R	R	R	R/W	R	R/W	R	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 (see individual bit descriptions)
Offset: 04h
Default: 0000h
Description: The command register provides control over the PCI1251A interface to the PCI bus. All bit functions adhere to the definitions in PCI Local Bus Specification 2.2. None of the bit functions in this register are shared between the two PCI1251A PCI functions. Two command registers exist in the PCI1251A, one for each function. Software must manipulate the two PCI1251A functions as separate entities when enabling functionality through the command register. The SERR_EN and PERR_EN enable bits in this register are internally wired OR between the two functions, and these control bits appear separately according to their software function.

Table 27. Command Register

BIT	SIGNAL	TYPE	FUNCTION
15–10	RSVD	R	Reserved. Bits 15–10 are read only and return 0s when read. Writes have no effect.
9	FBB_EN	R	Fast back-to-back enable. The PCI251 does not generate fast back-to-back transactions; therefore, bit 9 is read only and returns 0s 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 bit 8 and bit 6 must be set for the PCI1251A to report address parity errors. 0 = Disable $\overline{\text{SERR}}$ output driver (default). 1 = Enable $\overline{\text{SERR}}$ output driver.
7	ADSTPNG	R	Address/data stepping control. The PCI1251A does not support address/data stepping, and bit 7 is hardwired to 0. Writes to this bit have no effect.
6	PERR_EN	R/W	Parity error response enable. Bit 6 controls the PCI1251A's response to parity errors through $\overline{\text{PERR}}$. Data parity errors are indicated by asserting $\overline{\text{PERR}}$, whereas address parity errors are indicated by asserting $\overline{\text{SERR}}$. 0 = PCI1251A ignores detected parity error (default). 1 = PCI1251A responds to detected parity errors.
5	VGA_SNP	R	VGA palette snoop. Bit 5 controls how PCI devices handle accesses to video graphics array (VGA) palette registers. The PCI1251A does not support VGA palette snooping; therefore, this bit is hardwired to 0. Bit 5 is read only and returns 0 when read. Writes to this bit have no effect.
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 PCI1251A controller does not support memory write and invalidate commands; it uses memory write commands instead; therefore, this bit is hardwired to 0. Bit 4 is read only and returns 0 when read. Writes to this bit have no effect.
3	SP_CYCL	R	Special cycles. Bit 3 controls whether or not a PCI device ignores PCI special cycles. The PCI1251A does not respond to special cycle operations; therefore, this bit is hardwired to 0. Bit 3 is read only and returns 0 when read. Writes to this bit have no effect.
2	BUSMSTR	R/W	Bus master control. Bit 2 controls whether or not the PCI1251A can act as a PCI bus initiator (master). The PCI1251A can take control of the PCI bus only when this bit is set. 0 = Disables the PCI1251A's ability to generate PCI bus accesses (default). 1 = Enables the PCI1251A's ability to generate PCI bus accesses.
1	MEM_EN	R/W	Memory space enable. Bit 1 controls whether or not the PCI1251A can claim cycles in PCI memory space. 0 = Disables the PCI1251A's response to memory space accesses (default). 1 = Enables the PCI1251A's response to memory space accesses.
0	IO_EN	R/W	I/O space control. Bit 0 controls whether or not the PCI1251A can claim cycles in PCI I/O space. 0 = Disables the PCI1251A from responding to I/O space accesses (default). 1 = Enables the PCI1251A to respond to I/O space accesses.

status register

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

Register: **Status**
Type: Read only, read/write (see individual bit descriptions)
Offset: 06h (functions 0, 1)
Default: 0210h
Description: The status register provides device information to the host system. Bits in this register may be read normally. A bit in the status register is reset when a 1 is written to that bit location; a 0 written to a bit location has no effect. All bit functions adhere to the definitions in the PCI Local Bus Specification 2.2. PCI bus status is shown through each function (see Table 28).

Table 28. Status Register

BIT	SIGNAL	TYPE	FUNCTION
15	PAR_ERR	R/W	Detected parity error. Bit 15 is set when a parity error is detected (either address or data). Write a 1 to clear this bit.
14	SYS_ERR	R/W	Signaled system error. Bit 14 is set when $\overline{\text{SERR}}$ is enabled and the PCI1251A signals a system error to the host. Write a 1 to clear this bit.
13	MABORT	R/W	Received master abort. Bit 13 is set when a cycle initiated by the PCI1251A on the PCI bus has been terminated by a master abort. Write a 1 to clear this bit.
12	TABT_REC	R/W	Received target abort. Bit 12 is set when a cycle initiated by the PCI1251A on the PCI bus was terminated by a target abort. Write a 1 to clear this bit.
11	TABT_SIG	R/W	Signaled target abort. Bit 11 is set by the PCI1251A when it terminates a transaction on the PCI bus with a target abort. Write a 1 to clear this bit.
10–9	PCI_SPEED	R	DEVSEL timing. These read-only bits encode the timing of $\overline{\text{DEVSEL}}$ and are hardwired 01b, indicating that the PCI1251A asserts PCI_SPEED at a medium speed on nonconfiguration cycle accesses.
8	DATAPAR	R/W	Data parity error detected. Write a 1 to clear this bit. 0 = The conditions for setting bit 8 have not been met. 1 = A data parity error occurred, and the following conditions were met: a. $\overline{\text{PERR}}$ was asserted by any PCI device including the PCI1251A. b. The PCI1251A was the bus master during the data parity error. c. The parity error response bit is set in the command.
7	FBB_CAP	R	Fast back-to-back capable. The PCI1251A cannot accept fast back-to-back transactions; thus, bit 7 is hardwired to 0.
6	UDF	R	User-definable feature support. The PCI1251A does not support the user-definable features; thus, bit 6 is hardwired to 0.
5	66_CAP	R	66-MHz capable. The PCI1251A operates at a maximum PCLK frequency of 33 MHz; therefore, bit 5 is hardwired to 0.
4	CAP_LST	R	Capabilities list. Bit 4 is read only and 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.

revision ID register

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

Register: **Revision ID**
Type: Read only
Offset: 08h (functions 0, 1)
Default: 01h
Description: This read-only register indicates the silicon revision of the PCI1251A. This data sheet reflects the PCI1251A revision is 01h silicon.

PCI class code register

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

Register: **PCI Class code**
Type: Read only
Offset: 09h (functions 0, 1)
Default: 060700h
Description: The class code register recognizes the PCI1251A functions 0 and 1 as a bridge device (06h), and CardBus bridge device (07h) with a 00h programming interface.

cache line size register

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

Register: **Cache line size**
Type: Read/write
Offset: 0Ch (functions 0, 1)
Default: 00h
Description: The cache line size register is programmed by host software to indicate the system cache line size.

latency timer register

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

Register: **Latency timer**
Type: Read/write
Offset: 0Dh
Default: 00h
Description: The latency timer register specifies the latency timer for the PCI1251A in units of PCI clock cycles. When the PCI1251A is a PCI bus initiator and asserts $\overline{\text{FRAME}}$, the latency timer begins counting from zero. If the latency timer expires before the PCI1251A transaction has terminated, the PCI1251A terminates the transaction when its GNT is deasserted.

header type register

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

Register: **Header type**
Type: Read only
Offset: 0Eh (functions 0, 1)
Default: 82h
Description: This read-only register returns 82h when read, indicating that the PCI1251A functions 0 and 1 configuration spaces adhere to the CardBus bridge PCI header. The CardBus bridge PCI header ranges from PCI register 0 to 7Fh, and 80h–FFh is user-definable extension registers.

BIST register

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

Register: **BIST**
Type: Read only
Offset: 0Fh (functions 0, 1)
Default: 00h
Description: Because the PCI1251A does not support a built-in self-test (BIST), this register is read only and returns the value of 00h when read. This register is read only, returning 0s for the two PCI1251A functions.

CardBus socket registers/ExCA base-address register

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

capability pointer register

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

Register: **Capability pointer**
Type: Read only
Offset: 14h
Default: A0h
Description: The capability pointer register provides a pointer into the PCI configuration header where the PCI power management register block resides. PCI header doublewords at A0h and A4h provide the power management (PM) registers. Each socket has its own capability pointer register. This register is read only and returns A0h when read.

secondary status register

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

Register: **Secondary status**
Type: Read only, read/write (see individual bit descriptions)
Offset: 16h
Default: 0200h
Description: The secondary status register (see Table 29) is compatible with the PCI-to-PCI bridge secondary status register and indicates CardBus-related device information to the host system. This register is very similar to the PCI status register (offset 06h), and status bits are cleared by writing a 1. This register is not shared by the two socket functions and is accessed on a per-socket basis.

Table 29. Secondary Status Register

BIT	SIGNAL	TYPE	FUNCTION
15	CBPARITY	R/W	Detected parity error. Bit 15 is set when a CardBus parity error is detected (either address or data). Write a 1 to clear this bit.
14	CBSERR	R/W	Signaled system error. Bit 14 is set when $\overline{\text{CSERR}}$ is signaled by a CardBus card. The PCI1251A does not assert $\overline{\text{CSERR}}$. Write a 1 to clear this bit.
13	CBMABORT	R/W	Received master abort. Bit 13 is set when a cycle initiated by the PCI1251A on the CardBus bus has been terminated by a master abort. Write a 1 to clear this bit.
12	REC_CBTA	R/W	Received target abort. Bit 12 is set when a cycle initiated by the PCI1251A on the CardBus bus is terminated by a target abort. Write a 1 to clear this bit.
11	SIG_CBTA	R/W	Signaled target abort. Bit 11 is set by the PCI1251A when it terminates a transaction on the CardBus bus with a target abort. Write a 1 to clear this bit.
10–9	CB_SPEED	R	CDEVSEL timing. These read-only bits encode the timing of $\overline{\text{CDEVSEL}}$ and are hardwired 01b, indicating that the PCI1251A asserts CB_SPEED at a medium speed.
8	CB_DPAR	R/W	CardBus data parity error detected. Write a 1 to clear this bit. 0 = The conditions for setting bit 8 have not been met. 1 = A data parity error occurred and the following conditions were met: a. $\overline{\text{CPERR}}$ was asserted on the CardBus interface. b. The PCI1251A was the bus master during the data parity error. c. The parity error response bit is set in the bridge control.
7	CBFBB_CAP	R	Fast back-to-back capable. The PCI1251A cannot accept fast back-to-back transactions; thus, bit 7 is hardwired to 0.
6	CB_UDF	R	User-definable feature support. The PCI1251A does not support the user-definable features; thus, bit 6 is hardwired to 0.
5	CB66MHZ	R	66-MHz capable. The PCI1251A 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.

PCI bus number register

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

Register: **PCI bus number**
Type: Read/write
Offset: 18h (functions 0, 1)
Default: 00h
Description: This read/write register is programmed by the host system to indicate the bus number of the PCI bus to which the PCI1251A is connected. The PCI1251A uses this register in conjunction with the CardBus bus number and subordinate bus number registers to determine when to forward PCI configuration cycles to its secondary buses.

CardBus bus number register

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

Register: **CardBus bus number**
Type: Read/write
Offset: 19h
Default: 00h
Description: This read/write register is programmed by the host system to indicate the bus number of the CardBus bus to which the PCI1251A is connected. The PCI1251A 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 PCI1251A controller function.

subordinate bus number register

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

Register: **Subordinate bus number**
Type: Read/write
Offset: 1Ah
Default: 00h
Description: This read/write register is programmed by the host system to indicate the highest-numbered bus below the CardBus bus. The PCI1251A uses this register in conjunction with the PCI bus number and CardBus bus number registers to determine when to forward PCI configuration cycles to its secondary buses. This register is separate for each CardBus controller function.

CardBus latency timer register

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

Register: **CardBus latency timer**
Type: Read/write
Offset: 1Bh (functions 0, 1)
Default: 00h
Description: This read/write register is programmed by the host system to specify the latency timer for the PCI1251A CardBus interface in units of CCLK cycles. When the PCI1251A is a CardBus initiator and asserts $\overline{\text{CFRAME}}$, the CardBus latency timer begins counting. If the latency timer expires before the PCI1251A transaction has terminated, then the PCI1251A terminates the transaction at the end of the next data phase. A recommended minimum value for this register is 20h, which allows most transactions to be completed.

memory base registers 0, 1

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

Register: **Memory base registers 0, 1**
Type: Read only, read/write
Offset: 1 Ch, 24h
Default: 0000 0000h
Description: The Memory base registers indicate the lower address of a PCI memory address range. These registers are used by the PCI1251A to determine when to forward a memory transaction to the CardBus bus, and when to forward a CardBus cycle to PCI. Bits 31–12 of these registers are read/write which allow the memory base to be located anywhere in the 32-bit PCI memory space on 4K-byte boundaries. Bits 11–0 are read only and always return 0s. Writes to these bits have no effect. Bits 8 and 9 of the bridge control register specify whether memory windows 0 and 1 are prefetchable or nonprefetchable. The memory base register or the memory limit register must be nonzero for the PCI1251A to claim any memory transactions through CardBus memory windows (i.e., these windows are not enabled by default to pass the first 4K-bytes of memory to CardBus).

memory limit registers 0, 1

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

Register: **Memory limit registers 0, 1**
Type: Read only, read/write
Offset: 20h, 24h
Default: 0000 0000h
Description: The Memory limit registers indicate the upper address of a PCI memory address range. These registers are used by the PCI1251A to determine when to forward a memory transaction to the CardBus bus, and when to forward a CardBus cycle to PCI. Bits 31–12 of these registers are read/write and allow the memory base to be located anywhere in the 32-bit PCI memory space on 4K-byte boundaries. Bits 11–0 are read only and always return 0s. Writes to these bits have no effect. Bits 8 and 9 of the bridge control register specify whether memory windows 0 and 1 are prefetchable or nonprefetchable. The memory base register or the memory limit register must be nonzero for the PCI1251A to claim any memory transactions through CardBus memory windows (i.e., these windows are not enabled by default to pass the first 4K-bytes of memory to CardBus).

I/O base registers 0, 1

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

Register: **I/O base registers 0, 1**
Type: Read only, read/write
Offset: 2Ch, 34h
Default: 0000 0000h
Description: The I/O base registers indicate the lower address of a PCI I/O address range. These registers are used by the PCI1251A 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 64K-byte page, and the upper 16 bits (31–16) are all 0, which locates this 64K-byte page in the first page of the 32-bit PCI I/O address space. Bits 31–16 and bits 1–0 are read only and always return 0s, forcing I/O windows to be aligned on a natural doubleword boundary in the first 64 kilobyte 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.

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

I/O limit registers 0, 1

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

Register: **I/O limit registers 0, 1**
Type: Read only, read/write
Offset: 30h, 38h
Default: 0000 0000h
Description: The I/O limit registers indicate the upper address of a PCI I/O address range. These registers are used by the PCI1251A to determine when to forward an I/O transaction to the CardBus bus, and when to forward a CardBus cycle to PCI. The lower 16 bits of this register locate the top of the I/O window within a 64K-byte page, and the upper 16 bits are a page register that locates this 64K-byte page in 32-bit PCI I/O address space. Bits 15–2 are read/write and allow the I/O limit address to be located anywhere in the 64K-byte page (indicated by bits 31–16 of the appropriate I/O base) 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 PCI1251A assumes that the lower two 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).

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

interrupt line register

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

Register: **Interrupt line**
Type: Read/write
Offset: 3Ch
Default: FFh
Description: The interrupt line register is read/write and is used to communicate interrupt line routing information. This register is not used by the PCI1251A, because there are many programmable interrupt signaling options. This register is considered reserved; however, host software may read and write to this register. Each PCI1251A function 0 and 1 has an interrupt line register.

interrupt pin register

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

Register: **Interrupt pin**
Type: Read only
Offset: 3Dh
Default: Depends on the interrupt signaling mode
Description: The value read from the interrupt pin register is function dependent and depends on the interrupt INTRTIE bit in the system control register and the signaling mode selected through the device control register. When the INTRTIE bit is set, this register reads 0x01 (INTA) for both functions. The PCI1251A defaults to signaling PCI and IRQ interrupts through IRQSER serial interrupt terminal. Refer to Table 30 for a complete description of the register contents.

Table 30. 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 and parallel PCI interrupts	0	0x01 ($\overline{\text{INTA}}$)	0x02 ($\overline{\text{INTB}}$)
IRQ serialized (IRQSER) and parallel PCI interrupts	0	0x01 ($\overline{\text{INTA}}$)	0x02 ($\overline{\text{INTB}}$)
IRQ and 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 and parallel PCI interrupts	1	0x01 ($\overline{\text{INTA}}$)	0x01 ($\overline{\text{INTA}}$)
IRQ serialized (IRQSER) and parallel PCI interrupts	1	0x01 ($\overline{\text{INTA}}$)	0x01 ($\overline{\text{INTA}}$)
IRQ and PCI serialized (IRQSER) interrupts (default)	1	0x01 ($\overline{\text{INTA}}$)	0x01 ($\overline{\text{INTA}}$)

bridge control register

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

Register: **Bridge control**
Type: Read only, read/write (see individual bit descriptions)
Offset: 3Eh (functions 0, 1)
Default: 0340h
Description: The bridge control register provides control over various PCI1251A bridging functions. Some bits in this register are global and should be accessed only through function 0. Refer to Table 31 for a complete description of the register contents.

Table 31. Bridge Control Register

BIT	SIGNAL	TYPE	FUNCTION
15–11	RSVD	R	Reserved. Bits 15–11 return 0s when read.
10	POSTEN	R/W	Write posting enable. Enables write posting to and from the CardBus sockets. Write posting enables posting of write data on burst cycles. Operating with write posting disabled inhibits performance on burst cycles. Note that bursted write data can be posted, but various write transactions may not. Bit 10 is socket dependent and is not shared between functions 0 and 1.
9	PREFETCH1	R/W	Memory window 1 type. Bit 9 specifies whether or not memory window 1 is prefetchable. This bit is socket dependent. Bit 9 is encoded as: 0 = Memory window 1 is nonprefetchable. 1 = Memory window 1 is prefetchable (default).
8	PREFETCH0	R/W	Memory window 0 type. Bit 8 specifies whether or not memory window 0 is prefetchable. This bit is encoded as: 0 = Memory window 0 is nonprefetchable. 1 = Memory window 0 is prefetchable (default).
7	INT_RT_EN	R/W	PCI interrupt – $\overline{\text{IREQ}}$ routing enable. Bit 7 is used to select whether PC Card functional interrupts are routed to PCI interrupts or the IRQ specified in the ExCA registers. 0 = Functional interrupts routed to PCI interrupts (default) 1 = Functional interrupts routed by ExCA s
6	CRST	R/W	CardBus reset. When bit 6 is set, $\overline{\text{CRST}}$ is asserted on the CardBus interface. $\overline{\text{CRST}}$ can also be asserted by passing a $\overline{\text{PRST}}$ assertion to CardBus. 0 = $\overline{\text{CRST}}$ deasserted 1 = $\overline{\text{CRST}}$ asserted (default)
5†	MABTMODE	R/W	Master abort mode. Bit 5 controls how the PCI1251A responds to a master abort when the PCI1251A 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 $\overline{\text{SERR}}$ (if enabled)
4	RSVD	R	Reserved. Bit 4 returns 0 when read.
3	VGAEN	R/W	VGA enable. Bit 3 affects how the PCI1251A 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 PCI1251A passes I/O cycles within the 64K-byte ISA range. This bit is not common between sockets. When this bit is set, the PCI1251A 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 PCI1251A 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 PCI1251A to CardBus parity errors. This bit is separate for each socket. 0 = CardBus parity errors are ignored. 1 = CardBus parity errors are reported using $\overline{\text{CPERR}}$.

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

subsystem vendor ID register

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

Register: **Subsystem vendor ID**
Type: Read only (read/write when bit 5 in the system control register is 0)
Offset: 40h (functions 0, 1)
Default: 0000h
Description: The subsystem vendor ID register is used for system and option-card identification purposes, and may be required for certain operating systems. This register is read only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register. When bit 5 is 0, this register is read/write; when bit 5 is 1, this register is read only. The default mode is read only.

subsystem ID register

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

Register: **Subsystem ID**
Type: Read only (read/write when bit 5 in the system control register is 0)
Offset: 42h (functions 0, 1)
Default: 0000h
Description: The subsystem ID register is used for system and option-card identification purposes and may be required for certain operating systems. This register is read only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register. When bit 5 is 0, this register is read/write; when bit 5 is 1, this register is read only. The default mode is read only.

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

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

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

Register: **PC Card 16-bit I/F legacy-mode base address**
Type: Read only, read/write (see individual bit descriptions)
Offset: 44h (functions 0, 1)
Default: 0000 0000h
Description: The PCI1251A 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 *PCI to PCMCIA CardBus Bridge Register Description* (Yenta), this register is shared by functions 0 and 1. Refer to the ExCA register set description for register offsets.

system control register

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

Register: **System control**
Type: Read only, read/write (see individual bit descriptions)
Offset: 80h (functions 0, 1)
Default: 0044 9060h
Description: System-level initializations are performed through programming this doubleword register. Some of the bits are global and should be written only through function 0. Refer to Table 32 for a complete description of the register contents.

Table 32. System Control Register

BIT	SIGNAL	TYPE	FUNCTION
31–30†	SER_STEP	R/W	Serialized PCI interrupt routing step. Bits 31–30 are used to configure the serialized PCI interrupt stream signaling, and accomplish an even distribution of interrupts signaled on the four PCI interrupt slots. Bits 31–30 are global to both PCI1251A 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 this bit is set, the INTA and INTB signals are tied together internally and are signaled as INTA. INTA can then be shifted by using the SER_STEP bits. This bit is global to both PCI1251A 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 is read only and returns 0 when read.
27†	P2CCLK	R/W	P^2C power switch clock. The PCI1251A defaults CLOCK as an input clock to control the serial interface and the internal state machine. Bit 27 can be set to enable the PCI1251A to generate and drive the CLOCK from the PCI clock. When in a SUSPEND state, however, CLOCK must be input to the PCI1251A to successfully power down sockets after card removal without indicating to the system the removal event. 0 = CLOCK provided externally, input to PCI1251A (default) 1 = CLOCK generated by PCI clock and driven by PCI1251A
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 a write occurs to set the socket power, and the SMIENB bit is set. Writing a 1 to bit 25 clears the status. 0= SMI interrupt signaled 1 = SMI interrupt not signaled
24†	SMIENB	R/W	SMI interrupt mode enable. When bit 24 is set, the SMI interrupt signaling is enabled and 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
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 this bit is low (as default), these signals are 3-stated. 0 = 3-state CardBus RSVD 1 = Drive Cardbus RSVD low (default)
21	VCCPROT	R/W	V_{CC} protection enable. Bit 21 is socket dependent. 0 = V_{CC} protection enabled for 16-bit cards (default) 1 = V_{CC} protection disabled for 16-bit cards
20	RSVD	R/W	Reserved. Bit 20 returns 0 when read.
19	CDREQEN	R/W	PC/PCI DMA card enable. When bit 19 is set, the PCI1251A allows 16-bit PC Cards to request PC/PCI DMA using the DREQ signaling. DREQ is selected through the socket DMA register 0. 0 = Ignore DREQ signaling from PC Cards (default) 1 = Signal DMA request on DREQ
18–16	CDMACHAN	R/W	CDMACHANPC/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 = MRBURSTDN downstream is disabled. 1 = MRBURSTDN downstream is enabled (default).

† These bits are global and should be accessed only through function 0.

Table 32. System Control Register (Continued)

BIT	SIGNAL	TYPE	FUNCTION
14†	MRBURSTUP	R/W	Memory read burst enable upstream. When bit 14 is set, the PCI1251A allows memory read transactions to burst upstream. 0 = MRBURSTUP upstream is disabled (default). 1 = MRBURSTUP upstream is enabled.
13	SOCACTIVE	R	Socket activity status. When set, 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. Bit 12 is read only and returns 1 when read. This is the power-rail bit in functions 0 and 1.
11†	PWRSTREAM	R	Power stream in progress status bit. When set high, bit 11 indicates that a power stream to the power switch is in progress and a powering change has been requested. When this bit is clear, it indicates that the power stream is complete. 0 = Power stream is complete and delay has expired. 1 = Power stream is in progress.
10†	DELAYUP	R	Power-up delay in progress status. When set, bit 9 indicates that a power-up stream has been sent to the power switch and proper power may not yet be stable. This bit is cleared when the power-up delay has expired. 0 = Power-up delay has expired. 1 = Power-up stream sent to switch. Power may not be stable.
9†	DELAYDOWN	R	Power-down delay in progress status. When set, bit 10 indicates that a power-down stream has been sent to the power switch and proper power may not yet be stable. This bit is cleared when the power-down delay has expired. 0 = Power-down delay has expired. 1 = Power-down stream sent to switch. Power may not be stable.
8	INTERROGATE	R	Interrogation in progress. When set, bit 8 indicates an interrogation is in progress and clears when interrogation completes. This bit is socket dependent. 0 = Interrogation not in progress (default) 1 = Interrogation in progress
7	RSVD	R	Reserved. Bit 7 is read only and returns 0 when read.
6†	PWRSAVINGS	R/W	Power savings mode enable. When this bit is set, if a CB card is inserted, idle, and without a CB clock, the applicable CB state machine will not be clocked.
5†	SUBSYSRW	R/W	Subsystem ID (SSID), subsystem vendor ID (SSVID), ExCA ID, and revision register read/write enable. Bit 5 is shared by functions 0 and 1. 0 = SSID, SSVID, ExCA ID, and revision register are read/write. 1 = SSID, SSVID, ExCA ID, and revision register 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 $\overline{\text{SERR}}$ (default) 1 = CardBus data parity signaled on PCI $\overline{\text{SERR}}$
3†	CDMA_EN	R/W	PC/PCI DMA enable. Bit 3 enables PC/PCI DMA when set, and disables IRQMUX7 and IRQMUX6 signaling. 0 = Centralized DMA disabled (default) 1 = Centralized DMA enabled
2	RSVD	R	Reserved
1†	KEEPCLK	R/W	Keep clock. This bit works with PCI and CB $\overline{\text{CLKRUN}}$ protocols. 0 = Allows normal functioning of both $\overline{\text{CLKRUN}}$ protocols.(default) 1 = Does not allow CB clock or PCI clock to be stopped using the $\overline{\text{CLKRUN}}$ protocols.
0†	RIMUX	R/W	$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ multiplex enable. 0 = RI_OUT and PME signals are both routed to the $\overline{\text{RI\_OUT}}$/PME terminal. If both are enabled at the same time, $\overline{\text{RI\_OUT}}$ will have precedence over PME. 1 = Only PME signals are routed to the $\overline{\text{RI\_OUT}}$/PME terminal.

† These bits are global and should be accessed only through function 0.

multimedia control register

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

Register: **Multimedia control**
Type: Read only, read/write (see individual bit descriptions)
Offset: 84h (functions 0, 1)
Default: 00h
Description: The multimedia control register provides port mapping for the PCI1251A zoom video/data ports (see *zoom video support*, on page 29). Access this register only through function 0. Refer to Table 33 for a complete description of the register contents.

Table 33. Multimedia Control Register

BIT	SIGNAL	TYPE	FUNCTION
7	ZVOUTEN	R/W	ZV output enable. Bit 7 enables the output for the PCI1251A outsourcing ZV terminals. When this bit is reset 0, these terminals are in a high-impedance state. 0 = PCI1251A ZV output terminals disabled (default) 1 = PCI1251A 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 PCI1251A ZV port. 0 = Output card 0 ZV if enabled (default) 1 = Output card 1 ZV if enabled
5–2	RSVD	R	Reserved. Bits 5–2 return 0s when read. Writes have no effect.
1	ZVEN1	R/W	PC Card 1 ZV mode enable. Bit 1 enables the zoom video mode for socket 1. When set, the PCI1251A 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. Bit 0 enables the zoom video mode for socket 0. When set, the PCI1251A 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

general status register

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

Register: **General status**
Type: Read only
Offset: 85h (functions 0, 1)
Default: 0Xh
Description: The general status register provides general device status information. The status of the serial EEPROM interface is provided through this register. Refer to Table 34 for a complete description of the register contents.

Table 34. General Status Register

BIT	SIGNAL	TYPE	FUNCTION
7–3	RSVD	R	Reserved. Bits 7–3 are read only and return 0s when read.
2†	EEDETECT	R	Serial EEPROM detect. When bit 2 is cleared, it indicates that the PCI1251A serial EEPROM circuitry has detected an EEPROM. A pullup resistor must be implemented on LATCH for bit 2 to be set. This status bit is encoded as: 0 = EEPROM not detected (default) 1 = EEPROM detected
1†	DATAERR	R	Serial EEPROM data error status. Bit 1 indicates when a data error occurs on the serial EEPROM interface. Bit 2 may be set due to a missing acknowledge. This bit is cleared by writing a 1. 0 = No error detected (default) 1 = Data error detected
0†	EEBUSY	R	Serial EEPROM busy status. Bit 0 indicates the status of the PCI1251A serial EEPROM circuitry. This bit is set during the loading of the subsystem ID value. 0 = Serial EEPROM circuitry not busy (default) 1 = Serial EEPROM circuitry busy

† These bits are global and should be accessed only through function 0.

GPIO0 control register

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

Register: **GPIO0 control**
Type: Read only, read/write (see individual bit descriptions)
Offset: 88h (functions 0, 1)
Default: 80h
Description: The GPIO0 control register is used for control of the general-purpose I/O 0 (GPIO0). This terminal defaults to a general-purpose input but can be reconfigured as the socket 0 activity LED output, a zoom video enabled status output, or general-purpose output. Access this register only through function 0. Refer to Table 35 for a complete description of the register contents.

Table 35. GPIO0 Control Register

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

GPIO1 control register

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

Register: **GPIO1 control**
Type: Read only, read/write (see individual bit descriptions)
Offset: 89h (functions 0, 1)
Default: 80h
Description: The GPIO1 control register is used for control of the general-purpose I/O 1 (GPIO1). This terminal defaults to a general-purpose input, but can be reconfigured as the socket 1 activity LED output or general-purpose output. Access this register only through function 0. Refer to Table 36 for a complete description of the register contents.

Table 36. GPIO1 Control Register

BIT	SIGNAL	TYPE	FUNCTION
7–6	GP1	R/W	General-purpose 1 mode. Bits 7–6 select the functionality of LEDA2/GPIO1. These bits are encoded as: 00 = Signal LEDA2 to indicate PC Card socket 1 activity. 01 = Reserved. 10 = General-purpose input (GPI). 11 = General-purpose output (GPO).
5	RSVD	R	Reserved. Bit 5 returns 0 when read. Writes have no effect.
4	GPINTEN1	R/W	GP interrupt enable. When bit 4 is set, a socket A card status change (CSC) interrupt is generated when the DELTA1 bit is set.
3	DELTA1	R/W	DATAIN1 change status. Bit 3 is set when the DATAIN1 bit changes state when in GPI mode. Glitches on the GPI terminal may not be detected by software without bit 3. This bit is cleared by a write back of 1.
2	RSVD	R	Reserved. Bit 2 returns 0 when read. Writes have no effect.
1	DATAOUT1	R/W	General-purpose data output. When in general-purpose output mode, bit 1 represents the data. Data written to this bit in GPO mode is signaled to the output.
0	DATAIN1	R/W	General-purpose data input. When in either general-purpose input or output mode, bit 0 represents the data on the GPIO terminal. Data signaled on the GPI terminal is identified through this bit.

GPIO2 control register

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

Register: **GPIO2 control**
Type: Read only, read/write (see individual bit descriptions)
Offset: 8Ah (functions 0, 1)
Default: 80h
Description: The GPIO2 control register is used for control of the general-purpose I/O 2 (GPIO2). This terminal defaults to a general-purpose input but can be reconfigured as PCI $\overline{\text{LOCK}}$, a zoom video enabled status output, or general-purpose output. Access this register only through function 0. Refer to Table 37 for a complete description of the register contents.

Table 37. GPIO2 Control Register

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

GPIO3 control register

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

Register: **GPIO3 control**
Type: Read only, read/write (see individual bit descriptions)
Offset: 8Bh (functions 0, 1)
Default: 80h
Description: The GPIO3 control register is used for control of the general-purpose I/O 3 (GPIO3). This terminal defaults to a general-purpose input but can be reconfigured as PCI $\overline{\text{INTA}}$ or general-purpose output. Access this register only through function 0. Refer to Table 38 for a complete description of the register contents.

Table 38. GPIO3 Control Register

BIT	SIGNAL	TYPE	FUNCTION
7–6	GP3	R/W	General-purpose 3 mode. Bits 7–6 select the functionality of $\overline{\text{INTA}}$/GPIO3. These bits are encoded as: 00 = Terminal is configured as PCI $\overline{\text{INTA}}$. 01 = Reserved 10 = General-purpose input (GPI) 11 = General-purpose output (GPO)
5	RSVD	R	Reserved. Bit 5 returns 0 when read. Writes have no effect.
4	GPINTEN3	R/W	GP interrupt enable. When bit 4 is set, a socket B card status change (CSC) interrupt is generated when the DELTA3 bit is set.
3	DELTA3	R/W	DATAIN3 change status. Bit 3 is set when the DATAIN3 bit changes state when in GPI mode. Glitches on the GPI terminal may not be detected by software without bit 3. This bit is cleared by a write back of 1.
2	RSVD	R	Reserved. Bit 2 returns 0 when read. Writes have no effect.
1	DATAOUT3	R/W	General-purpose data output. When in general-purpose output mode, bit 1 represents the data. Data written to this bit in GPO mode is signaled to the output.
0	DATAIN3	R/W	General-purpose data input. When in either general-purpose input or output mode, bit 0 represents the data on the GPIO terminal. Data signaled on the GPI terminal is identified through this bit.

IRQMUX routing register

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

Register: **IRQMUX routing**
Type: Read/write (see individual bit descriptions)
Offset: 8Ch (functions 0, 1)
Default: 0000 0000h
Description: The IRQMUX routing register is used for the legacy interrupt mux routing feature of the PCI1251A, which is described in the p*rogrammable interrupt subsystem* on page 32. If the parallel IRQ interrupt scheme is selected, all PCI1251A interrupts sent to ISA IRQs are signaled on the corresponding IRQMUX7–IRQMUX0 signals. These signals are routed directly to various IRQ inputs on the system PIC, and the routing information is programmed through this register. Each terminal has at least one secondary function that can be selected by programming the bits appropriately. Access this register only through function 0. Refer to Table 39 for a complete description of the register contents.

Table 39. IRQMUX Routing Register

BIT	SIGNAL	TYPE	FUNCTION
31–28	IRQMUX7	R/W	IRQMUX7 routing. Bits 31–28 select one of 15 interrupts that may be routed on IRQMUX7. When these bits are 0000 and bit 3 in the system control register is set, this pin is used for $\overline{\text{PCREQ}}$ DMA signaling. NOTE: These bits must not be configured for IRQ signaling if IRQMUX7 is being used for $\overline{\text{PCREQ}}$ signaling. 0000 = EEPROM SCL routed on IRQMUX7 (default) 0000 = $\overline{\text{PCREQ}}$ routed on IRQMUX7 when bit 3 of the system control register is 1 0001 = $\overline{\text{PCREQ}}$ routed on IRQMUX7 0010 = IRQ2 routed on IRQMUX7 0011 = IRQ3 routed on IRQMUX7 : 1111 = IRQ15 routed on IRQMUX7
27–24	IRQMUX6	R/W	IRQMUX6 routing. Bits 27–24 select one of 15 interrupts that may be routed on IRQMUX6. When these bits are 0000 and bit 3 in the system control register is set, this pin is used for $\overline{\text{PCGNT}}$ DMA signaling. NOTES: 1. These bits must not be configured for IRQ signaling if IRQMUX6 is being used for $\overline{\text{PCGNT}}$ signaling. 2. An EEPROM cannot be used if IRQMUX7 and IRQMUX6 are being used for DMA $\overline{\text{PCREQ}}$ $\overline{\text{PCGNT}}$. 0000 = EEPROM SDA routed on IRQMUX6 (default) 0000 = $\overline{\text{PCGNT}}$ routed on IRQMUX6 when bit 3 of the system control register is 1 0001 = IRQ1 routed on IRQMUX6 0010 = IRQ2 routed on IRQMUX6 0011 = IRQ3 routed on IRQMUX6 : 1111 = IRQ15 routed on IRQMUX6

Table 39. IRQMUX Routing Register (Continued)

BIT	SIGNAL	TYPE	FUNCTION
23–20	IRQMUX5	R/W	IRQMUX5 routing. Bits 23–20 select one of 15 interrupts that may be routed on IRQMUX5. When these bits are 0000, then no routing is selected. 0000 = No IRQ routing selected (default) 0001 = CardBus audio (CBAUDIO) routed on IRQMUX5 0010 = IRQ2 routed on IRQMUX5 0011 = IRQ3 routed on IRQMUX5 : 1111 = IRQ15 routed on IRQMUX5
19–16	IRQMUX4	R/W	IRQMUX4 routing. Bits 19–16 select one of 15 interrupts that may be routed on IRQMUX4. When these bits are 0000, then no routing is selected. 0000 = No IRQ routing selected (default) 0001 = $\overline{\text{ZVSTAT}}$ routed on IRQMUX4 0010 = $\overline{\text{RI\_OUT}}$ routed on IRQMUX4 0011 = IRQ3 routed on IRQMUX4 0100 = IRQ4 routed on IRQMUX4 : 1111 = IRQ15 routed on IRQMUX4
15–12	IRQMUX3	R/W	IRQMUX3 routing. Bits 15–12 select one of 15 interrupts that may be routed on IRQMUX3. When these bits are 0000, then no routing is selected. 0000 = No IRQ routing selected (default) 0001 = LEDA or LEDB routed on IRQMUX3 0010 = $\overline{\text{RI\_OUT}}$ routed on IRQMUX3 0011 = IRQ3 routed on IRQMUX3 0100 = IRQ4 routed on IRQMUX3 : 1111 = IRQ15 routed on IRQMUX3
11–8	IRQMUX2	R/W	IRQMUX2 routing. Bits 11–8 select one of 15 interrupts that may be routed on IRQMUX2. When these bits are 0000, then no routing is selected. 0000 = No IRQ routing selected (default) 0001 = LEDB routed on IRQMUX2 0010 = IRQ2 routed on IRQMUX2 0011 = IRQ3 routed on IRQMUX2 : 1111 = IRQ15 routed on IRQMUX2
7–4	IRQMUX1	R/W	IRQMUX1 routing. Bits 7–4 select one of 15 interrupts that may be routed on IRQMUX1. When these bits are 0000, then no routing is selected. 0000 = No IRQ routing selected (default) 0001 = LEDA routed on IRQMUX1 0010 = IRQ2 routed on IRQMUX1 0011 = IRQ3 routed on IRQMUX1 : 1111 = IRQ15 routed on IRQMUX1
3–0	IRQMUX0	R/W	IRQMUX0 routing. Bits 3–0 select one of 15 interrupts that may be routed on IRQMUX0. When these bits are 0000, then no routing is selected. 0000 = No IRQ routing selected (default) 0001 = $\overline{\text{INTB}}$ routed on IRQMUX0 0010 = IRQ2 routed on IRQMUX0 0011 = IRQ3 routed on IRQMUX0 : 1111 = IRQ15 routed on IRQMUX0

retry status register

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

Register: **Retry status**
Type: Read only, read/write (see individual bit descriptions)
Offset: 90h (functions 0, 1)
Default: C0h
Description: The retry status register enables the retry timeout counters and displays the retry expiration status. The flags are set when the PCI1251A retries a PCI or CardBus master request, and the master does not return within 2^{15} PCI clock cycles. The flags are cleared by writing a 1 to the bit. These bits are expected to be incorporated into the PCI command, PCI status, and bridge control registers by the PCI SIG. Access this register only through function 0. Refer to Table 40 for a complete description of the register contents.

Table 40. Retry Status Register

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

† These bits are global and should be accessed only through function 0.

card control register

Bit	7	6	5	4	3	2	1	0
Name	Card control							
Type	R/W	R/W	R	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 (see individual bit descriptions)
Offset: 91h
Default: 00h
Description: The card control register is provided for PCI1130 compatibility. It provides the PC Card function interrupt flag (IFG) and an alias for the ZVEN0 and ZVEN1 bits found in the PCI1251A multimedia control register. When this register is accessed by function 0, the ZVEN0 bit will alias with ZVENABLE. When this register is accessed by function 1, the ZVEN1 bit will alias with ZVENABLE. Setting ZVENABLE only places the PC Card socket interface ZV terminals in a high-impedance state, but does not enable the PCI1251A to drive ZV data onto the ZV terminals. RI_OUT is enabled through this register, and the enable bit is shared between functions 0 and 1. Refer to Table 41 for a complete description of the register contents.

Table 41. Card Control Register

BIT	SIGNAL	TYPE	FUNCTION
7†	RIENB	R/W	Ring indicate output enable. 0 = Disables any routing of $\overline{\text{RI\_OUT}}$ signal (default). 1 = Enables $\overline{\text{RI\_OUT}}$ signal for routing to the $\overline{\text{RI\_OUT/PME}}$ terminal when RIMUX is set to 0, and for routing to IRQMUX3/4.
6	ZVENABLE	R/W	Compatibility ZV mode enable. When set, the corresponding PC Card Socket interface ZV terminals enter a high-impedance state. This bit defaults to 0.
5	RSVD	R	Reserved. Bit 5 returns 0 when read.
4–3	RSVD	R	Reserved. Bits 4–3 are read only and default to 0.
2	AUD2MUX	R/W	CardBus Audio-to-IRQMUX. When set, the CAUDIO CardBus signal is routed to the corresponding IRQMUX terminal. Function 0, A_CAUDIO is routed to IRQMUX0, and function 1, B_AUDIO is routed to IRQMUX1. If this bit is set for both functions, function 0 is routed. 0 = CAUDIO set to CAUDPWM on IRQMUX routed (default) 1 = CAUDIO is not routed.
1	SPKROUTEN	R/W	Speaker out enable. When bit 1 is set, $\overline{\text{SPKR}}$ on the PC Card is enabled and is routed to SPKROUT on the PCI bus. The $\overline{\text{SPKR}}$ signal from socket 0 is exclusive ORed with the $\overline{\text{SPKR}}$ signal from socket 1 and sent to SPKROUT. The SPKROUT terminal drives data only when either functions SPKROUTEN bit is set. This bit is encoded as: 0 = $\overline{\text{SPKR}}$ to SPKROUT not enabled 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. Bit 0 is set when a functional interrupt is signaled from a PC Card interface and is socket dependent (i.e., not global). Write back a 1 to clear this bit. 0 = No PC Card functional interrupt detected (default) 1 = PC Card functional interrupt detected

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

device control register

Bit	7	6	5	4	3	2	1	0
Name	Device control							
Type	R	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 (see individual bit descriptions)
Offset: 92h (functions 0, 1)
Default: 66h
Description: The device control register is provided for PCI1130 compatibility and contains bits that are shared between functions 0 and 1. The interrupt mode select is programmed through this register which is composed of PCI1251A global bits. The socket-capable force bits are also programmed through this register. Refer to Table 42 for a complete description of the register contents.

Table 42. Device Control Register

BIT	SIGNAL	TYPE	FUNCTION
7	RSVD	R	Reserved. Bit 7 Returns 0 when read.
6†	3VCAPABLE	R/W	3-V socket capable force 0 = Not 3-V capable 1 = 3-V capable (default)
5	IO16R2	R/W	Diagnostic bit. This bit defaults to 1.
4	RSVD	R	Reserved. Bit 4 returns 0 when read. Writes have no effect.
3†	TEST	R/W	TI test. Only a 0 should be written to bit 3. This bit can be set to shorten the interrogation counter.
2–1†	INTMODE	R/W	Interrupt mode. Bit 2–1 select the interrupt signaling mode. The interrupt mode bits are encoded: 00 = Parallel PCI interrupts only 01 = Parallel IRQ and parallel PCI interrupts 10 = IRQ serialized interrupts and parallel PCI interrupts $\overline{\text{INTA}}$ and $\overline{\text{INTB}}$ 11 = IRQ and PCI serialized interrupts (default)
0†	RSVD	R/W	Reserved. NAND tree enable bit. There is a NAND tree diagnostic structure in the PCI1251A, and it tests only the pins that are inputs or I/Os. Any output-only terminal on the PCI1251A is excluded from the NAND tree test.

† These bits are global and should be accessed only through function 0.

diagnostic register

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

Register: **Diagnostic**
Type: Read/write
Offset: 93h (functions 0, 1)
Default: 61h
Description: The diagnostic register is provided for internal TI test purposes. It is a read/write register, but only 0s should be written to this register. Refer to Table 43 for a complete description of the register contents.

Table 43. Diagnostic Register

BIT	SIGNAL	TYPE	FUNCTION
7†	TRUE_VAL	R/W	True value. This bit defaults to 0 when read. This bit is encoded as: 0 = Reads true values in PCI vendor ID and PCI device ID registers (default) 1 = Reads all 1s in reads to the PCI vendor ID and PCI device ID registers
6-5	RSVD	R/W	Reserved. These bits are reserved for TI internal test purposes. The value of these bits should not be changed for normal operation.
4†	DIAG	R/W	Diagnostic RETRY_DIS. Delayed transaction disable.
3†	DIAG	R/W	Diagnostic RETRY_EXT. Extends the latency from 16 to 64.
2†	DIAG	R/W	Diagnostic DISCARD_TIM_SEL_CB. Set = 2^{10}, reset = 2^{15}.
1†	DIAG	R/W	Diagnostic DISCARD_TIM_SEL_PCI. Set = 2^{10}, reset = 2^{15}.
0†	ASYNC_CSC	R/W	Asynchronous interrupt generation. 0 = CSC interrupt not generated asynchronously 1 = CSC interrupt generated asynchronously (default)

† These bits are global and should be accessed only through function 0.

socket DMA register 0

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

Register: **Socket DMA register 0**
Type: Read only, read/write (see individual bit descriptions)
Offset: 94h (functions 0, 1)
Default: 0000 0000h
Description: The socket DMA register 0 provides control over the PC Card DMA request ($\overline{\text{DREQ}}$) signaling. Refer to Table 44 for a complete description of the register contents.

Table 44. Socket DMA Register 0

BIT	SIGNAL	TYPE	FUNCTION
31–2	RSVD	R	Reserved. Bits 31–2 are read only and return 0s when read.
1–0	DREQPIN	R/W	DMA request ($\overline{\text{DREQ}}$). Bits 1–0 indicate which pin on the 16-bit PC Card interface acts as $\overline{\text{DREQ}}$ during DMA transfers. This field is encoded as: 00 = 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 $\overline{\text{INPACK}}$.

socket DMA register 1

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

Register: **Socket DMA register 1**
Type: Read only, read/write (see individual bit descriptions)
Offset: 98h (functions 0, 1)
Default: 0000 0000h
Description: The socket DMA register 1 provides control over the distributed DMA (DDMA) registers and the PCI portion of DMA transfers. The DMA base address locates the DDMA registers in a 16-byte region within the first 64K-bytes of PCI I/O address space. Refer to Table 45 for a complete description of the register contents.

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

Table 45. Socket DMA Register 1

BIT	SIGNAL	TYPE	FUNCTION
31–16	RSVD	R	Reserved. Bits 31–16 are read only and return 0s when read.
15–4	DMABASE	R/W	DMA base address. Locates the socket's DMA registers in PCI I/O space. This field represents a 16-bit PCI I/O address. The upper 16 bits of the address are hardwired to 0, forcing this window to within the lower 64K-bytes of I/O address space. The lower four bits are hardwired to 0 and are included in the address decode. Thus, the window is aligned to a natural 16-byte boundary.
3	EXTMODE	R	Extended addressing. This feature is not supported by the PCI1251A and always returns a 0.
2–1	XFERSIZE	R/W	Transfer size. Bits 2–1 specify the width of the DMA transfer on the PC Card interface and are encoded as: 00 = Transfers are 8 bits (default). 01 = Transfers are 16 bits. 10 = Reserved 11 = Reserved
0	DDMAEN	R/W	DDMA registers decode enable. Enables the decoding of the distributed DMA registers based on the value of DMABASE. 0 = Disabled (default) 1 = Enabled

capability ID register

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

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

next-item pointer register

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

Register: **Next-item pointer**
Type: Read only
Offset: A1h
Default: 00h
Description: The next-item pointer register is used to indicate the next item in the linked list of the PCI power management capabilities. Because the PCI1251A functions only include one capabilities item, this register returns 0s when read.

power-management capabilities register

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

Register: **Power-management capabilities**
Type: Read only (see individual bit descriptions)
Offset: A2h (functions 0, 1)
Default: 7E21h
Description: The power-management capabilities register contains information on the capabilities of the PC Card function related to power management. Both PCI1251A CardBus bridge functions support D0, D2, and D3 power states. Refer to Table 46 for a complete description of the register contents.

Table 46. Power-Management Capabilities Register

BIT	SIGNAL	TYPE	FUNCTION
15–11	PME_SUP	R	PME support. This 4-bit field indicates the power states from which the PCI1251A supports asserting $\overline{\text{PME}}$. A 0 for any bit indicates that the CardBus function cannot assert $\overline{\text{PME}}$ from that power state. These four bits return 1110b when read. Each of these bits is described below: Bit 15 contains the value 0, indicating that $\overline{\text{PME}}$ cannot be asserted from $D3_{cold}$ state. Bit 14 contains the value 1, indicating that $\overline{\text{PME}}$ can be asserted from $D3_{hot}$ state. Bit 13 contains the value 1, indicating that $\overline{\text{PME}}$ can be asserted from D2 state. Bit 12 contains the value 1, indicating that $\overline{\text{PME}}$ can be asserted from D1 state. Bit 11 contains the value 1, indicating that $\overline{\text{PME}}$ can be asserted from the D0 state.
10	D2_SUP	R	D2 support. Bit 10 returns a 1 when read, indicating that the CardBus function supports the D2 device power state.
9	D1_SUP	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. These bits are reserved and return 000b when read.
5	DSI	R	Device-specific initialization. Bit 5 is read only and returns 1 when read, indicating that the CardBus controller functions require special initialization (beyond the standard PCI configuration header) before the generic class device driver is able to use it.
4	AUX_PWR	R	Auxiliary power source. This bit is meaningful only if bit 15 ($D3_{cold}$ supporting $\overline{\text{PME}}$) is set. When set, bit 4 indicates that support for $\overline{\text{PME}}$ in D3cold requires auxiliary power. 0 = Function supplies its own auxiliary power source 1 = Support for $\overline{\text{PME}}$ in $D3c_{old}$ requires auxiliary power supplied to the system by a proprietary source.
3	PMECLK	R	$\overline{\text{PME}}$ clock. When set, bit 3 indicates that the function relies on the presence of the PCI clock for $\overline{\text{PME}}$ operation. 0 = PCI clock not required for the function to generate $\overline{\text{PME}}$ 1 = PCI function required to generate $\overline{\text{PME}}$
2–0		R	Version. Bits 2–0 return 001b when read, indicating that there are four bytes of general-purpose power management (PM) registers as described in the PCI Bus Power Management Interface Specification Revision 1.0.

power-management control/status register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Power-management control/status															
Type	R/W	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 (see individual bit descriptions)
Offset: A4h (functions 0, 1)
Default: 000000h
Description: The power-management control/status register determines and changes the current power state of the PCI1251A CardBus function. The contents of this register are not affected by the internally-generated reset caused by the transition from $D3_{hot}$ to D0 state. Refer to Table 47 for a complete description of the register contents.

NOTE:
A transition from the $D3_{hot}$ state to D0 state resets all PCI, ExCA, and CardBus registers. TI specific, PCI power management, and legacy base address registers are not affected.

Table 47. Power-Management Control/Status Register

BIT	SIGNAL	TYPE	FUNCTION
15	PMESTAT	R/W	PME status. Bit 15 is set when the CardBus function would normally assert $\overline{PME}$, independent of the state of the PME_EN bit. 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 is read only, returning 0s when read. The CardBus function does not return any dynamic data as indicated by the DYN_DATA bit.
12–9	DATASEL	R	Data select. This 4-bit field is read only and returns 0s when read. The CardBus function does not return any dynamic data as indicated by the DYN_DATA bit.
8	PME_EN	R/W	PME enable. Bit 8 enables the function to assert $\overline{PME}$. If this bit is cleared, assertion of $\overline{PME}$ is disabled.
7–2	RSVD	R	Reserved. Bits 7–2 are read only and 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 = $D3_{hot}$

power-management control/status register bridge support extensions

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

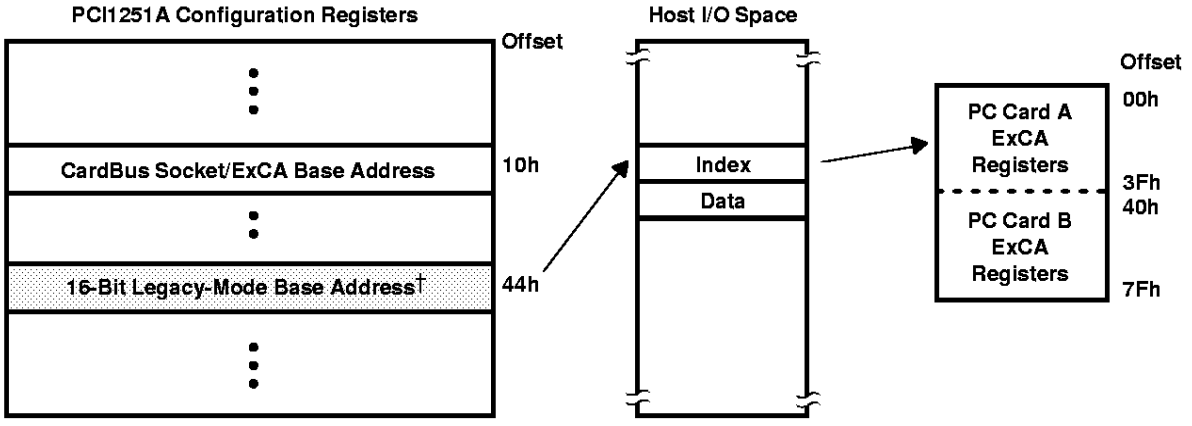
Register: **Power-management control/status register bridge support extensions**
Type: Read only
Offset: A6h (functions 0, 1)
Default: 80h
Description: The power-management control/status register bridge support extensions support PCI bridge specific functionality and are required for all PCI-to-PCI bridges. Refer to Table 48 for a complete description of the register contents.

Table 48. Power-Management Control/Status Register Bridge Support Extensions

BIT	SIGNAL	TYPE	FUNCTION
7	BPCC_EN	R	Bus power/clock control. When read, bit 7 returns 1b. This bit is encoded as: 0 = Bus power/clock control is disabled. 1 = Bus power/clock control is enabled (default). A 0 indicates that the bus power/clock control policies defined in the PCI Power Management specification are disabled. When the bus power/clock control enable mechanism is disabled the bridge's PMCSR PowerState field cannot be used by the system software to control the power or the clock of the bridge's secondary bus. A 1 indicates that the bus power/clock control mechanism is enabled.
6	$\overline{\text{B2\_B3}}$	R	B2/B3 support for $D3_{hot}$. The state of this bit determines the action that is to occur as a direct result of programming the function to $D3_{hot}$. This bit is meaningful only if bit 7 BPCC_Enable is a 1. This bit is encoded as: 0 = When the bridge is programmed to $D3_{hot}$, its secondary bus has its power removed (B3) 1 = When the bridge is programmed to $D3_{hot}$, its secondary bus's PCI clock is stopped (B2).
5–0	RSVD	R	Reserved. These bits are read only and return 0s when read.

ExCA compatibility registers (functions 0 and 1)

The ExCA (Exchangeable Card Architecture) registers implemented in the PCI1251A are register-compatible with the popular Intel 82365SL–DF PCMCIA controller. ExCA registers are identified by an offset value that is compatible with the legacy I/O index/data scheme used on the Intel 82365 ISA controller. The ExCA registers are accessed through this scheme by writing the register offset value into the index register (I/O base) and reading or writing the data register (I/O base + 1). The I/O base address used in the index/data scheme is programmed in the PC Card 16-Bit I/F legacy mode base address register, which is shared by both card sockets. The offsets from this base address run contiguous from 00h to 3Fh for socket A, and from 40h to 7Fh for socket B. Refer to Figure 16 for an ExCA I/O mapping illustration. Table 49 identifies each ExCA register and its respective ExCA offset.

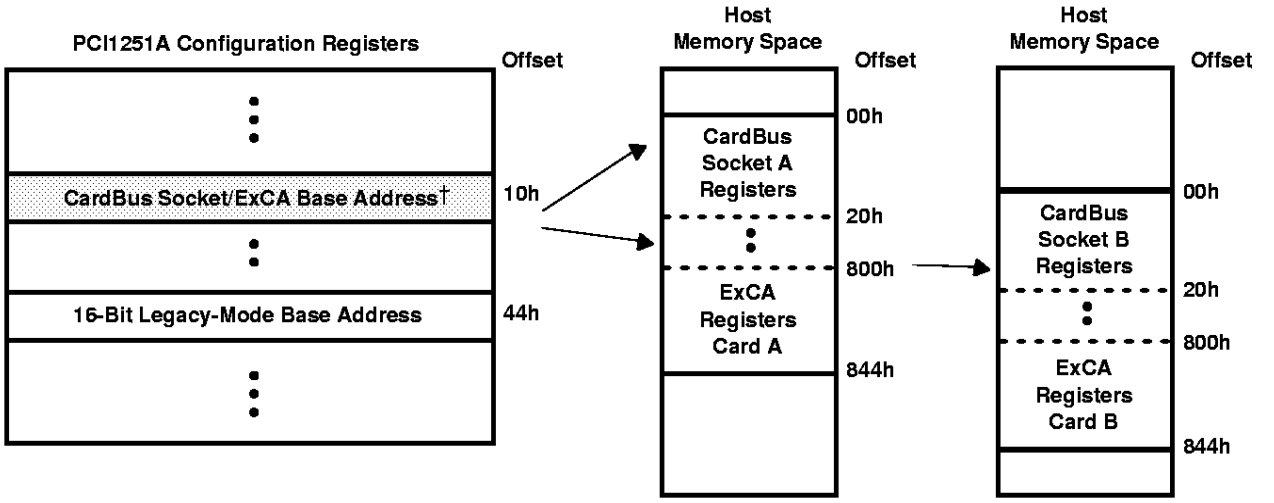


† The 16-bit legacy mode base address register is shared by functions 0 and 1 as indicated by the shading.

Figure 16. ExCA Register Access Through I/O

The TI PCI1251A also provides a memory mapped alias of the ExCA registers by directly mapping them into PCI memory space. They are located through the CardBus Socket Registers/ExCA Registers Base Address Register (PCI Register 10h) at memory offset 800h. Each socket has a separate base address programmable by function. Refer to Figure 17 for an ExCA memory mapping illustration. The memory offsets are 800h–844h for both functions 0 and 1. This illustration also identifies the CardBus Socket Register mapping, which are mapped into the same 4 K window at memory offset 0h.

ExCA compatibility registers (functions 0 and 1) (continued)



† The CardBus socket/ExCA base address mode register is separate for functions 0 and 1.

Figure 17. ExCA Register Access Through Memory

The interrupt registers, as defined by the 82365SL Specification, in the ExCA register 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 PCI1251A to ensure that all possible PCI1251A 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.

Table 49. ExCA Registers and Offsets

REGISTER NAME	PCI MEMORY ADDRESS OFFSET (HEX)	ExCA OFFSET (HEX)	
		CARD A	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

Table 49. ExCA Registers and Offsets (Continued)

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

ExCA identification and revision register† (index 00h)

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

Register: **ExCA identification and revision**
Type: Read only, read/write (see individual bit descriptions)
Offset: CardBus socket address + 800h; Card A ExCA offset 00h
Card B ExCA offset 40h
Default: 84h
Description: This register provides host software with information on 16-bit PC Card support and Intel 82365SL-DF compatibility. Refer to Table 50 for a complete description of the register contents.

† When bit 5 in the system control register is 1, this register is read only.

Table 50. ExCA Identification and Revision Register (Index 00h)

BIT	SIGNAL	TYPE	FUNCTION
7–6	IFTYPE	R/W	Interface type. These bits default to 10b and identify the 16-bit PC Card support provided by the PCI1251A. The PCI1251A supports both I/O and memory 16-bit PC cards.
5–4	RSVD	R/W	Reserved. Bits 5–4 can be used for Intel 82365SL-DF emulation.
3–0	365REV	R/W	Intel 82365SL-DF revision. This read/write field stores the Intel 82365SL-DF revision supported by the PCI1251A. Host software can read this field to determine compatibility to the Intel 82365SL-DF register set. This field defaults to 0100b upon PCI1251A reset.

ExCA interface status register (index 01h)

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

Register: **ExCA interface status**
Type: Read only (see individual bit descriptions)
Offset: CardBus socket address + 801h; Card A ExCA offset 01h
Card B ExCA offset 41h
Default: 00XX XXXXb
Description: This register provides information on the current status of the PC Card interface. An X in the default bit value indicates that the value of the bit after reset depends on the state of the PC Card interface. Refer to Table 51 for a complete description of the register contents.

Table 51. ExCA Interface Status Register (Index 01h)

BIT	SIGNAL	TYPE	FUNCTION
7	RSVD	R	Reserved. Bit 7 is read only and returns 0 when read. Writes have no effect.
6	CARDPWR	R	Card Power. Bit 6 indicates the current power status of the PC Card socket. This bit reflects how the power control register has been programmed. Bit 6 is encoded as: 0 = V_{CC} and V_{PP} to the socket turned off (default) 1 = V_{CC} and V_{PP} to the socket turned on
5	READY	R	Ready. Bit 5 indicates the current status of the READY signal at the PC Card interface. 0 = PC Card not ready for data transfer 1 = PC Card ready for data transfer
4	CARDWP	R	Card write protect. Bit 4 indicates the current status of WP at the PC Card interface. This signal reports to the PCI1251A whether or not the memory card is write protected. Furthermore, write protection for an entire PCI1251A 16-bit memory window is available by setting the appropriate bit in the memory window offset high-byte register. 0 = WP is 0. PC Card is R/W. 1 = WP is 1. PC Card is read only.
3	CDETECT2	R	Card detect 2. Bit 3 indicates the status of CD2 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 is 1. No PC Card is inserted. 1 = CD2 is 0. PC Card is at least partially inserted.
2	CDETECT1	R	Card detect 1. Bit 2 indicates the status of CD1 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 is 1. No PC Card is inserted. 1 = CD1 is 0. PC Card is at least partially inserted.
1–0	BVDSTAT	R	Battery voltage detect. When a 16-bit memory card is inserted, the field indicates the status of the battery voltage detect signals (BVD1, BVD2) at the PC Card interface, where bit 1 reflects the BVD1 status and bit 0 reflects BVD2. 00 = Battery dead 01 = Battery dead 10 = Battery low; warning 11 = Battery good When a 16-bit I/O card is inserted, this field indicates the status of $\overline{\text{SPKR}}$ (bit 1) and $\overline{\text{STSCHG}}$ (bit 0) at the PC Card interface. In this case, the two bits in this field directly reflect the current state of these card outputs.

ExCA power-control register (index 02h)

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

Register: **ExCA power control**
Type: Read only, read/write (see individual bit descriptions)
Offset: CardBus socket address + 802h; Card A ExCA offset 02h
Card B ExCA offset 42h
Default: 00h
Description: This register provides PC Card power control. Bit 7 of this register controls the 16-bit outputs on the socket interface and can be used for power management in 16-bit PC Card applications. Refer to Table 52 for a complete description of the register contents.

Table 52. ExCA Power-Control Register (Index 02h)

BIT	SIGNAL	TYPE	FUNCTION
7	COE	R/W	Card output enable. Bit 7 controls the state of all of the 16-bit outputs on the PCI1251A. This bit is encoded as: 0 = 16-bit PC Card outputs disabled (default) 1 = 16-bit PC Card outputs enabled
6–5	RSVD	R	Reserved. Bits 6–5 are read only and return 0s when read. Writes have no effect.
4–3	EXCAVCC	R/W	V_{CC}. Bits 4–3 are used to request changes to card V_{CC}. This field is encoded as: 00 = 0 V (default) 01 = 0 V reserved 10 = 5 V 11 = 3 V
2	RSVD	R	Reserved. Bit 2 is read only and returns 0 when read. Writes have no effect.
1–0	EXCAVPP	R/W	V_{PP}. Bits 1–0 are used to request changes to card V_{PP}. The PCI1251A ignores this field unless V_{CC} to the socket is enabled (i.e., 5 V or 3.3 V). This field is encoded as: 00 = 0 V (default) 01 = V_{CC} 10 = 12 V 11 = 0 V reserved

ExCA interrupt and general-control register (index 03h)

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

Register: **ExCA interrupt and general control**
Type: Read/write (see individual bit descriptions)
Offset: CardBus socket address + 803h; Card A ExCA offset 03h
Card B ExCA offset 43h
Default: 00h
Description: This register controls interrupt routing for I/O interrupts, as well as other critical 16-bit PC Card functions. Refer to Table 53 for a complete description of the register contents.

Table 53. ExCA Interrupt and General-Control Register (Index 03h)

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

ExCA card status-change register (index 04h)

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

Register: **ExCA card status change**
Type: Read only (see individual bit descriptions)
Offset: CardBus socket address + 804h; Card A ExCA offset 04h
Card B ExCA offset 44h
Default: 00h
Description: This register reflects the status of PC Card CSC interrupt sources. The ExCA card status-change interrupt configuration register, Table 55, enables these interrupt sources to generate an interrupt to the host. When the interrupt source is disabled, the corresponding bit in this register always reads 0. When an interrupt source is enabled, the corresponding bit in this register is set to indicate that the interrupt source is active. After generating the interrupt to the host, the interrupt service routine must read this register to determine the source of the interrupt. The interrupt service routine is responsible for resetting the bits in this register as well. Resetting a bit is accomplished by one of two methods: a read of this register or an explicit 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 global control register. Refer to Table 54 for a complete description of the register contents.

Table 54. ExCA Card Status-Change Register (Index 04h)

BIT	SIGNAL	TYPE	FUNCTION
7–4	RSVD	R	Reserved. Bits 7–4 are read only and return 0s when read. Writes have no effect.
3	CDCHANGE	R	Card detect change. Bit 3 indicates whether a change on CD1 or CD2 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 = Change detected on either CD1 or CD2
2	READYCHANGE	R	Ready change. When a 16-bit memory is installed in the socket, bit 2 includes whether the source of a PCI1251A interrupt was due to a change on READY at the PC Card interface, indicating that the PC Card is now ready to accept new data. 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 low-to-high transition on READY When a 16-bit I/O card is installed, bit 2 is always 0.
1	BATWARN	R	Battery warning change. When a 16-bit memory card is installed in the socket, bit 1 indicates whether the source of a PCI1251A 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 battery warning condition When a 16-bit I/O card is installed, bit 1 is always 0.
0	BATDEAD	R	Battery dead or status change. When a 16-bit memory card is installed in the socket, bit 0 indicates whether the source of a PCI1251A 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 = $\overline{\text{STSCHG}}$ deasserted (default) 1 = $\overline{\text{STSCHG}}$ asserted Ring indicate. When the PCI1251A is configured for ring indicate operation, bit 0 indicates the status of RI.

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

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

Register: **ExCA card status-change-interrupt configuration**
Type: Read/write (see individual bit descriptions)
Offset: CardBus socket address + 805h; Card A ExCA offset 05h
Card B ExCA offset 45h
Default: 00h
Description: This register controls interrupt routing for card status-change interrupts, as well as masking CSC interrupt sources. Refer to Table 55 for a complete description of the register contents.

Table 55. ExCA Card Status-Change-Interrupt Configuration Register (Index 05h)

BIT	SIGNAL	TYPE	FUNCTION
7–4	CSCSELECT	R/W	Interrupt select for card status change. Bits 7–4 select the interrupt routing for card status change interrupts. This field is encoded as: 0000 = No interrupt routing (default) 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. Bit 3 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. Bit 2 enables/disables a low-to-high transition on PC Card READY to generate a host interrupt. This interrupt source is considered a card status change. This bit is encoded as: 0 = Disables host interrupt generation (default) 1 = Enables host interrupt generation
1	BATWARNEN	R/W	Battery Warning Enable. Bit 1 enables/disables a battery warning condition to generate a CSC interrupt. This bit is encoded as: 0 = Disables host interrupt generation (default) 1 = Enables host interrupt generation
0	BATDEADEN	R/W	Battery dead enable. Bit 0 enables/disables 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

ExCA address window enable register (index 06h)

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

Register: **ExCA address window enable**
Type: Read only, read/write (see individual bit descriptions)
Offset: CardBus socket address + 806h; Card A ExCA offset 06h
Card B ExCA offset 46h
Default: 00h
Description: This register enables/disables the memory and I/O windows to the 16-bit PC Card. By default, all windows to the card are disabled. The PCI1251A does not acknowledge PCI memory or I/O cycles to the card if the corresponding enable bit in this register is 0, regardless of the programming of the memory or I/O window start/end/offset address registers. Refer to Table 56 for a complete description of the register contents.

Table 56. ExCA Address Window Enable Register (Index 06h)

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

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

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

Register: **ExCA I/O window control**
Type: Read/write (see individual bit descriptions)
Offset: CardBus socket address + 807h; Card A ExCA offset 07h
Card B ExCA offset 47h
Default: 00h
Description: This register contains parameters related to I/O window sizing and cycle timing. Refer to Table 57 for a complete description of the register contents.

Table 57. ExCA I/O Window Control Register (Index 07h)

BIT	SIGNAL	TYPE	FUNCTION
7	WAITSTATE1	R/W	I/O window 1 wait state. Bit 7 controls the I/O window 1 wait state for 16-bit I/O accesses. Bit 7 has no effect on 8-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 16-bit cycles have standard length (default). 1 = 16-bit cycles are extended by one equivalent ISA wait state.
6	ZEROWS1	R/W	I/O window 1 zero wait state. Bit 6 controls the I/O window 1 wait state for 8-bit I/O accesses. Bit 6 has no effect on 16-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 8-bit cycles have standard length (default). 1 = 8-bit cycles are reduced to equivalent of three ISA cycles.
5	IOIS16W1	R/W	I/O window 1 $\overline{\text{IOIS16}}$ source. Bit 5 controls the I/O window 1 automatic data sizing feature that uses $\overline{\text{IOIS16}}$ from the PC Card to determine the data width of the I/O data transfer. This bit is encoded as: 0 = Window data width determined by DATASIZE1, bit 4 (default). 1 = Window data width determined by $\overline{\text{IOIS16}}$.
4	DATASIZE1	R/W	I/O window 1 data size. Bit 4 controls the I/O window 1 data size. Bit 4 is ignored if the I/O window 1 $\overline{\text{IOIS16}}$ 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. Bit 3 controls the I/O window 0 wait state for 16-bit I/O accesses. Bit 3 has no effect on 8-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 16-bit cycles have standard length (default). 1 = 16-bit cycles are extended by one equivalent ISA wait state.
2	ZEROWS0	R/W	I/O window 0 zero wait state. Bit 2 controls the I/O window 0 wait state for 8-bit I/O accesses. Bit 2 has no effect on 16-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 8-bit cycles have standard length (default). 1 = 8-bit cycles are reduced to equivalent of three ISA cycles.
1	IOIS16W0	R/W	I/O window 0 $\overline{\text{IOIS16}}$ source. Bit 1 controls the I/O window 0 automatic data sizing feature that uses $\overline{\text{IOIS16}}$ from the PC Card to determine the data width of the I/O data transfer. This bit is encoded as: 0 = Window data width is determined by DATASIZE0, bit 0 (default). 1 = Window data width is determined by $\overline{\text{IOIS16}}$.
0	DATASIZE0	R/W	I/O window 0 data size. Bit 0 controls the I/O window 0 data size. Bit 0 is ignored if the I/O window 0 $\overline{\text{IOIS16}}$ source bit (bit 1) is set. This bit is encoded as: 0 = Window data width is 8 bits (default). 1 = Window data width is 16 bits.

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

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O window 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
Description: These registers contain the low byte of the 16-bit I/O window start address for I/O windows 0 and 1. The eight bits of these registers correspond to the lower eight bits of the start address.

ExCA I/O window 0 and 1 start-address high-byte register (index 09h, 0Dh)

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O window 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
Description: These registers contain the high byte of the 16-bit I/O window start address for I/O windows 0 and 1. The eight bits of these registers correspond to the upper eight bits of the start address.

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

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O window 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
Description: These registers contain the low 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 lower eight bits of the end address.

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

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O window 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
Description: These registers contain the high byte of the 16-bit I/O window end address for I/O windows 0 and 1. The eight bits of these registers correspond to the upper eight bits of the end address.

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

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

Register: **ExCA memory window 0 start-address low byte**
Offset: CardBus socket address + 810h; Card A ExCA offset 10h
Card B ExCA offset 50h
Register: **ExCA memory window 1 start-address low byte**
Offset: CardBus socket address + 818h; Card A ExCA offset 18h
Card B ExCA offset 58h
Register: **ExCA memory window 2 start-address low byte**
Offset: CardBus socket address + 820h; Card A ExCA offset 20h
Card B ExCA offset 60h

Register: **ExCA memory window 3 start-address low byte**
Offset: CardBus socket address + 828h; Card A ExCA offset 28h
Card B ExCA offset 68h
Register: **ExCA memory window 4 start-address low byte**
Offset: CardBus socket address + 830h; Card A ExCA offset 30h
Card B ExCA offset 70h

Type: Read/write
Default: 00h
Size: One byte
Description: These registers contain the low byte of the 16-bit memory window start address for memory windows 0, 1, 2, 3, and 4. The eight bits of these registers correspond to bits A19–A12 of the start address.

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

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

Register: **ExCA memory window 0 start-address high byte**
Offset: CardBus socket address + 811h; Card A ExCA offset 11h
Card B ExCA offset 51h
Register: **ExCA memory window 1 start-address high byte**
Offset: CardBus socket address + 819h; Card A ExCA offset 19h
Card B ExCA offset 59h
Register: **ExCA memory window 2 start-address high byte**
Offset: CardBus socket address + 821h; Card A ExCA offset 21h
Card B ExCA offset 61h
Register: **ExCA memory window 3 start-address high byte**
Offset: CardBus socket address + 829h; Card A ExCA offset 29h
Card B ExCA offset 69h
Register: **ExCA memory window 4 start-address high byte**
Offset: CardBus socket address + 831h; Card A ExCA offset 31h
Card B ExCA offset 71h
Type: Read/write
Default: 00h
Size: One byte
Description: These registers contain the high nibble of the 16-bit memory window start address for memory windows 0, 1, 2, 3, and 4. The lower four 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. Refer to Table 58 for a complete description of the register contents.

Table 58. ExCA Memory Window 0–4 Start-Address High-Byte Register (Index 11h, 19h, 21h, 29h, 31h)

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

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

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

Register: **ExCA memory window 0 end-address low byte**
Offset: CardBus socket address + 812h; Card A ExCA offset 12h
Card B ExCA offset 52h
Register: **ExCA memory window 1 end-address low byte**
Offset: CardBus socket address + 81Ah; Card A ExCA offset 1Ah
Card B ExCA offset 5Ah
Register: **ExCA memory window 2 end-address low byte**
Offset: CardBus socket address + 822h; Card A ExCA offset 22h
Card B ExCA offset 62h
Register: **ExCA memory window 3 end-address low byte**
Offset: CardBus socket address + 82Ah; Card A ExCA offset 2Ah
Card B ExCA offset 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
Description: These registers contain the low byte of the 16-bit memory window end address for memory windows 0, 1, 2, 3, and 4. The eight bits of these registers correspond to bits A19–A12 of the end address.

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

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

Register: **ExCA memory window 0 end-address high byte**
Offset: CardBus socket address + 813h; Card A ExCA offset 13h
Card B ExCA offset 53h
Register: **ExCA memory window 1 end-address high byte**
Offset: CardBus socket address + 81Bh; Card A ExCA offset 1Bh
Card B ExCA offset 5Bh
Register: **ExCA memory window 2 end-address high byte**
Offset: CardBus socket address + 823h; Card A ExCA offset 23h
Card B ExCA offset 63h
Register: **ExCA memory window 3 end-address high byte**
Offset: CardBus socket address + 82Bh; Card A ExCA offset 2Bh
Card B ExCA offset 6Bh
Register: **ExCA memory window 4 end-address high byte**
Offset: CardBus socket address + 833h; Card A ExCA offset 33h
Card B ExCA offset 73h
Type: Read only, read/write (see individual bit descriptions)
Default: 00h
Size: One byte
Description: These registers contain the high nibble of the 16-bit memory window end address for memory windows 0, 1, 2, 3, and 4. The lower four 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. Refer to Table 59 for a complete description of the register contents.

Table 59. ExCA Memory Window 0–4 End-Address High-Byte Register (Index 13h, 1Bh, 23h, 2Bh, 33h)

BIT	SIGNAL	TYPE	FUNCTION
7–6	MEMWS	R/W	Wait state. Bits 7–6 specify the number of equivalent ISA wait states to be added to 16-bit memory accesses. The number of wait states added is equal to the binary value of these two bits.
5–4	RSVD	R	Reserved. Bits 5–4 are read only and return 0s when read. Writes have no effect.
3–0	ENDHN	R/W	End-address high nibble. Bits 3–0 represent the upper address bits A23–A20 of the memory window end address.

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

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

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

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

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

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

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

Type: Read/write
Default: 00h
Size: One byte
Description: These registers contain the low byte of the 16-bit memory window offset address for memory windows 0, 1, 2, 3, and 4. The eight bits of these registers correspond to bits A19–A12 of the offset address.

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

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

Register: **ExCA memory window 0 offset-address high byte**
Offset: CardBus socket address + 815h; Card A ExCA offset 15h
Card B ExCA offset 55h
Register: **ExCA memory window 1 offset-address high byte**
Offset: CardBus socket address + 81Dh; Card A ExCA offset 1Dh
Card B ExCA offset 5Dh
Register: **ExCA memory window 2 offset-address high byte**
Offset: CardBus socket address + 825h; Card A ExCA offset 25h
Card B ExCA offset 65h
Register: **ExCA memory window 3 offset-address high byte**
Offset: CardBus socket address + 82Dh; Card A ExCA offset 2Dh
Card B ExCA offset 6Dh
Register: **ExCA memory window 4 offset-address high byte**
Offset: CardBus socket address + 835h; Card A ExCA offset 35h
Card B ExCA offset 75h
Type: Read only, read/write (see individual bit descriptions)
Default: 00h
Size: One byte
Description: These registers contain the high six bits of the 16-bit memory window offset address for memory windows 0, 1, 2, 3, and 4. The lower six 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. Refer to Table 60 for a complete description of the register contents.

Table 60. ExCA Memory Window 0–4 Offset-Address High-Byte Register (Index 15h, 1Dh, 25h, 2Dh, 35h)

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

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

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

Register: **ExCA I/O window 0 offset-address low byte**
Offset: CardBus socket address + 836h; Card A ExCA offset 36h
Card B ExCA offset 76h
Register: **ExCA I/O window 1 offset-address low byte**
Offset: CardBus socket address + 838h; Card A ExCA offset 38h
Card B ExCA offset 78h
Type: Read/write
Default: 00h
Size: One byte
Description: These registers contain the low byte of the 16-bit I/O window offset address for I/O windows 0 and 1. The eight bits of these registers correspond to the lower eight bits of the offset address, and bit 0 is always 0.

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

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O window 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
Description: These registers contain the high byte of the 16-bit I/O window offset address for I/O windows 0 and 1. The eight bits of these registers correspond to the upper eight bits of the offset address.

ExCA card detect and general-control register (index 16h)

Bit	7	6	5	4	3	2	1	0
Name	ExCA I/O 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 (see individual bit descriptions)
Offset: CardBus socket address + 816h; Card A ExCA offset 16h
Card B ExCA offset 56h
Default: XX00 0000b
Description: This register controls how the ExCA registers for the socket respond to card removal, as well as reports the status of $\overline{VS1}$ and $\overline{VS2}$ at the PC Card interface. Refer to Table 61 for a complete description of the register contents.

Table 61. ExCA Card Detect and General-Control Register (Index 16h)

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

ExCA global-control register (index 1Eh)

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

Register: **ExCA global control**
Type: Read only, read/write (see individual bit descriptions)
Offset: CardBus socket address + 81Eh; Card A ExCA offset 1Eh
Card B ExCA offset 5Eh
Default: 00h
Description: This register controls both PC Card sockets and is not duplicated for each socket. The host interrupt mode bits in this register are retained for Intel 82365SL-DF compatibility. Refer to Table 62 for a complete description of the register contents.

Table 62. ExCA Global-Control Register (Index 1Eh)

BIT	SIGNAL	TYPE	FUNCTION
7–5	RSVD	R	Reserved. Bits 7–5 are is read only and returns 0s when read. Writes have no effect.
4	INTMODEB	R/W	Level/edge interrupt mode select – card B. Bit 4 selects the signaling mode for the PCI1251A 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. Bit 3 selects the signaling mode for the PCI1251A 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. Bit 2 selects the interrupt flag clear mechanism for the flags in the ExCA card status change register. This bit is encoded as: 0 = Interrupt flags are cleared by read of CSC register (default). 1 = Interrupt flags are cleared by explicit write back of 1.
1	CSCMODE	R/W	Card status change level/edge mode select. Bit 1 selects the signaling mode for the PCI1251A host interrupt for card status changes. This bit is encoded as: 0 = Host interrupt is edge mode (default). 1 = Host interrupt is level mode.
0	PWRDWN	R/W	Power-down mode select. When bit 0 is set to 1, the PCI1251A is in power-down mode. In power-down mode, the PCI1251A card outputs are 3-stated until an active cycle is executed on the card interface. Following an active cycle, the outputs are again 3-stated. The PCI1251A still receives DMA requests, functional interrupts, and/or card status change interrupts; however, an actual card access is required to wake up the interface. This bit is encoded as: 0 = Power-down mode is disabled (default). 1 = Power-down mode is enabled.

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

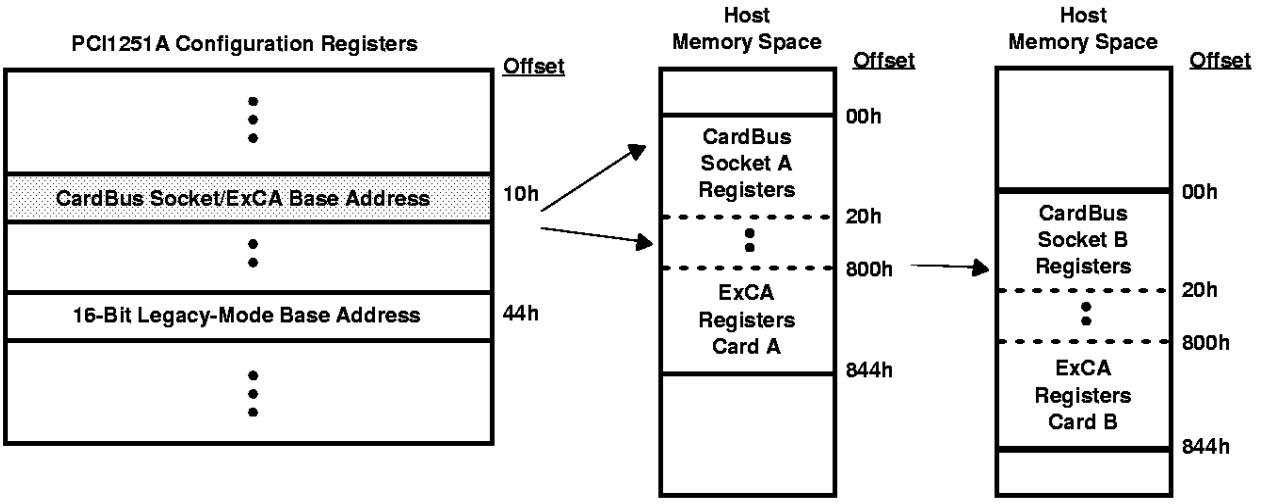
Bit	7	6	5	4	3	2	1	0
Name	ExCA memory window 0–4 page							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

Register: **ExCA memory window 0–4 page**
Type: Read/write
Offset: CardBus socket address + 840h 841h, 842h, 843h, 844h
Default: 00h
Description: The upper eight 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 can locate 16-bit memory windows in any one of 256 16M-byte regions in the 4G-byte PCI address space. These registers are only accessible when the ExCA registers are memory mapped, i.e., these registers can not be accessed using the index/data I/O scheme.

CardBus socket registers (functions 0 and 1)

The PCMCIA CardBus specification requires a CardBus socket controller to provide five 32-bit registers that report and control socket-specific functions. The PCI1251A 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 18). Table 63 gives the location of the socket registers in relation to the CardBus socket/ExCA base address.

The PCI1251A implements an additional register at offset 20h that provides power management control for the socket.



NOTE: The CardBus socket/ExCA base address mode register is separate for functions 0 and 1.

Figure 18. Accessing CardBus Socket Registers Through PCI Memory

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

CardBus socket registers (functions 0 and 1) (continued)

Table 63. 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

socket event register

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Socket event															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Socket event															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R/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 event**
Type: Read only, read/write (see individual bit descriptions)
Offset: CardBus socket address + 00h
Default: 0000 0000h
Description: The socket event register indicates a change in socket status has occurred. These bits do not indicate what the change is, only that one has occurred. Software must read the socket present state register for current status. Each bit in this register can be cleared by writing a 1 to that bit. The bits in this register can be set to a 1 by software by writing a 1 to the corresponding bit in the socket force event register. All bits in this register are cleared by PCI reset. They can be immediately set again, if, when coming out of PC Card reset, the bridge finds the status unchanged (i.e., CSTSCHG reasserted or card detect is still true). Software must clear this register before enabling interrupts. If it is not cleared, when interrupts are enabled an interrupt is generated (but not masked) based on any bit set. Refer to Table 64 for a complete description of the register contents.

Table 64. Socket Event Register

BIT	SIGNAL	TYPE	FUNCTION
31–4	RSVD	R	Reserved. Bits 31–4 are read only and return 0s when read.
3	PWREVENT	R/W	Power cycle. Bit 3 is set when the PCI1251A detects that the PWRCYCLE bit in the socket present-state register has changed. This bit is cleared by writing a 1.
2	CD2EVENT	R/W	$\overline{\text{CCD2}}$. Bit 2 is set when the PCI1251A detects that the CDETECT2 field in the socket present-state register has changed. This bit is cleared by writing a 1.
1	CD1EVENT	R/W	$\overline{\text{CCD1}}$. Bit 3 is set when the PCI1251A detects that the CDETECT1 field in the socket present-state register has changed. This bit is cleared by writing a 1.
0	CSTSEVENT	R/W	CSTSCHG. Bit 0 is set when the CARDSTS field in the socket present-state register has changed state. For CardBus cards, bit 0 is set on the rising edge of CSTSCHG. For 16-bit PC Cards, bit 0 is set on both transitions of CSTSCHG. This bit is reset by writing a 1.

socket mask register

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

Register: **Socket mask**
Type: Read only, read/write (see individual bit descriptions)
Offset: CardBus socket address + 04h
Default: 0000 0000h
Description: The socket mask register allows software to control the CardBus card events that generate a status change interrupt. The state of these mask bits does not prevent the corresponding bits from reacting in the socket event register. Refer to Table 65 for a complete description of the register contents.

Table 65. Socket Mask Register

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

socket present-state register

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

Register: **Socket present state**
Type: Read only, read/write (see individual bit descriptions)
Offset: CardBus socket address + 08h
Default: 3000 000Xh
Description: The socket present-state register reports information about the socket interface. Writes to the socket force event register are reflected here, as well as general socket interface status. Information about PC Card V_{CC} support and card type is only updated at each insertion. Also note that the PCI1251A uses $\overline{CCD1}$ and $\overline{CCD2}$ during card identification, and changes on these signals during this operation are not reflected in this register. Refer to Table 66 for a complete description of the register contents.

Table 66. Socket Present-State Register

BIT	SIGNAL	TYPE	FUNCTION
31	YVSOCKET	R	YV socket. Bit 31 indicates whether or not the socket can supply V_{CC} = Y.Y V to PC Cards. The PCI1251A does not support Y.Y-V V_{CC}; therefore, this bit is always reset unless overridden by the socket force event register. This bit is hardwired to 0.
30	XVSOCKET	R	XV socket. Bit 30 indicates whether or not the socket can supply V_{CC} = X.X V to PC Cards. The PCI1251A does not support X.X-V V_{CC}; therefore, this bit is always reset unless overridden by the socket force event register. This bit is hardwired to 0.
29	3VSOCKET	R	3-V socket. Bit 29 indicates whether or not the socket can supply V_{CC} = 3.3 V to PC Cards. The PCI1251A does support 3.3-V V_{CC}; therefore, this bit is always set unless overridden by bit 6 of the device control register.
28	5VSOCKET	R	5-V socket. Bit 28 indicates whether or not the socket can supply V_{CC} = 5 V to PC Cards. The PCI1251A does support 5-V V_{CC}; therefore, this bit is always 1.
27–14	RSVD	R	Reserved. Bits 27–14 are read only and return 0s when read.
13	YVCARD	R	YV card. Bit 13 indicates whether or not the PC Card inserted in the socket supports V_{CC} = Y.Y V. This bit can be set by writing to the corresponding bit in the socket force event register.
12	XVCARD	R	XV card. Bit 12 indicates whether or not the PC Card inserted in the socket supports V_{CC} = X.X V. This bit can be set by writing to the corresponding bit in the socket force event register.
11	3VCARD	R	3-V card. Bit 11 indicates whether or not the PC Card inserted in the socket supports V_{CC} = 3.3 V. This bit can be set by writing to the corresponding bit in the socket force event register.
10	5VCARD	R	5-V card. Bit 10 indicates whether or not the PC Card inserted in the socket supports V_{CC} = 5 V. This bit can be set by writing to the corresponding bit in the socket force event register.
9	BADVCCREQ	R	Bad V_{CC} request. Bit 9 indicates that the host software has requested that the socket be powered at an invalid voltage. 0 = Normal operation (default) 1 = Invalid V_{CC} request by host software
8	DATALOST	R	Data lost. Bit 8 indicates that a PC Card removal event may have caused lost data because the cycle did not terminate properly or because write data still resides in the PCI1251A. 0 = Normal operation (default) 1 = Potential data loss due to card removal
7	NOTACARD	R	Not a card. Bit 7 indicates that an unrecognizable PC Card has been inserted in the socket. This bit is not updated until a valid PC Card is inserted into the socket. 0 = Normal operation (default) 1 = Unrecognizable PC Card detected

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

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

socket force event register

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

Register: **Socket force event**
Type: Read only, write only (see individual bit descriptions)
Offset: CardBus socket address + 0Ch
Default: 0000 00XXh
Description: The socket force event register is used to force changes to the socket event register and the socket present state register. The CVSTEST bit in this register must be written when forcing changes that require card interrogation. Refer to Table 67 for a complete description of the register contents.

Table 67. Socket Force Event Register

BIT	SIGNAL	TYPE	FUNCTION
31–15	RSVD	R	Reserved. Bits 31–15 are read only and return 0s when read.
14	CVSTEST	W	Card VS test. When bit 14 is set, the PCI1251A reinterrogates the PC Card, updates the socket present state register, and reenables the socket power control.
13	FYVCARD	W	Force YV card. Writes to bit 13 cause the YVCARD bit in the socket present state register to be written. When set, this bit disables the socket power control.
12	FXVCARD	W	Force XV card. Writes to bit 12 cause the XVCARD bit in the socket present state register to be written. When set, this bit disables the socket power control.
11	F3VCARD	W	Force 3-V card. Writes to bit 11 cause the 3VCARD bit in the socket present state register to be written. When set, this bit disables the socket power control.
10	F5VCARD	W	Force 5-V card. Writes to bit 10 cause the 5VCARD bit in the socket present state register to be written. When set, this bit disables the socket power control.
9	FBADVCCREQ	W	Force bad V_{CC} request. Changes to the BADVCCREQ bit in the socket present state register can be made by writing to bit 9.
8	FDATALOST	W	Force data lost. Writes to bit 8 cause the DATALOST bit in the socket present state register to be written.
7	FNOTACARD	W	Force not a card. Writes to bit 7 cause the NOTACARD bit in the socket present state register to be written.
6	RSVD	R	Reserved. Bit 6 is read only and returns 0 when read.
5	FCBCARD	W	Force CardBus card. Writes to bit 5 cause the CBCARD bit in the socket present state register to be written.
4	F16BITCARD	W	Force 16-bit card. Writes to bit 4 cause the 16BITCARD bit in the socket present state register to be written.
3	FPWRCYCLE	W	Force power cycle. Writes to bit 3 cause the PWREVENT bit in the socket event register to be written, and the PWRCYCLE bit in the socket present state register is unaffected.
2	FCDETECT2	W	Force $\overline{\text{CCD2}}$. Writes to bit 2 cause the CD2EVENT bit in the socket event register to be written, and the CDETECT2 bit in the socket present state register is unaffected.
1	FCDETECT1	W	Force $\overline{\text{CCD1}}$. Writes to bit 1 cause the CD1EVENT bit in the socket event register to be written, and the CDETECT1 bit in the socket present state register is unaffected.
0	FCARDSTS	W	Force CSTSCHG. Writes to bit 0 cause the CSTSEVENT bit in the socket event register to be written, and the CARDSTS bit in the socket present state register is unaffected.

socket control register

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

Register: **Socket control**
Type: Read only, read/write (see individual bit descriptions)
Offset: CardBus socket address + 10h
Default: 0000 0000h
Description: The socket control register provides control of the voltages applied to the socket and instructions for CB $\overline{\text{CLKRUN}}$ protocol. The PCI1251A ensures that the socket is powered up only at acceptable voltages when a CardBus card is inserted. Refer to Table 68 for a complete description of the register contents.

Table 68. Socket Control Register

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

socket power management register

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

Register: **Socket power management**
Type: Read only, read/write (see individual bit descriptions)
Offset: CardBus socket address + 20h
Default: 0000 0000h
Description: This register provides power management control over the socket through a mechanism for slowing or stopping the clock on the card interface when the card is idle.

Table 69. Socket Power Management Register

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

distributed DMA (DDMA) registers

The DMA base address, programmable in PCI configuration space at offset 98h, points to a 16-byte region in PCI I/O space where the DDMA registers reside. The names and locations of these registers are summarized in Table 70. These PCI1251A register definitions are identical to those registers of the same name in the Intel 8237 DMA controller; however, some register bits defined in the 8237 do not apply to distributed DMA in a PCI environment. The similarity between the register models retains some level of compatibility with legacy DMA and simplifies the translation required by the master DMA device when it forwards legacy DMA writes to DMA channels.

While the DMA register definitions are identical to those in the 8237 of the same name, some register bits defined in the 8237 do not apply to distributed DMA in a PCI environment. In such cases, the PCI1251A implements these obsolete register bits as read-only nonfunctional bits. The reserved registers shown in Table 70 are implemented as read only and return 0s when read. Writes to reserved registers have no effect.

Table 70. DDMA Registers

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

DMA current address/base address register

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

Register: **DMA current address/base address**
Type: Read/write
Offset: DMA base address + 00h
Default: 0000h
Size: Two bytes
Description: This read/write 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 current address register contents are presented on AD15–0 of the PCI bus during the address phase. Bits 7–0 of the page register are presented on AD23–AD16 of the PCI bus during the address phase.

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

DMA page register

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

Register: **DMA page**
Type: Read/write
Offset: DMA base address + 02h
Default: 00h
Size: One byte
Description: This read/write 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 *DMA current address/base address register*.

DMA current count/base count register

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

Register: **DMA current count/base count**
Type: Read/write
Offset: DMA base address + 04h
Default: 0000h
Size: Two bytes
Description: This read/write register is used to set the total transfer count, in bytes, of a direct memory transfer. Reads to this register indicate the current count of a direct memory transfer. In the 8-bit transfer mode, the count is decremented by 1 after each transfer and decremented by 2 after each transfer in the 16-bit transfer mode.

DMA command register

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

Register: **DMA command**
Type: Read only, read/write (see individual bit descriptions)
Offset: DMA base address + 08h
Default: 00h
Size: One byte
Description: This register is used to enable and disable the DMA controller. Bit 2, the only read/write bit, defaults to 0, enabling the DMA controller. All other bits are reserved. Refer to Table 71 for a complete description of the register contents.

Table 71. DMA Command Register

BIT	TYPE	FUNCTION
7–3	R	Reserved. Bits 7–3 are read only and return 0s when read.
2	R/W	DMA controller enable. Bit 2 enables and disables the distributed DMA slave controller in the PCI1251A and defaults to the enabled state. 0 = DMA controller enabled (default) 1 = DMA controller disabled
1–0	R	Reserved. Bits 1–0 are read only and return 0s when read.

DMA status register

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

Register: **DMA status**
Type: Read only (see individual bit descriptions)
Offset: DMA base address + 08h
Default: 00h
Size: One byte
Description: This read-only register indicates the terminal count and DMA request ($\overline{\text{DREQ}}$) status. Refer to Table 72 for a complete description of the register contents.

Table 72. DMA Status Register

BIT	SIGNAL	TYPE	FUNCTION
7–4	DREQSTAT	R	Channel request. In the 8237, bits 7–4 indicate the status of $\overline{\text{DREQ}}$ of each DMA channel. In the PCI1251A, 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 $\overline{\text{DREQ}}$ and are reset when $\overline{\text{DREQ}}$ is deasserted. The status of the mask bit in the multichannel mask register has no effect on these bits.
3–0	TC	R	Channel terminal count. The 8327 uses bits 3–0 to indicate the TC status of each of its four DMA channels. In the PCI1251A, these bits report information about a single DMA channel; therefore, all four of these register bits indicate the TC status of the single socket being serviced by this register. All four bits are set when the TC is reached by the DMA channel. These bits are reset when read or the DMA channel is reset.

DMA request register

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

Register: **DMA request**
Type: Write only
Offset: DMA base address + 09h
Default: 00h
Size: One byte
Description: This write-only register is used to request a DDMA transfer through software. Any write to this register enables software requests, and this register is to be used in block mode only.

DMA mode register

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

Register: **DMA mode**
Type: Read only, read/write (see individual bit descriptions)
Offset: DMA base address + 0Bh
Default: 00h
Size: One byte
Description: This write-only register is used to set the DMA transfer mode. Refer to Table 73 for a complete description of the register contents.

Table 73. DMA Mode Register

BIT	SIGNAL	TYPE	FUNCTION
7–6	DMAMODE	R/W	Mode select. The PCI1251A uses bits 7–6 to determine the transfer mode. 00 = Demand mode select (default) 01 = Single mode select 10 = Block mode select 11 = Reserved
5	INCDEC	R/W	Address increment/decrement. The PCI1251A uses bit 5 to select the memory address in the current address/base address register to increment or decrement after each data transfer. This is in accordance with the 8237 use of this register bit, and is encoded as follows: 0 = Addresses increment (default). 1 = Addresses decrement.
4	AUTOINIT	R/W	Auto initialization 0 = Auto initialization disabled (default) 1 = Auto initialization enabled
3–2	XFERTYPE	R/W	Transfer type. Bits 3–2 select the type of direct memory transfer to be performed. A memory write transfer moves data from the PCI1251A PC Card interface to memory, and a memory read transfer moves data from memory to the PCI1251A PC Card interface. The field is encoded as: 00 = No transfer selected (default) 01 = Write transfer 10 = Read transfer 11 = Reserved
1–0	RSVD	R	Reserved. Bits 1–0 are read only and return 0s when read.

DMA master clear register

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

Register: **DMA master clear**
Type: Write only
Offset: DMA base address + 0Dh
Default: 00h
Size: One byte
Description: This write-only register is used to reset the DDMA controller and resets all DDMA registers.

DMA multichannel/mask register

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

Register: **DMA multichannel/mask**
Type: Read only (see individual bit descriptions)
Offset: DMA base address + 0Fh
Default: 00h
Size: One byte
Description: The PCI1251A uses only the least-significant bit of this register to mask the PC Card DMA channel. The PCI1251A sets the mask bit when the PC Card is removed. Host software is responsible for either resetting the socket's DMA controller or reenabling the mask bit. Refer to Table 74 for a complete description of the register contents.

Table 74. DMA Multichannel/Mask Register

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

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

Supply voltage range, V_{CC}	–0.5 V to 4.6 V
Clamping voltage range, V_{CCP}, V_{CCA}, V_{CCB}, V_{CCZ}, V_{CCI}	–0.5 V to 6 V
Input voltage range, V_I: PCI	–0.5 V to V_{CCP} + 0.5 V
Card A	–0.5 to V_{CCA} + 0.5 V
Card B	–0.5 to V_{CCB} + 0.5 V
ZV	–0.5 to V_{CCZ} + 0.5 V
MISC	–0.5 to V_{CCI} + 0.5 V
Fail safe	–0.5 V to V_{CC} + 0.5 V
Output voltage range, V_O: PCI	–0.5 V to V_{CCP} + 0.5 V
Card A	–0.5 to V_{CCA} + 0.5 V
Card B	–0.5 to V_{CCB} + 0.5 V
ZV	–0.5 to V_{CCZ} + 0.5 V
MISC	–0.5 to V_{CCI} + 0.5 V
Fail safe	–0.5 V to V_{CC} + 0.5 V
Input clamp current, I_{IK} ($V_I < 0$ or $V_I > V_{CC}$) (see Note 1)	±20 mA
Output clamp current, I_{OK} ($V_O < 0$ or $V_O > V_{CC}$) (see Note 2)	±20 mA
Storage temperature range, T_{stg}	–65°C to 150°C
Virtual junction temperature, T_J	150°C

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

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

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

recommended operating conditions (see Note 3)

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

† Applies to external inputs and bidirectional buffers without hysteresis

‡ Miscellaneous GFN pins are V13, W13, Y13, U12, V12, W12, U11, V11, W11, Y11, Y10, W10, Y09, W09, V09, U09, Y08, all IRQMUXx pins, LEDAx pins, SUSPEND, SPKROUT, RI_OUT, INTA, INTB, and power switch control pins.

§ Fail-safe GFN pins are A11, B14, C09, G03, H20, U03, W06, and Y03 (card detect and voltage sense pins).

¶ Applies to external output buffers

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

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

electrical characteristics over recommended operating conditions (unless otherwise noted)

PARAMETER		PINS	OPERATION	TEST CONDITIONS	MIN	MAX	UNIT
V_{OH}	High-level output voltage (see Note 4)	PCI	3.3 V	I_{OH} = –0.5 mA	0.9 V_{CC}		V
			5 V	I_{OH} = –2 mA	2.4		
		PC Card	3.3 V	I_{OH} = –0.15 mA	0.9 V_{CC}		
			5 V	I_{OH} = –0.15 mA	2.4		
		ZV		I_{OH} = –4 mA	V_{CC}–0.6		
		MISC		I_{OH} = –4 mA	V_{CC}–0.6		
V_{OL}	Low-level output voltage	PCI	3.3 V	I_{OL} = 1.5 mA		0.1 V_{CC}	V
			5 V	I_{OL} = 6 mA		0.55	
		PC Card	3.3 V	I_{OL} = 0.7 mA		0.1 V_{CC}	
			5 V	I_{OL} = 0.7 mA		0.55	
		ZV		I_{OL} = 4 mA		0.5	
		MISC		I_{OL} = 4 mA		0.5	
		$\overline{SERR}$		I_{OL} = 12 mA		0.5	
I_{OZL}	3-state output, high-impedance state current (see Note 4)	Output pins	3.6 V	V_I = V_{CC}		–1	µA
			5.25 V	V_I = V_{CC}		–1	
I_{OZH}	3-state output, high-impedance state current	Output pins	3.6 V	V_I = V_{CC}†		10	µA
			5.25 V	V_I = V_{CC}†		25	
I_{IL}	Low-level input current	Input pins		V_I = GND		–1	µA
		I/O pins		V_I = GND		–10	
		Latch		V_I = GND		–2	
I_{IH}	High-level input current (see Note 5)	Input pins	3.6 V	V_I = V_{CC}‡		10	µA
			5.25 V	V_I = V_{CC}‡		20	
		I/O pins	3.6 V	V_I = V_{CC}‡		10	
			5.25 V	V_I = V_{CC}‡		25	
		Fail-safe pins	3.6 V	V_I = V_{CC}		10	

† For PCI pins, V_I = V_{CCP}. For PC Card pins, V_I = $V_{CC(A/B)}$. For ZV pins, V_I = V_{CCZ}. For miscellaneous pins, V_I = V_{CCI}.
‡ For I/O pins, input leakage (I_{IL} and I_{IH}) includes I_{OZ} leakage of the disabled output.
NOTES: 4. V_{OH} and I_{OL} are not tested on SERR (GFN pin U19)and $\overline{RI\_OUT}$ (GFN pin Y13) because they are open-drain outputs.
5. I_{IH} is not tested on LATCH (GFN pin W12) because it is pulled up with an internal resistor.

PCI clock/reset timing requirements over recommended ranges of supply voltage and operating free-air temperature (see Figures 20 and 21)

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

PCI timing requirements over recommended ranges of supply voltage and operating free-air temperature (see Note 4 and Figures 19 and 22)

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

NOTES: 6. PCI shared signals are AD31–0, C/$\overline{\text{BE3–0}}$, $\overline{\text{FRAME}}$, $\overline{\text{TRDY}}$, $\overline{\text{IRDY}}$, $\overline{\text{STOP}}$, IDSEL, $\overline{\text{DEVSEL}}$, and PAR.

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

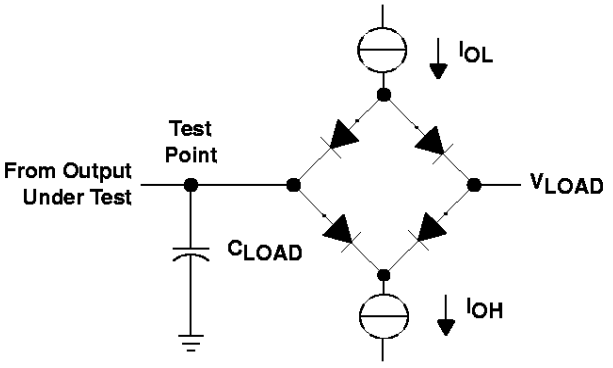
PARAMETER MEASUREMENT INFORMATION

LOAD CIRCUIT PARAMETERS

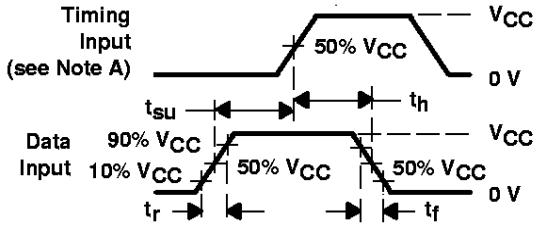
TIMING PARAMETER		C_{LOAD}† (pF)	I_{OL} (mA)	I_{OH} (mA)	V_{LOAD}‡ (V)
t_{en}	t_{PZH}	50	8	–8	0
	t_{PZL}				3
t_{dis}	t_{PHZ}	50	8	–8	1.5
	t_{PLZ}				
t_{pd}		50	8	–8	‡

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

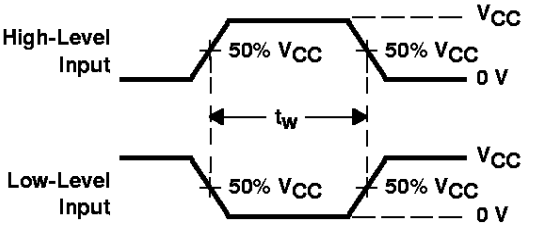
‡ $\frac{V_{LOAD} - V_{OL}}{I_{OL}}$ = 50 Ω, where V_{OL} = 0.6 V, I_{OL} = 8 mA



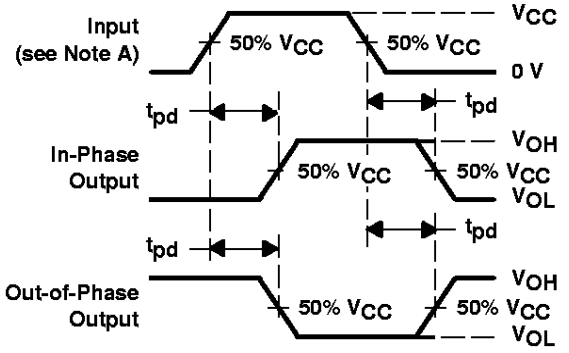
LOAD CIRCUIT



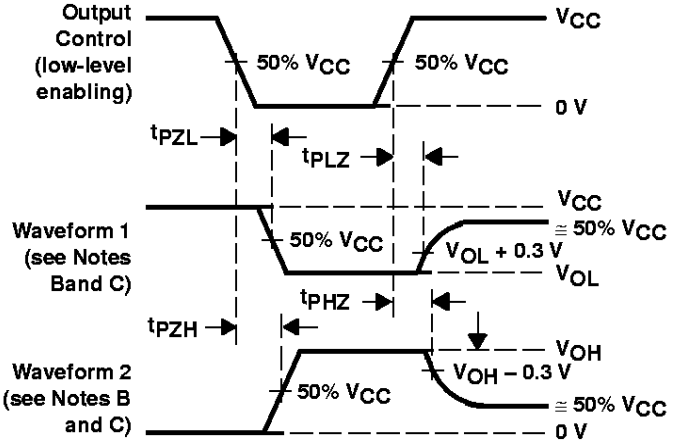
VOLTAGE WAVEFORMS
SETUP AND HOLD TIMES
INPUT RISE AND FALL TIMES



VOLTAGE WAVEFORMS
PULSE DURATION



VOLTAGE WAVEFORMS
PROPAGATION DELAY TIMES



VOLTAGE WAVEFORMS
ENABLE AND DISABLE TIMES, 3-STATE OUTPUTS

NOTES: A. Phase relationships between waveforms were chosen arbitrarily. All input pulses are supplied by pulse generators having the following characteristics: PRR = 1 MHz, Z_O = 50 Ω, t_r = 6 ns.

B. Waveform 1 is for an output with internal conditions such that the output is low except when disabled by the output control. Waveform 2 is for an output with internal conditions such that the output is high except when disabled by the output control.

C. For t_{PLZ} and t_{PHZ}, V_{OL} and V_{OH} are measured values.

Figure 19. Load Circuit and Voltage Waveforms

PCI BUS PARAMETER MEASUREMENT INFORMATION

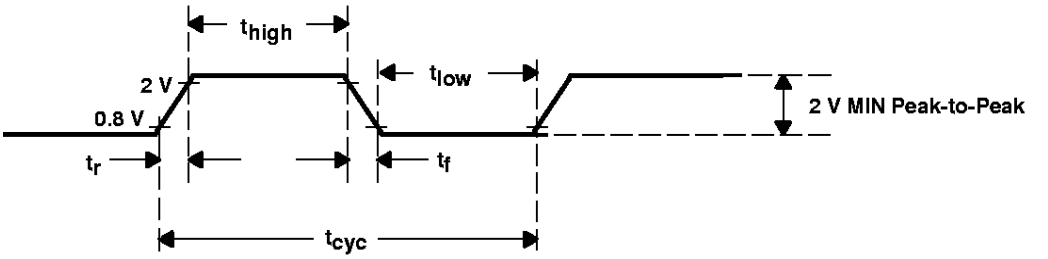


Figure 20. PCLK Timing Waveform

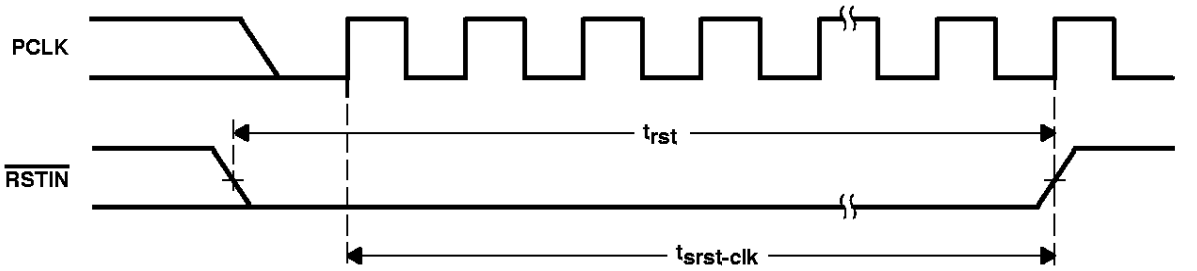


Figure 21. $\overline{\text{RSTIN}}$ Timing Waveforms

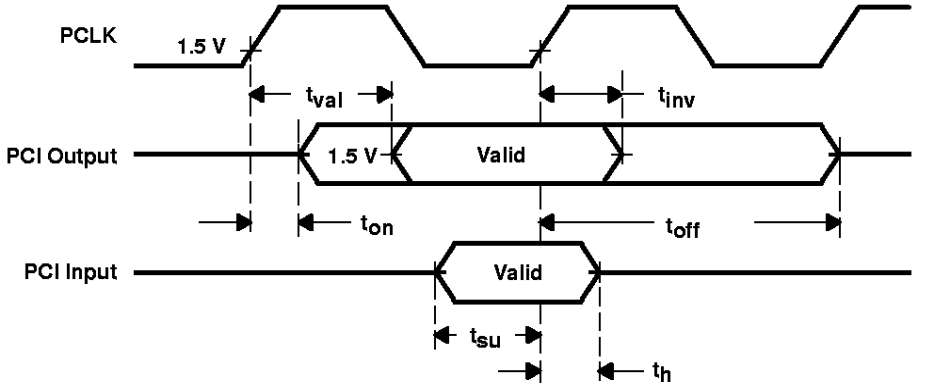


Figure 22. Shared Signals Timing Waveforms

PC Card cycle timing

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

The PC Card address setup and hold times are a function of the wait-state bits. Table 75 shows address setup time in PCLK cycles and nanoseconds for I/O and memory cycles. Table 76 and Table 77 show command active time in PCLK cycles and nanoseconds for I/O and memory cycles. Table 78 shows address hold time in PCLK cycles and nanoseconds for I/O and memory cycles.

Table 75. 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 76. 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 77. 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 78. PC Card Address Hold Time, $t_{h(A)}$, 8-Bit and 16-Bit PCI Cycles

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

timing requirements over recommended ranges of supply voltage and operating free-air temperature, memory cycles (for 100-ns common memory) (see Note 5 and Figure 23)

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

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

timing requirements over recommended ranges of supply voltage and operating free-air temperature, I/O cycles (see Figure 24)

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

switching characteristics over recommended ranges of supply voltage and operating free-air temperature, miscellaneous (see Figure 25)

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

PC Card PARAMETER MEASUREMENT INFORMATION

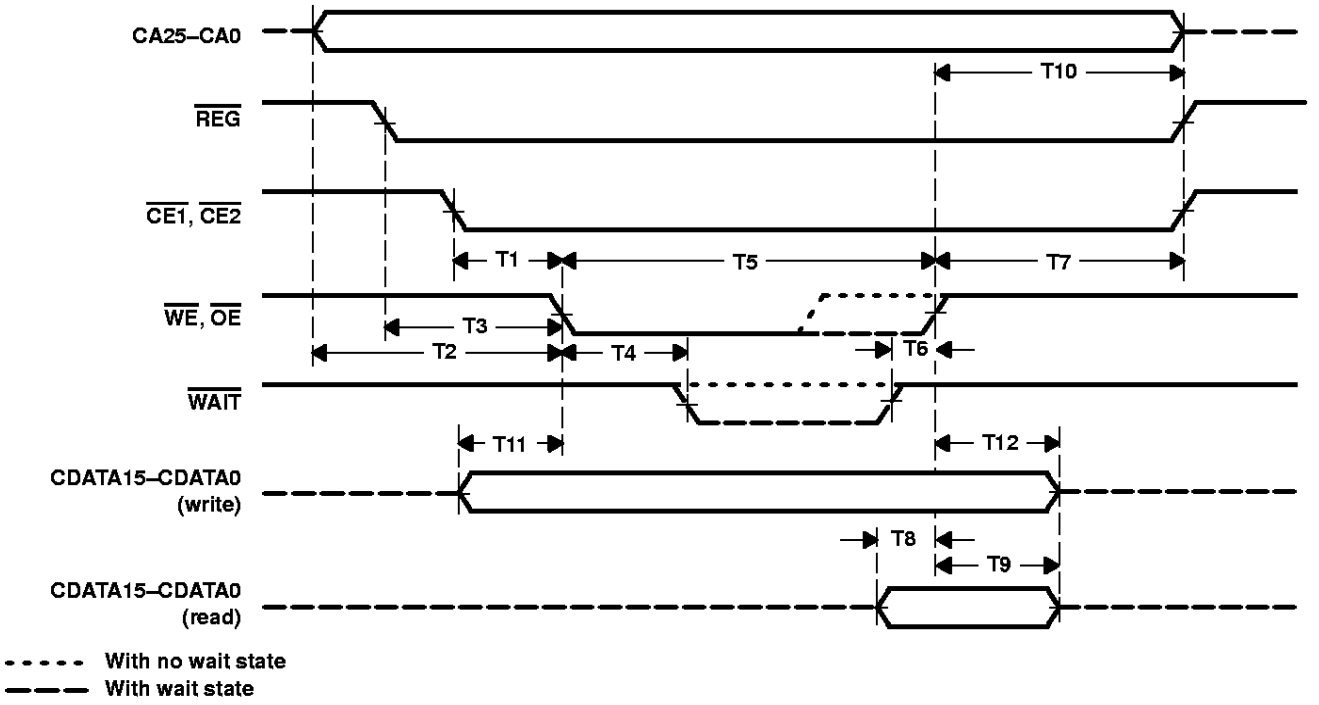


Figure 23. PC Card Memory Cycle

PC Card PARAMETER MEASUREMENT INFORMATION

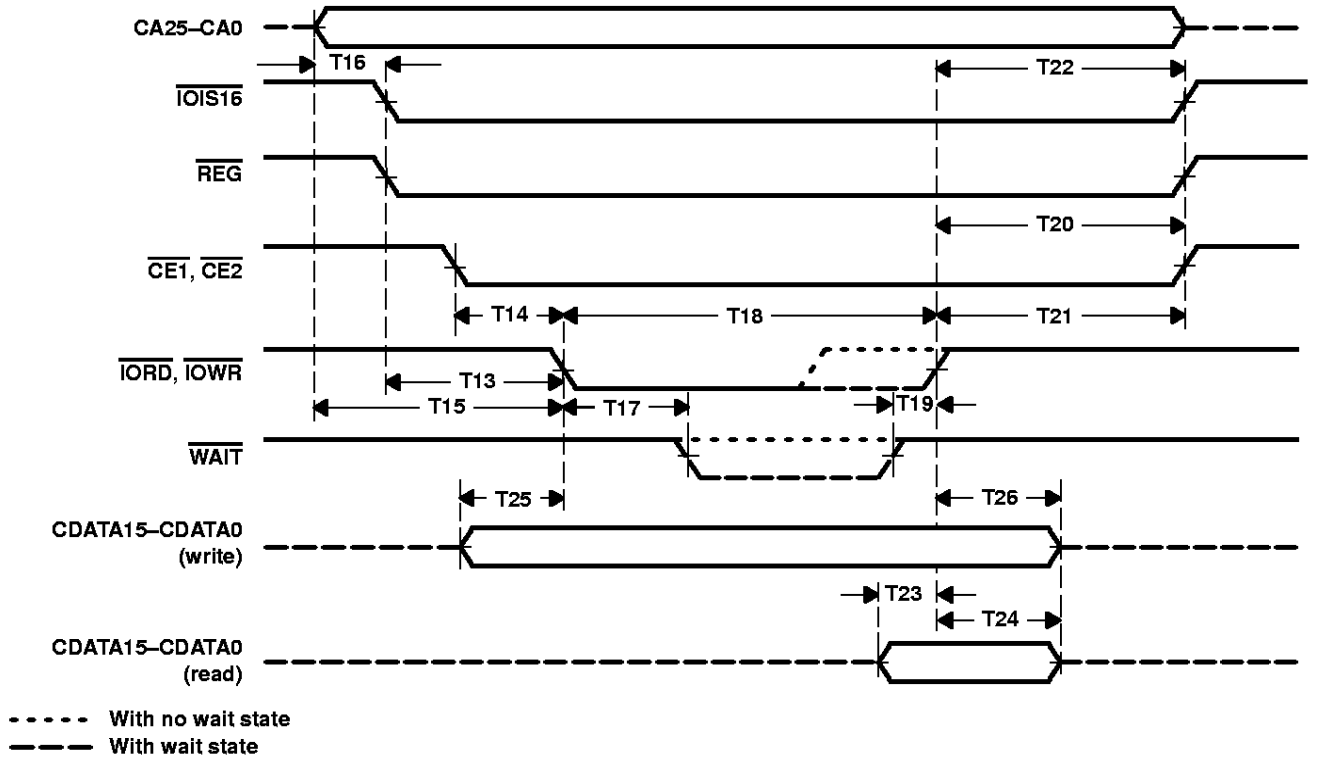


Figure 24. PC Card I/O Cycle

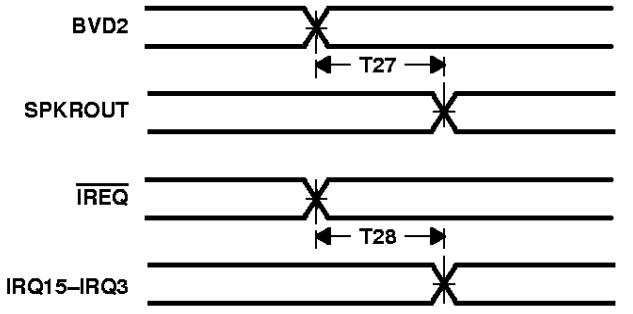
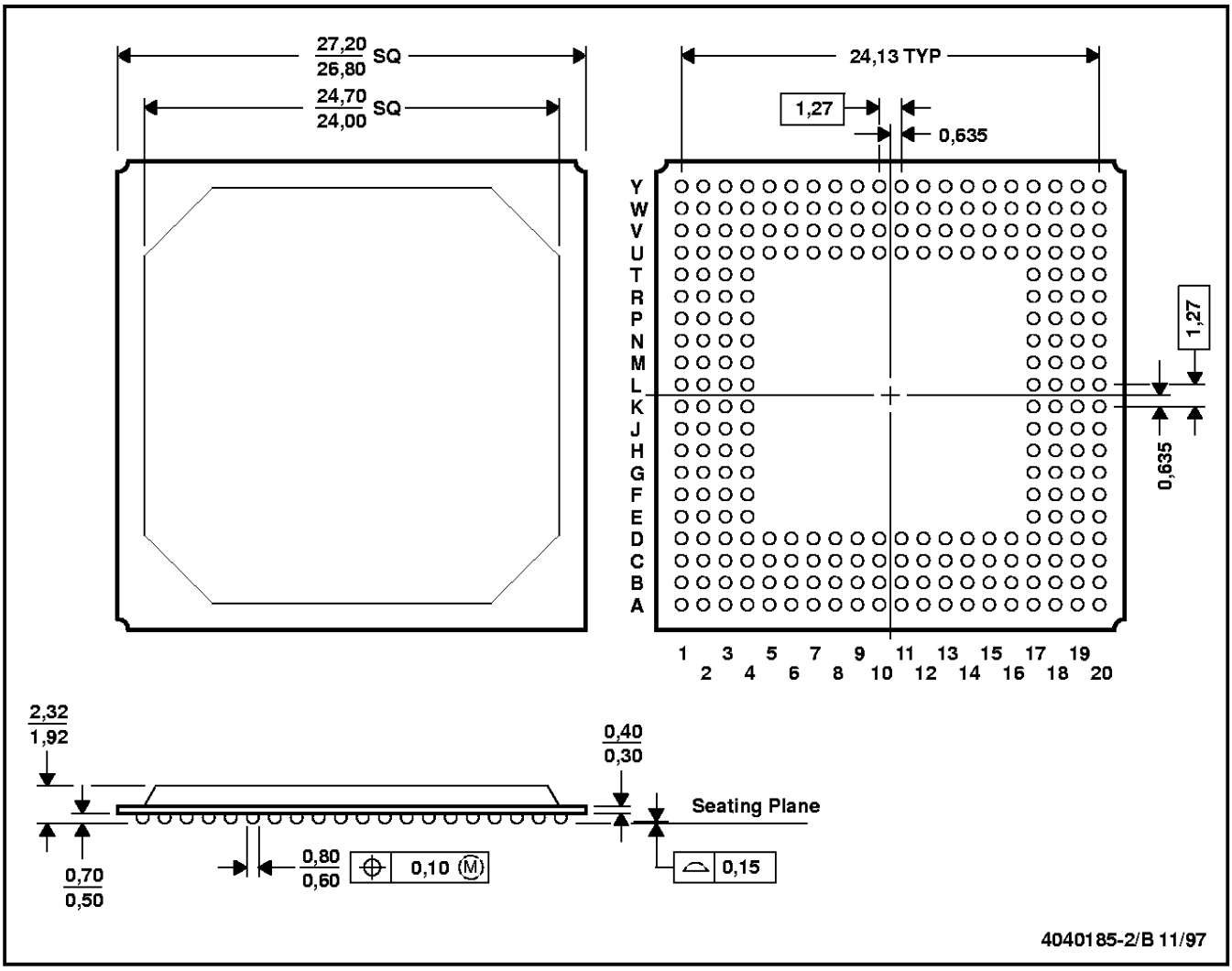


Figure 25. Miscellaneous PC Card Delay Times

MECHANICAL DATA

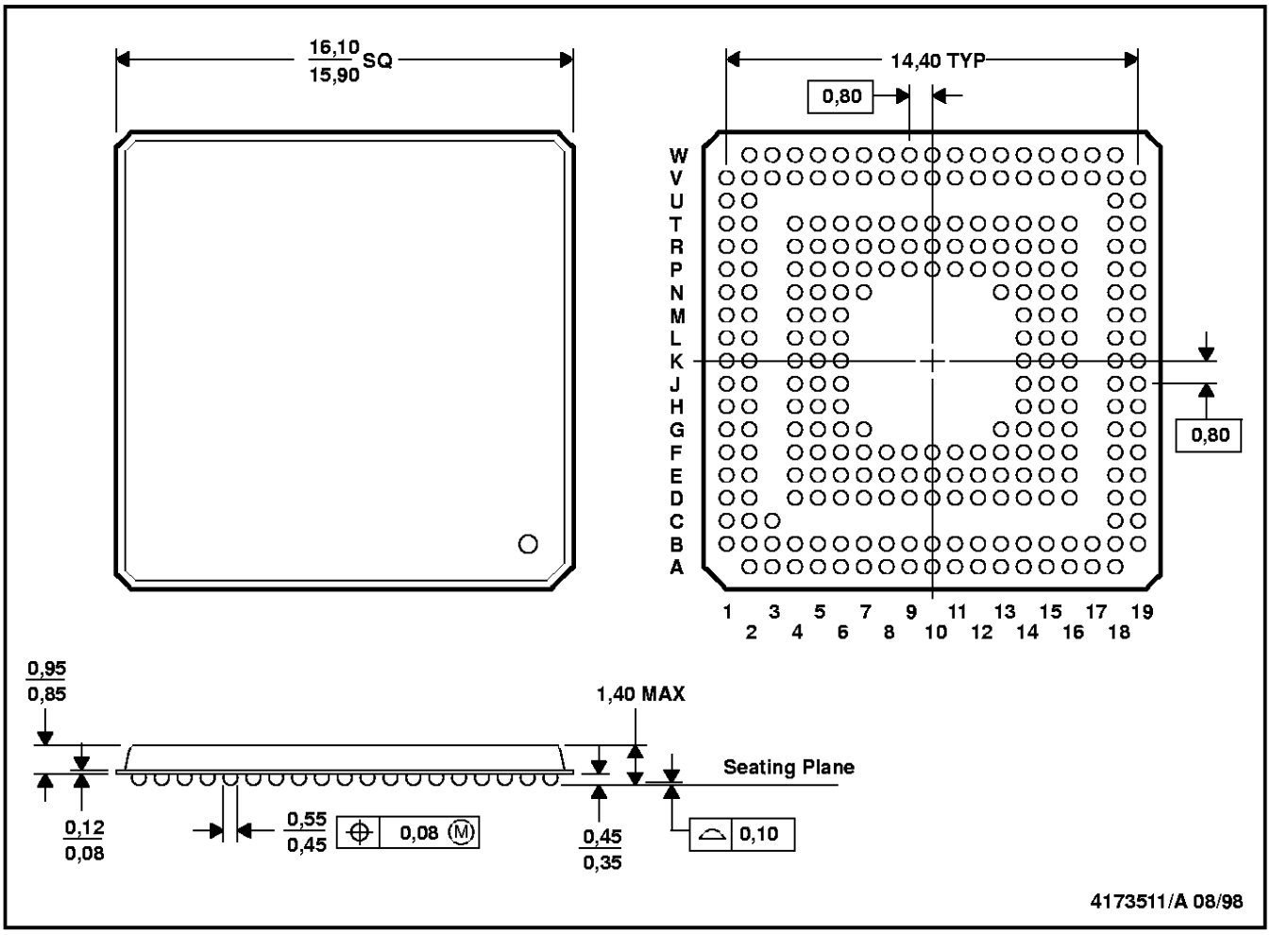
GFN (S-PBGA-N256) **PLASTIC BALL GRID ARRAY**



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

MECHANICAL DATA

GJG (S-PBGA-N257) **PLASTIC BALL GRID ARRAY**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. MicroStar™ BGA configuration

Micro Star is a trademark of Texas Instruments Incorporated.

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

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.

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.