TEXAS INSTRUMENTS

PCI1520-EP

PC Card Controllers

Data Manual

2003

PCIBus Solutions

SGLS168

IMPORTANT NOTICE

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

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

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

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

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

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

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

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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

Contents

List of Illustrations

List of Tables

1 Introduction

1.1 Description

The Texas Instruments PCI1520, a 208-terminal dual-slot CardBus controller designed to meet the *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges*, is an ultralow-power high-performance PCI-to-CardBus controller that supports two independent card sockets compliant with the *PC Card Standard* (rev. 7.1). The PCI1520 provides features that make 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 *PCI Local Bus Specification* and defines the new 32-bit PC Card, CardBus, capable of full 32-bit data transfers at 33 MHz. The PCI1520 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 PCI1520 is compliant with the *PCI Local Bus Specification*, and its PCI interface can act as either a PCI master device or a PCI slave device. The PCI bus mastering is initiated during CardBus PC Card bridging transactions. The PCI1520 is also compliant with *PCI Bus Power Management Interface Specification* (rev. 1.1).

All card signals are internally buffered to allow hot insertion and removal without external buffering. The PCI1520 is register-compatible with the Intel 82365SL-DF and 82365SL ExCA controllers. The PCI1520 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 PCI1520 can also be programmed to accept fast posted writes to improve system-bus utilization.

Multiple system-interrupt signaling options are provided, including parallel PCI, parallel ISA, serialized ISA, and serialized PCI. Furthermore, general-purpose inputs and outputs are provided for the board designer to implement sideband functions. Many other features designed into the PCI1520, such as socket activity light-emitting diode (LED) outputs, are discussed in detail throughout the design specification.

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

1.2 Features

The PCI1520-EP supports the following features:

- Controlled Baseline
  - One Assembly/Test Site, One Fabrication Site
- Extended Temperature Performance of –40°C to 85°C
- Enhanced Diminishing Manufacturing Sources (DMS) Support
- Enhanced Product-Change Notification
- Qualification Pedigree†
- A 209-terminal MicroStar BGA™ ball-grid array (GHK) package
- 2.5-V core logic and 3.3-V I/O with universal PCI interfaces compatible with 3.3-V and 5-V PCI signaling environments
- Integrated low-dropout voltage regulator (LDO-VR) eliminates the need for an external 2.5-V power supply

† Component qualification in accordance with JEDEC and industry standards to ensure reliable operation over an extended temperature range. This includes, but is not limited to, Highly Accelerated Stress Test (HAST) or biased 85/85, temperature cycle, autoclave or unbiased HAST, electromigration, bond intermetallic life, and mold compound life. Such qualification testing should not be viewed as justifying use of this component beyond specified performance and environmental limits.

- Mix-and-match 5-V/3.3-V 16-bit PC Cards and 3.3-V CardBus Cards
- Two PC Card or CardBus slots with hot insertion and removal
- Serial interface to TI™ TPS222X dual-slot PC Card power switch
- Burst transfers to maximize data throughput with CardBus Cards
- Interrupt configurations: parallel PCI, serialized PCI, parallel ISA, and serialized ISA
- Serial EEPROM interface for loading subsystem ID and subsystem vendor ID
- Pipelined architecture for greater than 130-Mbps throughput from CardBus-to-PCI and from PCI-to-CardBus
- Up to five general-purpose I/Os
- Programmable output select for $\overline{\text{CLKRUN}}$
- Multifunction PCI device with separate configuration space for each socket
- Five PCI memory windows and two I/O windows available for each 16-bit interface
- Two I/O windows and two memory windows available to each CardBus socket
- Exchangeable-card-architecture- (ExCA-) compatible registers are mapped in memory and I/O space
- Intel™ 82365SL-DF and 82365SL register compatible
- Ring indicate, $\overline{\text{SUSPEND}}$, PCI $\overline{\text{CLKRUN}}$, and CardBus $\overline{\text{CCLKRUN}}$
- Socket activity LED terminals
- PCI bus lock ($\overline{\text{LOCK}}$)
- Advanced quarter-micron, ultralow-power CMOS technology
- Internal ring oscillator

1.3 Related Documents

- *Advanced Configuration and Power Interface (ACPI) Specification* (revision 1.1)
- *PCI Bus Power Management Interface Specification* (revision 1.1)
- *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges* (revision 0.6)
- *PCI to PCMCIA CardBus Bridge Register Description* (Yenta) (revision 2.1)
- *PCI Local Bus Specification* (revision 2.2)
- *PCI Mobile Design Guide* (revision 1.0)
- *PC Card Standard* (revision 7.1)
- *PC 2001*
- *Serialized IRQ Support for PCI Systems* (revision 6)

1.4 Trademarks

1.5 Ordering Information

TEMPERATURE	PACKAGE	ORDERING NUMBER	TOP-SIDE MARKING
–40°C to 85°C	209-ball PBGA	PCI1520IGHKEP	PCI1520IEP

1.6 PCI1520-EP Data Manual Document History

DATE	PAGE NUMBER	REVISION
05/2003		Original draft

2 Terminal Descriptions

The PCI1520 is available in a 209-terminal MicroStar BGA™ package (GHK). The terminal layout for the GHK package is shown in Figure 2–1.

GHK PLASTIC BALL GRID ARRAY (PBGA) PACKAGE
BOTTOM VIEW

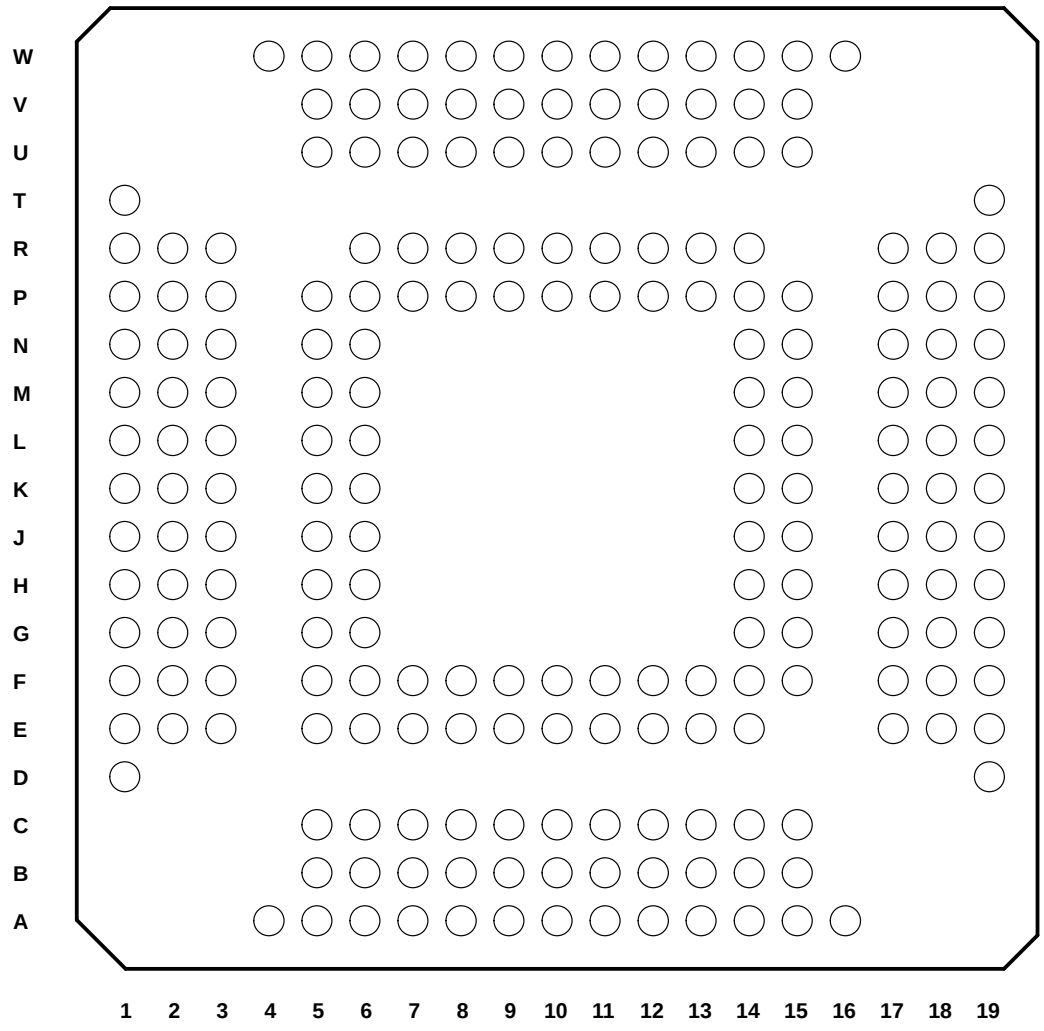


Figure 2–1. PCI1520 GHK-Package Terminal Diagram

Table 2–1 lists the terminal assignments arranged in terminal-number order, with corresponding signal names for both CardBus and 16-bit PC Cards; Table 2–1 is for terminals on the GHK package. Table 2–2 and Table 2–3 list the terminal assignments arranged in alphanumerical order by signal name, with corresponding terminal numbers for GHK package; Table 2–2 is for CardBus signal names and Table 2–3 is for 16-bit PC Card signal names.

Terminal E5 on the GHK package is an identification ball used for device orientation; it has no internal connection within the device.

Table 2–1. Signal Names by GHK Terminal Number

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CardBus PC Card	16-Bit PC Card		CardBus PC Card	16-Bit PC Card		CardBus PC Card	16-Bit PC Card
A04	AD12	AD12	E07	$\overline{\text{PERR}}$	$\overline{\text{PERR}}$	H06	AD2	AD2
A05	PAR	PAR	E08	$\overline{\text{FRAME}}$	$\overline{\text{FRAME}}$	H14	A_CSTSCHG	A_BVD1($\overline{\text{STSCHG}}$/$\overline{\text{RI}}$)
A06	GND	GND	E09	AD19	AD19	H15	A_$\overline{\text{CCLKRUN}}$	A_WP($\overline{\text{IOIS16}}$)
A07	V_{CC}	V_{CC}	E10	IDSEL	IDSEL	H17	A_CAUDIO	A_BVD2($\overline{\text{SPKR}}$)
A08	AD18	AD18	E11	AD27	AD27	H18	A_$\overline{\text{CSERR}}$	A_$\overline{\text{WAIT}}$
A09	GND	GND	E12	AD31	AD31	H19	A_$\overline{\text{CINT}}$	A_READY($\overline{\text{IREQ}}$)
A10	V_{CCP}	V_{CCP}	E13	$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	J01	B_CAD4	B_D12
A11	AD29	AD29	E14	MFUNC2	MFUNC2	J02	B_CAD3	B_D5
A12	V_{CC}	V_{CC}	E17	DATA	DATA	J03	B_CAD6	B_D13
A13	$\overline{\text{REQ}}$	$\overline{\text{REQ}}$	E18	LATCH	LATCH	J05	B_CAD5	B_D6
A14	GND	GND	E19	A_CAD31	A_D10	J06	B_RSVD	B_D14
A15	MFUNC5	MFUNC5	F01	AD3	AD3	J14	A_CAD26	A_A0
A16	MFUNC1	MFUNC1	F02	AD5	AD5	J15	A_CVS1	A_$\overline{\text{VS1}}$
B05	AD15	AD15	F03	AD6	AD6	J17	A_CAD25	A_A1
B06	$\overline{\text{STOP}}$	$\overline{\text{STOP}}$	F05	AD8	AD8	J18	A_CAD24	A_A2
B07	$\overline{\text{IRDY}}$	$\overline{\text{IRDY}}$	F06	C/$\overline{\text{BE1}}$	C/$\overline{\text{BE1}}$	J19	V_{CC}	V_{CC}
B08	AD17	AD17	F07	$\overline{\text{DEVSEL}}$	$\overline{\text{DEVSEL}}$	K01	GND	GND
B09	AD22	AD22	F08	C/$\overline{\text{BE2}}$	C/$\overline{\text{BE2}}$	K02	B_CAD7	B_D7
B10	AD24	AD24	F09	AD20	AD20	K03	B_CAD8	B_D15
B11	AD28	AD28	F10	AD23	AD23	K05	B_CC/$\overline{\text{BE0}}$	B_$\overline{\text{CE1}}$
B12	AD11	AD11	F11	AD26	AD26	K06	B_CAD9	B_A10
B13	$\overline{\text{GNT}}$	$\overline{\text{GNT}}$	F12	AD25	AD25	K14	A_CC/$\overline{\text{BE3}}$	A_$\overline{\text{REG}}$
B14	C/$\overline{\text{BE3}}$	C/$\overline{\text{BE3}}$	F13	MFUNC3/IRQSER	MFUNC3/IRQSER	K15	A_CAD23	A_A3
B15	MFUNC4	MFUNC4	F14	SPKROUT	SPKROUT	K17	A_$\overline{\text{CREQ}}$	A_$\overline{\text{INPACK}}$
C05	AD13	AD13	F15	CLOCK	CLOCK	K18	A_CAD22	A_A4
C06	$\overline{\text{SERR}}$	$\overline{\text{SERR}}$	F17	A_RSVD	A_D2	K19	VR_PORT	VR_PORT
C07	$\overline{\text{TRDY}}$	$\overline{\text{TRDY}}$	F18	A_CAD29	A_D1	L01	$\overline{\text{VR\_EN}}$	$\overline{\text{VR\_EN}}$
C08	AD16	AD16	F19	GND	GND	L02	B_CAD10	B_$\overline{\text{CE2}}$
C09	AD21	AD21	G01	V_{CC}	V_{CC}	L03	B_CAD11	B_$\overline{\text{OE}}$
C10	PCLK	PCLK	G02	AD0	AD0	L05	B_CAD13	B_$\overline{\text{IORD}}$
C11	$\overline{\text{GRST}}$	$\overline{\text{GRST}}$	G03	AD1	AD1	L06	B_CAD12	B_A11
C12	AD30	AD30	G05	AD4	AD4	L14	A_CAD21	A_A5
C13	$\overline{\text{PRST}}$	$\overline{\text{PRST}}$	G06	C/$\overline{\text{BE0}}$	C/$\overline{\text{BE0}}$	L15	A_$\overline{\text{CRST}}$	A_RESET
C14	MFUNC6/ $\overline{\text{CLKRUN}}$	MFUNC6/ $\overline{\text{CLKRUN}}$	G14	A_CAD28	A_D8	L17	A_CAD20	A_A6
C15	$\overline{\text{SUSPEND}}$	$\overline{\text{SUSPEND}}$	G15	A_CAD30	A_D9	L18	A_CVS2	A_$\overline{\text{VS2}}$
D01	AD10	AD10	G17	A_CAD27	A_D0	L19	A_CAD19	A_A25
D19	MFUNC0	MFUNC0	G18	A_$\overline{\text{CCD2}}$	A_$\overline{\text{CD2}}$	M01	B_CAD15	B_$\overline{\text{IOWR}}$
E01	GND	GND	G19	V_{CC}	V_{CC}	M02	B_CAD14	B_A9
E02	AD7	AD7	H01	B_CAD1	B_D4	M03	B_CAD16	B_A17
E03	AD9	AD9	H02	B_CAD2	B_D11	M05	B_RSVD	B_A18
E05	NC	NC	H03	B_CAD0	B_D3	M06	B_CC/$\overline{\text{BE1}}$	B_A8
E06	AD14	AD14	H05	B_$\overline{\text{CCD1}}$	B_$\overline{\text{CD1}}$	M14	A_CCLK	A_A16

Table 2–1. Signal Names by GHK Terminal Number (Continued)

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CardBus PC Card	16-Bit PC Card		CardBus PC Card	16-Bit PC Card		CardBus PC Card	16-Bit PC Card
M15	A_$\overline{\text{CFRAME}}$	A_A23	P17	A_$\overline{\text{CSTOP}}$	A_A20	U13	A_CAD7	A_D7
M17	A_CC/$\overline{\text{BE2}}$	A_A12	P18	A_$\overline{\text{CGNT}}$	A_$\overline{\text{WE}}$	U14	A_CAD10	A_$\overline{\text{CE2}}$
M18	A_CAD17	A_A24	P19	V_{CCA}	V_{CCA}	U15	A_CAD14	A_A9
M19	A_CAD18	A_A7	R01	V_{CCB}	V_{CCB}	V05	B_CAD20	B_A6
N01	V_{CC}	V_{CC}	R02	B_$\overline{\text{CTRDY}}$	B_A22	V06	B_CAD22	B_A4
N02	B_CPAR	B_A13	R03	B_$\overline{\text{CFRAME}}$	B_A23	V07	B_CAD24	B_A2
N03	B_$\overline{\text{CBLOCK}}$	B_A19	R06	B_CAD19	B_A25	V08	B_$\overline{\text{CINT}}$	B_READY($\overline{\text{IREQ}}$)
N05	B_$\overline{\text{CGNT}}$	B_$\overline{\text{WE}}$	R07	B_$\overline{\text{CREQ}}$	B_$\overline{\text{INPACK}}$	V09	B_CAUDIO	B_BVD2($\overline{\text{SPKR}}$)
N06	B_$\overline{\text{CPERR}}$	B_A14	R08	B_CAD26	B_A0	V10	B_CAD28	B_D8
N14	A_$\overline{\text{CBLOCK}}$	A_A19	R09	B_$\overline{\text{CCLKRUN}}$	B_WP($\overline{\text{IOIS16}}$)	V11	B_CAD31	B_D10
N15	A_$\overline{\text{CDEVSEL}}$	A_A21	R10	B_CAD30	B_D9	V12	A_CAD4	A_D12
N17	A_$\overline{\text{CTRDY}}$	A_A22	R11	A_CAD2	A_D11	V13	A_RSVD	A_D14
N18	A_$\overline{\text{CIRDY}}$	A_A15	R12	A_CAD5	A_D6	V14	A_CC/$\overline{\text{BE0}}$	A_$\overline{\text{CE1}}$
N19	V_{CC}	V_{CC}	R13	A_CAD9	A_A10	V15	A_CAD13	A_$\overline{\text{IORD}}$
P01	GND	GND	R14	A_CAD15	A_$\overline{\text{IOWR}}$	W04	B_CAD17	B_A24
P02	B_$\overline{\text{CSTOP}}$	B_A20	R17	A_RSVD	A_A18	W05	B_$\overline{\text{CRST}}$	B_RESET
P03	B_$\overline{\text{CDEVSEL}}$	B_A21	R18	A_$\overline{\text{CPERR}}$	A_A14	W06	GND	GND
P05	B_$\overline{\text{CIRDY}}$	B_A15	R19	GND	GND	W07	B_CAD25	B_A1
P06	B_CCLK	B_A16	T01	B_CC/$\overline{\text{BE2}}$	B_A12	W08	V_{CC}	V_{CC}
P07	B_CVS2	B_$\overline{\text{VS2}}$	T19	A_CC/$\overline{\text{BE1}}$	A_A8	W09	B_$\overline{\text{CSERR}}$	B_$\overline{\text{WAIT}}$
P08	B_CAD23	B_A3	U05	B_CAD18	B_A7	W10	B_CAD27	B_D0
P09	B_$\overline{\text{CCD2}}$	B_$\overline{\text{CD2}}$	U06	B_CAD21	B_A5	W11	NC†	NC†
P10	B_RSVD	B_D2	U07	B_CC/$\overline{\text{BE3}}$	B_$\overline{\text{REG}}$	W12	A_CAD1	A_D4
P11	A_CAD0	A_D3	U08	B_CVS1	B_$\overline{\text{VS1}}$	W13	V_{CC}	V_{CC}
P12	A_CAD6	A_D13	U09	B_CSTSCHG	B_BVD1($\overline{\text{STSCHG}}$/$\overline{\text{RI}}$)	W14	GND	GND
P13	A_CAD8	A_D15	U10	B_CAD29	B_D1	W15	A_CAD11	A_$\overline{\text{OE}}$
P14	A_CAD12	A_A11	U11	A_$\overline{\text{CCD1}}$	A_$\overline{\text{CD1}}$	W16	A_CAD16	A_A17
P15	A_CPAR	A_A13	U12	A_CAD3	A_D5			

† Terminal W11 is an NC on the PCI1520 to allow for terminal compatibility with the next generation of devices.

Table 2–2. CardBus PC Card Signal Names Sorted Alphabetically

SIGNAL NAME	TERM NO. GHK	SIGNAL NAME	TERM. NO. GHK	SIGNAL NAME	TERM. NO. GHK
A_CAD0	P11	A_$\overline{\text{CDEVSEL}}$	N15	AD24	B10
A_CAD1	W12	A_$\overline{\text{CFRAME}}$	M15	AD25	F12
A_CAD2	R11	A_$\overline{\text{CGNT}}$	P18	AD26	F11
A_CAD3	U12	A_$\overline{\text{CINT}}$	H19	AD27	E11
A_CAD4	V12	A_$\overline{\text{CIRDY}}$	N18	AD28	B11
A_CAD5	R12	A_CPAR	P15	AD29	A11
A_CAD6	P12	A_$\overline{\text{CPERR}}$	R18	AD30	C12
A_CAD7	U13	A_$\overline{\text{CREQ}}$	K17	AD31	E12
A_CAD8	P13	A_$\overline{\text{CRST}}$	L15	B_CAD0	H03
A_CAD9	R13	A_$\overline{\text{CSERR}}$	H18	B_CAD1	H01
A_CAD10	U14	A_$\overline{\text{CSTOP}}$	P17	B_CAD2	H02
A_CAD11	W15	A_CSTSCHG	H14	B_CAD3	J02
A_CAD12	P14	A_$\overline{\text{CTRDY}}$	N17	B_CAD4	J01
A_CAD13	V15	A_CVS1	J15	B_CAD5	J05
A_CAD14	U15	A_CVS2	L18	B_CAD6	J03
A_CAD15	R14	A_RSVD	R17	B_CAD7	K02
A_CAD16	W16	A_RSVD	V13	B_CAD8	K03
A_CAD17	M18	A_RSVD	F17	B_CAD9	K06
A_CAD18	M19	AD0	G02	B_CAD10	L02
A_CAD19	L19	AD1	G03	B_CAD11	L03
A_CAD20	L17	AD2	H06	B_CAD12	L06
A_CAD21	L14	AD3	F01	B_CAD13	L05
A_CAD22	K18	AD4	G05	B_CAD14	M02
A_CAD23	K15	AD5	F02	B_CAD15	M01
A_CAD24	J18	AD6	F03	B_CAD16	M03
A_CAD25	J17	AD7	E02	B_CAD17	W04
A_CAD26	J14	AD8	F05	B_CAD18	U05
A_CAD27	G17	AD9	E03	B_CAD19	R06
A_CAD28	G14	AD10	D01	B_CAD20	V05
A_CAD29	F18	AD11	B12	B_CAD21	U06
A_CAD30	G15	AD12	A04	B_CAD22	V06
A_CAD31	E19	AD13	C05	B_CAD23	P08
A_CAUDIO	H17	AD14	E06	B_CAD24	V07
A_$\overline{\text{CBLOCK}}$	N14	AD15	B05	B_CAD25	W07
A_CC/$\overline{\text{BE0}}$	V14	AD16	C08	B_CAD26	R08
A_CC/$\overline{\text{BE1}}$	T19	AD17	B08	B_CAD27	W10
A_CC/$\overline{\text{BE2}}$	M17	AD18	A08	B_CAD28	V10
A_CC/$\overline{\text{BE3}}$	K14	AD19	E09	B_CAD29	U10
A_$\overline{\text{CCD1}}$	U11	AD20	F09	B_CAD30	R10
A_$\overline{\text{CCD2}}$	G18	AD21	C09	B_CAD31	V11
A_CCLK	M14	AD22	B09	B_CAUDIO	V09
A_$\overline{\text{CCLKRUN}}$	H15	AD23	F10	B_$\overline{\text{CBLOCK}}$	N03

Table 2–2. CardBus PC Card Signal Names Sorted Alphabetically (Continued)

SIGNAL NAME	TERM NO. GHK	SIGNAL NAME	TERM NO. GHK	SIGNAL NAME	TERM NO. GHK
B_CC/$\overline{BE0}$	K05	C/$\overline{BE2}$	F08	NC	E05
B_CC/$\overline{BE1}$	M06	C/$\overline{BE3}$	B14	NC†	W11
B_CC/$\overline{BE2}$	T01	CLOCK	F15	PAR	A05
B_CC/$\overline{BE3}$	U07	DATA	E17	PCLK	C10
B_$\overline{CCD1}$	H05	$\overline{DEVSEL}$	F07	$\overline{PERR}$	E07
B_$\overline{CCD2}$	P09	$\overline{FRAME}$	E08	$\overline{PRST}$	C13
B_CCLK	P06	GND	A06	$\overline{REQ}$	A13
B_$\overline{CCLKRUN}$	R09	GND	A09	$\overline{RI\_OUT}$/$\overline{PME}$	E13
B_$\overline{CDEVSEL}$	P03	GND	A14	$\overline{SERR}$	C06
B_$\overline{CFRAME}$	R03	GND	E01	SPKROUT	F14
B_$\overline{CGNT}$	N05	GND	K01	$\overline{STOP}$	B06
B_$\overline{CINT}$	V08	GND	P01	$\overline{SUSPEND}$	C15
B_$\overline{CIRDY}$	P05	GND	R19	$\overline{TRDY}$	C07
B_CPAR	N02	GND	W06	V_{CC}	A07
B_$\overline{CPERR}$	N06	GND	F19	V_{CC}	A12
B_$\overline{CREQ}$	R07	GND	W14	V_{CC}	G01
B_$\overline{CRST}$	W05	$\overline{GNT}$	B13	V_{CC}	G19
B_$\overline{CSERR}$	W09	$\overline{GRST}$	C11	V_{CC}	J19
B_$\overline{CSTOP}$	P02	IDSEL	E10	V_{CC}	N01
B_CSTSCHG	U09	$\overline{IRDY}$	B07	V_{CC}	N19
B_$\overline{CTRDY}$	R02	LATCH	E18	V_{CC}	W08
B_CVS1	U08	MFUNC0	D19	V_{CC}	W13
B_CVS2	P07	MFUNC1	A16	V_{CCA}	P19
B_RSVD	J06	MFUNC2	E14	V_{CCB}	R01
B_RSVD	M05	MFUNC3/IRQSER	F13	V_{CCP}	A10
B_RSVD	P10	MFUNC4	B15	$\overline{VR\_EN}$	L01
C/$\overline{BE0}$	G06	MFUNC5	A15	VR_PORT	K19
C/$\overline{BE1}$	F06	MFUNC6/$\overline{CLKRUN}$	C14		

† Terminals 81 and W11 are NC on the PCI1520 to allow for terminal compatibility with the next generation of devices.

Table 2–3. 16-Bit PC Card Signal Names Sorted Alphabetically

SIGNAL NAME	TERM. NO. GHK	SIGNAL NAME	TERM. NO. GHK	SIGNAL NAME	TERM NO. GHK
A_A0	J14	A_D10	E19	AD24	B10
A_A1	J17	A_D11	R11	AD25	F12
A_A2	J18	A_D12	V12	AD26	F11
A_A3	K15	A_D13	P12	AD27	E11
A_A4	K18	A_D14	V13	AD28	B11
A_A5	L14	A_D15	P13	AD29	A11
A_A6	L17	A_$\overline{\text{INPACK}}$	K17	AD30	C12
A_A7	M19	A_$\overline{\text{IORD}}$	V15	AD31	E12
A_A8	T19	A_$\overline{\text{IOWR}}$	R14	B_A0	R08
A_A9	U15	A_$\overline{\text{OE}}$	W15	B_A1	W07
A_A10	R13	A_READY($\overline{\text{IREQ}}$)	H19	B_A2	V07
A_A11	P14	A_$\overline{\text{REG}}$	K14	B_A3	P08
A_A12	M17	A_RESET	L15	B_A4	V06
A_A13	P15	A_$\overline{\text{VS1}}$	J15	B_A5	U06
A_A14	R18	A_$\overline{\text{VS2}}$	L18	B_A6	V05
A_A15	N18	A_$\overline{\text{WAIT}}$	H18	B_A7	U05
A_A16	M14	A_$\overline{\text{WE}}$	P18	B_A8	M06
A_A17	W16	A_WP($\overline{\text{IOIS16}}$)	H15	B_A9	M02
A_A18	R17	AD0	G02	B_A10	K06
A_A19	N14	AD1	G03	B_A11	L06
A_A20	P17	AD2	H06	B_A12	T01
A_A21	N15	AD3	F01	B_A13	N02
A_A22	N17	AD4	G05	B_A14	N06
A_A23	M15	AD5	F02	B_A15	P05
A_A24	M18	AD6	F03	B_A16	P06
A_A25	L19	AD7	E02	B_A17	M03
A_BVD1($\overline{\text{STSCHG}}$/$\overline{\text{RI}}$)	H14	AD8	F05	B_A18	M05
A_BVD2($\overline{\text{SPKR}}$)	H17	AD9	E03	B_A19	N03
A_$\overline{\text{CD1}}$	U11	AD10	D01	B_A20	P02
A_$\overline{\text{CD2}}$	G18	AD11	B12	B_A21	P03
A_$\overline{\text{CE1}}$	V14	AD12	A04	B_A22	R02
A_$\overline{\text{CE2}}$	U14	AD13	C05	B_A23	R03
A_D0	G17	AD14	E06	B_A24	W04
A_D1	F18	AD15	B05	B_A25	R06
A_D2	F17	AD16	C08	B_BVD1($\overline{\text{STSCHG}}$/$\overline{\text{RI}}$)	U09
A_D3	P11	AD17	B08	B_BVD2($\overline{\text{SPKR}}$)	V09
A_D4	W12	AD18	A08	B_$\overline{\text{CD1}}$	H05
A_D5	U12	AD19	E09	B_$\overline{\text{CD2}}$	P09
A_D6	R12	AD20	F09	B_$\overline{\text{CE1}}$	K05
A_D7	U13	AD21	C09	B_$\overline{\text{CE2}}$	L02
A_D8	G14	AD22	B09	B_D0	W10
A_D9	G15	AD23	F10	B_D1	U10

Table 2–3. 16-Bit PC Card Signal Names Sorted Alphabetically (Continued)

SIGNAL NAME	TERM NO. GHK	SIGNAL NAME	TERM NO. GHK	SIGNAL NAME	TERM NO. GHK
B_D2	P10	C/$\overline{BE2}$	F08	NC	E05
B_D3	H03	C/$\overline{BE3}$	B14	NC†	W11
B_D4	H01	CLOCK	F15	PAR	A05
B_D5	J02	DATA	E17	PCLK	C10
B_D6	J05	$\overline{DEVSEL}$	F07	$\overline{PERR}$	E07
B_D7	K02	$\overline{FRAME}$	E08	$\overline{PRST}$	C13
B_D8	V10	GND	A06	$\overline{REQ}$	A13
B_D9	R10	GND	A09	$\overline{RI\_OUT}$/$\overline{PME}$	E13
B_D10	V11	GND	A14	$\overline{SERR}$	C06
B_D11	H02	GND	E01	SPKROUT	F14
B_D12	J01	GND	K01	$\overline{STOP}$	B06
B_D13	J03	GND	P01	$\overline{SUSPEND}$	C15
B_D14	J06	GND	R19	$\overline{TRDY}$	C07
B_D15	K03	GND	W06	V_{CC}	A07
B_$\overline{INPACK}$	R07	GND	F19	V_{CC}	A12
B_$\overline{IORD}$	L05	GND	W14	V_{CC}	G01
B_$\overline{IOWR}$	M01	$\overline{GNT}$	B13	V_{CC}	G19
B_$\overline{OE}$	L03	$\overline{GRST}$	C11	V_{CC}	J19
B_READY($\overline{IREQ}$)	V08	IDSEL	E10	V_{CC}	N01
B_$\overline{REG}$	U07	$\overline{IRDY}$	B07	V_{CC}	N19
B_RESET	W05	LATCH	E18	V_{CC}	W08
B_$\overline{VS1}$	U08	MFUNC0	D19	V_{CC}	W13
B_$\overline{VS2}$	P07	MFUNC1	A16	V_{CCA}	P19
B_$\overline{WAIT}$	W09	MFUNC2	E14	V_{CCB}	R01
B_$\overline{WE}$	N05	MFUNC3/IRQSER	F13	V_{CCP}	A10
B_WP($\overline{IOIS16}$)	R09	MFUNC4	B15	$\overline{VR\_EN}$	L01
C/$\overline{BE0}$	G06	MFUNC5	A15	VR_PORT	K19
C/$\overline{BE1}$	F06	MFUNC6/$\overline{CLKRUN}$	C14		

† Terminals 81 and W11 are NC on the PCI1520 to allow for terminal compatibility with the next generation of devices.

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 2–4. Power Supply Terminals

TERMINAL		I/O	DESCRIPTION
NAME	NO. GHK		
GND	A06, A09, A14, E01, F19, K01, P01, R19, W06, W14	–	Device ground terminals
V_{CC}	A07, A12, G01, G19, J19, N01, N19, W08, W13	–	Power supply terminal for I/O and internal voltage regulator
V_{CCA}	P19	–	Clamp voltage for PC Card A interface. Matches card A signaling environment, 5 V or 3.3 V
V_{CCB}	R01	–	Clamp voltage for PC Card B interface. Matches card B signaling environment, 5 V or 3.3 V
V_{CCP}	A10	–	Clamp voltage for PCI and miscellaneous I/O, 5 V or 3.3 V
$\overline{VR\_EN}$	L01	I	Internal voltage regulator enable. Active-low
VR_PORT	K19	I/O	Internal voltage regulator input/output. When VR_EN is low, the regulator is enabled and this terminal is an output. An external bypass capacitor is required on this terminal. When VR_EN is high, the regulator is disabled and this terminal is an input for an external 2.5-V core power source.

Table 2–5. PC Card Power Switch Terminals

TERMINAL		I/O	DESCRIPTION
NAME	NO. GHK		
CLOCK	F15	I/O	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 PCI1520 output by using bit 27 (P2CCLK) in the system control register (offset 80h, see Section 4.29). The TPS222X defines the maximum frequency of this signal to be 2 MHz. However, PCI1520 requires a 16-KHz to 100-KHz frequency range. If a system design defines this terminal as an output, then this terminal requires an external pulldown resistor. The frequency of the PCI1520 output CLOCK is derived from the internal ring oscillator (16 KHz typical).
DATA	E17	O	Power switch data. DATA is used to communicate socket power control information serially to the power switch.
LATCH	E18	I/O	Power switch latch. LATCH is asserted by the PCI1520 to indicate to the power switch that the data on the DATA line is valid. When a pulldown resistor is implemented on this terminal, the MFUNC1 and MFUNC4 terminals provide the serial EEPROM SDA and SCL interface.

Table 2–6. PCI System Terminals

TERMINAL		I/O	DESCRIPTION
NAME	NO. GHK		
$\overline{GRST}$	C11	I	Global reset. When the global reset is asserted, the $\overline{GRST}$ signal causes the PCI1520 to place all output buffers in a high-impedance state and reset all internal registers. When $\overline{GRST}$ is asserted, the device is completely in its default state. For systems that require wake-up from D3, $\overline{GRST}$ normally is asserted only during initial boot. $\overline{PRST}$ should be asserted following initial boot so that PME context is retained during the transition from D3 to D0. For systems that do not require wake-up from D3, $\overline{GRST}$ should be tied to $\overline{PRST}$. When the $\overline{SUSPEND}$ mode is enabled, the device is protected from $\overline{GRST}$, and the internal registers are preserved. All outputs are placed in a high-impedance state.
PCLK	C10	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.
$\overline{PRST}$	C13	I	PCI reset. When the PCI bus reset is asserted, $\overline{PRST}$ causes the PCI1520 to place all output buffers in a high-impedance state and reset internal registers. When $\overline{PRST}$ is asserted, the device can generate the $\overline{PME}$ signal only if it is enabled. After $\overline{PRST}$ is deasserted, the PCI1520 is in a default state. When the $\overline{SUSPEND}$ mode is enabled, the device is protected from $\overline{PRST}$, and the internal registers are preserved. All outputs are placed in a high-impedance state.

Table 2–7. PCI Address and Data Terminals

TERMINAL NAME	TERMINAL NO. GHK	I/O	DESCRIPTION
AD31 AD30 AD29 AD28 AD27 AD26 AD25 AD24 AD23 AD22 AD21 AD20 AD19 AD18 AD17 AD16 AD15 AD14 AD13 AD12 AD11 AD10 AD9 AD8 AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0	E12 C12 A11 B11 E11 F11 F12 B10 F10 B09 C09 F09 E09 A08 B08 C08 B05 E06 C05 A04 B12 D01 E03 F05 E02 F03 F02 G05 F01 H06 G03 G02	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.
C/$\overline{BE3}$ C/$\overline{BE2}$ C/$\overline{BE1}$ C/$\overline{BE0}$	B14 F08 F06 G06	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).
PAR	A05	I/O	PCI-bus parity. In all PCI-bus read and write cycles, the PCI1520 calculates even parity across the AD31–AD0 and C/$\overline{BE3}$–C/$\overline{BE0}$ buses. As an initiator during PCI cycles, the PCI1520 outputs this parity indicator with a one-PCLK delay. As a target during PCI cycles, the PCI1520 compares its calculated parity to the parity indicator of the initiator. A compare error results in the assertion of a parity error ($\overline{PERR}$).

Table 2–8. PCI Interface Control Terminals

TERMINAL NAME	TERMINAL NO. GHK	I/O	DESCRIPTION
$\overline{\text{DEVSEL}}$	F07	I/O	PCI device select. The PCI1520 asserts $\overline{\text{DEVSEL}}$ to claim a PCI cycle as the target device. As a PCI initiator on the bus, the PCI1520 monitors $\overline{\text{DEVSEL}}$ until a target responds. If no target responds before timeout occurs, then the PCI1520 terminates the cycle with an initiator abort.
$\overline{\text{FRAME}}$	E08	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.
$\overline{\text{GNT}}$	B13	I	PCI bus grant. $\overline{\text{GNT}}$ is driven by the PCI bus arbiter to grant the PCI1520 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.
IDSEL	E10	I	Initialization device select. IDSEL selects the PCI1520 during configuration space accesses. IDSEL can be connected to one of the upper 24 PCI address lines on the PCI bus.
$\overline{\text{IRDY}}$	B07	I/O	PCI initiator ready. $\overline{\text{IRDY}}$ indicates the ability of the PCI bus initiator to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCLK where both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. Until $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are both sampled asserted, wait states are inserted.
$\overline{\text{PERR}}$	E07	I/O	PCI parity error indicator. $\overline{\text{PERR}}$ is driven by a PCI device to indicate that calculated parity does not match PAR when $\overline{\text{PERR}}$ is enabled through bit 6 of the command register (PCI offset 04h, see Section 4.4).
$\overline{\text{REQ}}$	A13	O	PCI bus request. $\overline{\text{REQ}}$ is asserted by the PCI1520 to request access to the PCI bus as an initiator.
$\overline{\text{SERR}}$	C06	O	PCI system error. $\overline{\text{SERR}}$ is an output that is pulsed from the PCI1520 when enabled through bit 8 of the command register (PCI offset 04h, see Section 4.4) indicating a system error has occurred. The PCI1520 need not be the target of the PCI cycle to assert this signal. When $\overline{\text{SERR}}$ is enabled in the command register, this signal also pulses, indicating that an address parity error has occurred on a CardBus interface.
$\overline{\text{STOP}}$	B06	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.
$\overline{\text{TRDY}}$	C07	I/O	PCI target ready. $\overline{\text{TRDY}}$ indicates the ability of the primary bus target to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCLK when both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. Until both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted, wait states are inserted.

Table 2–9. Multifunction and Miscellaneous Terminals

TERMINAL NAME	TERMINAL NO. GHK	I/O	DESCRIPTION
MFUNC0	D19	I/O	Multifunction terminal 0. MFUNC0 can be configured as parallel PCI interrupt INTA, GPI0, GPO0, socket activity LED output, ZV switching output, CardBus audio PWM, GPE, or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details.
MFUNC1	A16	I/O	Multifunction terminal 1. MFUNC1 can be configured as parallel PCI interrupt INTB, GPI1, GPO1, socket activity LED output, ZV switching output, CardBus audio PWM, GPE, or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details. Serial data (SDA). When LATCH is detected low after the deassertion of GRST, the MFUNC1 terminal provides the SDA signaling for the serial bus interface. The two-terminal serial interface loads the subsystem identification and other register defaults from an EEPROM after a PCI reset. See Section 3.6.1, *Serial Bus Interface Implementation*, for details on other serial bus applications.
MFUNC2	E14	I/O	Multifunction terminal 2. MFUNC2 can be configured as GPI2, GPO2, socket activity LED output, ZV switching output, CardBus audio PWM, GPE, RI_OUT, D3_STAT, or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details.
MFUNC3/ IRQSER	F13	I/O	Multifunction terminal 3. MFUNC3 can be configured as a parallel IRQ or the serialized interrupt signal IRQSER. This terminal is IRQSER by default. See Section 4.30, *Multifunction Routing Register*, for configuration details.
MFUNC4	B15	I/O	Multifunction terminal 4. MFUNC4 can be configured as PCI LOCK, GPI3, GPO3, socket activity LED output, ZV switching output, CardBus audio PWM, GPE, D3_STAT, RI_OUT, or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details. Serial clock (SCL). When LATCH is detected low after the deassertion of GRST, the MFUNC4 terminal provides the SCL signaling for the serial bus interface. The two-terminal serial interface loads the subsystem identification and other register defaults from an EEPROM after a PCI reset. See Section 3.6.1, *Serial Bus Interface Implementation*, for details on other serial bus applications.
MFUNC5	A15	I/O	Multifunction terminal 5. MFUNC5 can be configured as GPI4, GPO4, socket activity LED output, ZV switching output, CardBus audio PWM, D3_STAT, GPE, or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details.
MFUNC6/ CLKRUN	C14	I/O	Multifunction terminal 6. MFUNC6 can be configured as a PCI CLKRUN or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details.
NC	E05 W11		No connect. These terminals have no connection anywhere within the package. Terminal E05 on the GHK package is used as a key to indicate the location of the A1 corner of the BGA package. Terminals W11 on the GHK package and 81 on the PDV package will be used as a 48-MHz clock input on future-generation devices.
RI_OUT/PME	E13	O	Ring indicate out and power management event output. This terminal provides an output for ring-indicate or PME signals.
SPKROUT	F14	O	Speaker output. SPKROUT is the output to the host system that can carry SPKR or CAUDIO through the PCI1520 from the PC Card interface. SPKROUT is driven as the exclusive-OR combination of card SPKR//CAUDIO inputs.
SUSPEND	C15	I	Suspend. SUSPEND protects the internal registers from clearing when the GRST or PRST signal is asserted. See Section 3.8.5, *Suspend Mode*, for details.

Table 2–10. 16-Bit PC Card Address and Data Terminals (Slots A and B)

TERMINAL			I/O	DESCRIPTION
NAME	NUMBER			
	SLOT A†	SLOT B‡		
	GHK	GHK		
A25	L19	R06	O	PC Card address. 16-bit PC Card address lines. A25 is the most significant bit.
A24	M18	W04		
A23	M15	R03		
A22	N17	R02		
A21	N15	P03		
A20	P17	P02		
A19	N14	N03		
A18	R17	M05		
A17	W16	M03		
A16	M14	P06		
A15	N18	P05		
A14	R18	N06		
A13	P15	N02		
A12	M17	T01		
A11	P14	L06		
A10	R13	K06		
A9	U15	M02		
A8	T19	M06		
A7	M19	U05		
A6	L17	V05		
A5	L14	U06		
A4	K18	V06		
A3	K15	P08		
A2	J18	V07		
A1	J17	W07		
A0	J14	R08		
D15	P13	K03	I/O	PC Card data. 16-bit PC Card data lines. D15 is the most significant bit.
D14	V13	J06		
D13	P12	J03		
D12	V12	J01		
D11	R11	H02		
D10	E19	V11		
D9	G15	R10		
D8	G14	V10		
D7	U13	K02		
D6	R12	J05		
D5	U12	J02		
D4	W12	H01		
D3	P11	H03		
D2	F17	P10		
D1	F18	U10		
D0	G17	W10		

† Terminal name for slot A is preceded with A_. For example, the full name for terminals 123 and L19 is A_A25.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 55 and R06 is B_A25.

Table 2–11. 16-Bit PC Card Interface Control Terminals (Slots A and B)

TERMINAL			I/O	DESCRIPTION
NAME	NUMBER			
	SLOT A†	SLOT B‡		
	GHK	GHK		
BVD1 (STSCHG/RI)	H14	U09	I	Battery voltage detect 1. BVD1 is generated by 16-bit memory PC Cards that include batteries. BVD1 is used with BVD2 as an indication of the condition of the batteries on a memory PC Card. Both BVD1 and BVD2 are high when the battery is good. When BVD2 is low and BVD1 is high, the battery is weak and should be replaced. When BVD1 is low, the battery is no longer serviceable and the data in the memory PC Card is lost. See Section 5.6, *ExCA Card Status-Change Interrupt Configuration Register*, for enable bits. See Section 5.5, *ExCA Card Status-Change Register*, and Section 5.2, *ExCA Interface Status Register*, for the status bits for this signal. Status change. $\overline{\text{STSCHG}}$ 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. $\overline{\text{RI}}$ is used by 16-bit modem cards to indicate a ring detection.
BVD2 (SPKR)	H17	V09	I	Battery voltage detect 2. BVD2 is generated by 16-bit memory PC Cards that include batteries. BVD2 is used with BVD1 as an indication of the condition of the batteries on a memory PC Card. Both BVD1 and BVD2 are high when the battery is good. When BVD2 is low and BVD1 is high, the battery is weak and should be replaced. When BVD1 is low, the battery is no longer serviceable and the data in the memory PC Card is lost. See Section 5.6, *ExCA Card Status-Change Interrupt Configuration Register*, for enable bits. See Section 5.5, *ExCA Card Status-Change Register*, and Section 5.2, *ExCA Interface Status Register*, for the status bits for this signal. Speaker. $\overline{\text{SPKR}}$ 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 PCI1520 and are output on SPKROUT.
$\overline{\text{CD1}}$ $\overline{\text{CD2}}$	U11 G18	H05 P09	I	Card detect 1 and card detect 2. $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ are internally connected to ground on the PC Card. When a PC Card is inserted into a socket, $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ are pulled low. For signal status, see Section 5.2, *ExCA Interface Status Register*.
$\overline{\text{CE1}}$ $\overline{\text{CE2}}$	V14 U14	K05 L02	O	Card enable 1 and card enable 2. $\overline{\text{CE1}}$ and $\overline{\text{CE2}}$ enable even- and odd-numbered address bytes. $\overline{\text{CE1}}$ enables even-numbered address bytes, and $\overline{\text{CE2}}$ enables odd-numbered address bytes.
$\overline{\text{INPACK}}$	K17	R07	I	Input acknowledge. $\overline{\text{INPACK}}$ is asserted by the PC Card when it can respond to an I/O read cycle at the current address.
$\overline{\text{IORD}}$	V15	L05	O	I/O read. $\overline{\text{IORD}}$ is asserted by the PCI1520 to enable 16-bit I/O PC Card data output during host I/O read cycles.
$\overline{\text{IOWR}}$	R14	M01	O	I/O write. $\overline{\text{IOWR}}$ is driven low by the PCI1520 to strobe write data into 16-bit I/O PC Cards during host I/O write cycles.

† Terminal name for slot A is preceded with A_. For example, the full name for terminals 130 and K17 is A_$\overline{\text{INPACK}}$.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 61 and R07 is B_$\overline{\text{INPACK}}$.

Table 2–11. 16-Bit PC Card Interface Control Terminals (Slots A and B) (Continued)

TERMINAL NAME	TERMINAL NUMBER SLOT A† GHK	TERMINAL NUMBER SLOT B‡ GHK	I/O	DESCRIPTION
$\overline{OE}$	W15	L03	O	Output enable. $\overline{OE}$ is driven low by the PCI1520 to enable 16-bit memory PC Card data output during host memory read cycles.
READY ($\overline{IREQ}$)	H19	V08	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 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.
$\overline{REG}$	K14	U07	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.
RESET	L15	W05	O	PC Card reset. RESET forces a hard reset to a 16-bit PC Card.
$\overline{VS1}$ $\overline{VS2}$	J15 L18	U08 P07	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 PC Card.
$\overline{WAIT}$	H18	W09	I	Bus cycle wait. $\overline{WAIT}$ is driven by a 16-bit PC Card to extend the completion of the memory or I/O cycle in progress.
$\overline{WE}$	P18	N05	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.
WP ($\overline{IOIS16}$)	H15	R09	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.

† Terminal name for slot A is preceded with A_. For example, the full name for terminals 112 and P18 is A_$\overline{WE}$.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 45 and N05 is B_$\overline{WE}$.

Table 2–12. CardBus PC Card Interface System Terminals (Slots A and B)

TERMINAL NAME	TERMINAL NUMBER SLOT A† GHK	TERMINAL NUMBER SLOT B‡ GHK	I/O	DESCRIPTION
CCLK	M14	P06	O	CardBus clock. CCLK provides synchronous timing for all transactions on the CardBus interface. All signals except $\overline{CRST}$, $\overline{CCLKRUN}$, $\overline{CINT}$, CSTSCHG, CAUDIO, $\overline{CCD2}$, $\overline{CCD1}$, CVS2, and CVS1 are sampled on the rising edge of CCLK, and all timing parameters are defined with the rising edge of this signal. CCLK operates at the PCI bus clock frequency, but it can be stopped in the low state or slowed down for power savings.
$\overline{CCLKRUN}$	H15	R09	I/O	CardBus clock run. $\overline{CCLKRUN}$ is used by a CardBus PC Card to request an increase in the CCLK frequency, and by the PCI1520 to indicate that the CCLK frequency is going to be decreased.
$\overline{CRST}$	L15	W05	O	CardBus reset. $\overline{CRST}$ brings CardBus PC Card-specific registers, sequencers, and signals to a known state. When $\overline{CRST}$ is asserted, all CardBus PC Card signals are placed in a high-impedance state, and the PCI1520 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 terminals 115 and M14 is A_CCLK.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 48 and P06 is B_CCLK.

Table 2–13. CardBus PC Card Address and Data Terminals (Slots A and B)

TERMINAL NAME	TERMINAL NUMBER SLOT A† GHK	TERMINAL NUMBER SLOT B‡ GHK	I/O	DESCRIPTION
CAD31	E19	V11	I/O	CardBus address and data. These signals make up the multiplexed CardBus address and data bus on the CardBus interface. During the address phase of a CardBus cycle, CAD31–CAD0 contain a 32-bit address. During the data phase of a CardBus cycle, CAD31–CAD0 contain data. CAD31 is the most significant bit.
CAD30	G15	R10		
CAD29	F18	U10		
CAD28	G14	V10		
CAD27	G17	W10		
CAD26	J14	R08		
CAD25	J17	W07		
CAD24	J18	V07		
CAD23	K15	P08		
CAD22	K18	V06		
CAD21	L14	U06		
CAD20	L17	V05		
CAD19	L19	R06		
CAD18	M19	U05		
CAD17	M18	W04		
CAD16	W16	M03		
CAD15	R14	M01		
CAD14	U15	M02		
CAD13	V15	L05		
CAD12	P14	L06		
CAD11	W15	L03		
CAD10	U14	L02		
CAD9	R13	K06		
CAD8	P13	K03		
CAD7	U13	K02		
CAD6	P12	J03		
CAD5	R12	J05		
CAD4	V12	J01		
CAD3	U12	J02		
CAD2	R11	H02		
CAD1	W12	H01		
CAD0	P11	H03		
CC/$\overline{\text{BE3}}$	K14	U07	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/$\overline{\text{BE0}}$ define the bus command. During the data phase, this 4-bit bus is used as byte enables. The byte enables determine which byte paths of the full 32-bit data bus carry meaningful data. CC/$\overline{\text{BE0}}$ applies to byte 0 (CAD7–CAD0), CC/$\overline{\text{BE1}}$ applies to byte 1 (CAD15–CAD8), CC/$\overline{\text{BE2}}$ applies to byte 2 (CAD23–CAD16), and CC/$\overline{\text{BE3}}$ applies to byte 3 (CAD31–CAD24).
CC/$\overline{\text{BE2}}$	M17	T01		
CC/$\overline{\text{BE1}}$	T19	M06		
CC/$\overline{\text{BE0}}$	V14	K05		
CPAR	P15	N02	I/O	CardBus parity. In all CardBus read and write cycles, the PCI1520 calculates even parity across the CAD and CC/$\overline{\text{BE}}$ buses. As an initiator during CardBus cycles, the PCI1520 outputs CPAR with a one-CCLK delay. As a target during CardBus cycles, the PCI1520 compares its calculated parity to the parity indicator of the initiator; a compare error results in a parity error assertion.

† Terminal name for slot A is preceded with A_. For example, the full name for terminals 107 and P15 is A_CPAR.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 40 and N02 is B_CPAR.

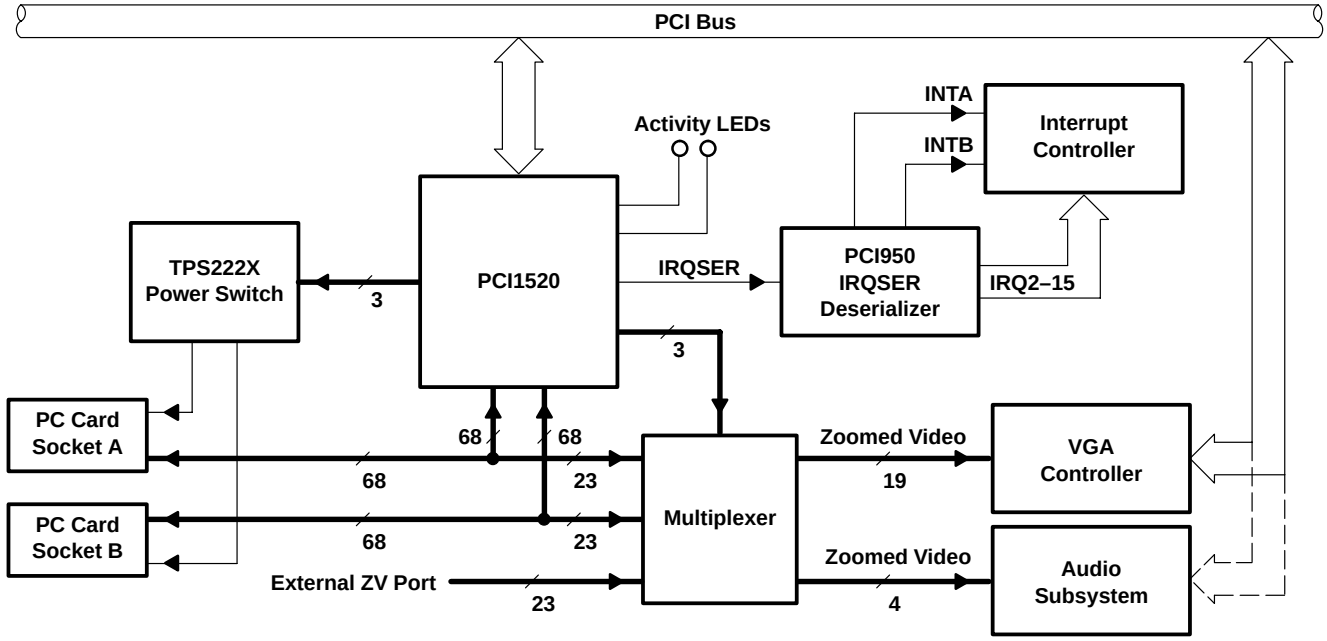
Table 2–14. CardBus PC Card Interface Control Terminals (Slots A and B)

TERMINAL			I/O	DESCRIPTION
NAME	NUMBER			
	SLOT A†	SLOT B‡		
	GHK	GHK		
CAUDIO	H17	V09	I	CardBus audio. CAUDIO is a digital input signal from a PC Card to the system speaker. The PCI1520 supports the binary audio mode and outputs a binary signal from the card to SPKROUT.
$\overline{\text{CBLOCK}}$	N14	N03	I/O	CardBus lock. $\overline{\text{CBLOCK}}$ is used to gain exclusive access to a target.
$\overline{\text{CCD1}}$ $\overline{\text{CCD2}}$	U11 G18	H05 P09	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.
$\overline{\text{CDEVSEL}}$	N15	P03	I/O	CardBus device select. The PCI1520 asserts $\overline{\text{CDEVSEL}}$ to claim a CardBus cycle as the target device. As a CardBus initiator on the bus, the PCI1520 monitors $\overline{\text{CDEVSEL}}$ until a target responds. If no target responds before timeout occurs, then the PCI1520 terminates the cycle with an initiator abort.
$\overline{\text{CFRAME}}$	M15	R03	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.
$\overline{\text{CGNT}}$	P18	N05	O	CardBus bus grant. $\overline{\text{CGNT}}$ is driven by the PCI1520 to grant a CardBus PC Card access to the CardBus bus after the current data transaction has been completed.
$\overline{\text{CINT}}$	H19	V08	I	CardBus interrupt. $\overline{\text{CINT}}$ is asserted low by a CardBus PC Card to request interrupt servicing from the host.
$\overline{\text{CIRDY}}$	N18	P05	I/O	CardBus initiator ready. $\overline{\text{CIRDY}}$ indicates the ability of the CardBus initiator to complete the current data phase of the transaction. A data phase is completed on a rising edge of CCLK when both $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are asserted. Until $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are both sampled asserted, wait states are inserted.
$\overline{\text{CPERR}}$	R18	N06	I/O	CardBus parity error. $\overline{\text{CPERR}}$ reports parity errors during CardBus transactions, except during special cycles. It is driven low by a target two clocks following the data cycle during which a parity error is detected.
$\overline{\text{CREQ}}$	K17	R07	I	CardBus request. $\overline{\text{CREQ}}$ indicates to the arbiter that the CardBus PC Card desires use of the CardBus bus as an initiator.
$\overline{\text{CSERR}}$	H18	W09	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; deassertion may take several CCLK periods. The PCI1520 can report $\overline{\text{CSERR}}$ to the system by assertion of $\overline{\text{SERR}}$ on the PCI interface.
$\overline{\text{CSTOP}}$	P17	P02	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	H14	U09	I	CardBus status change. CSTSCHG alerts the system to a change in the card status, and is used as a wake-up mechanism.
$\overline{\text{CTRDY}}$	N17	R02	I/O	CardBus target ready. $\overline{\text{CTRDY}}$ indicates the ability of the CardBus target to complete the current data phase of the transaction. A data phase is completed on a rising edge of CCLK, when both $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are asserted; until this time, wait states are inserted.
CVS1 CVS2	J15 L18	U08 P07	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 terminals 140 and H18 is A_CAUDIO.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 72 and V09 is B_CAUDIO.

3 Feature/Protocol Descriptions

The following sections give an overview of the PCI1520. Figure 3–1 shows a simplified block diagram of the PCI1520. The PCI interface includes all address/data and control signals for PCI protocol. The interrupt interface includes terminals for parallel PCI, parallel ISA, and serialized PCI and ISA signaling. Miscellaneous system interface terminals include multifunction terminals: $\overline{\text{SUSPEND}}$, $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ (power management control signal), and SPKROUT.



NOTE: The PC Card interface is 68 terminals for CardBus and 16-bit PC Cards. In zoomed video mode 23 terminals are used for routing the zoomed video signals to the VGA controller and audio subsystem.

Figure 3–1. PCI1520 Simplified Block Diagram

3.1 Power Supply Sequencing

The PCI1520 contains 3.3-V I/O buffers with 5-V tolerance requiring an I/O power supply and an LDO-VR power supply for core logic. The core power supply, which is always 2.5 V, can be supplied through the VR_PORT terminal (when $\overline{\text{VR\_EN}}$ is high) or from the integrated LDO-VR. The LDO-VR needs a 3.3-V power supply via the V_{CC} terminals. The clamping voltages (V_{CCA}, V_{CCB}, and V_{CCP}) 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. Assert $\overline{\text{GRST}}$ to the device to disable the outputs during power up. Output drivers must be powered up in the high-impedance state to prevent high current levels through the clamp diodes to the 5-V clamping rails (V_{CCA}, V_{CCB}, and V_{CCP}).
2. Apply 3.3-V power to V_{CC}.
3. Apply the clamp voltage.

The power-down sequence is:

1. Assert $\overline{\text{GRST}}$ to the device to disable the outputs during power down. Output drivers must be powered down in the high-impedance state to prevent high current levels through the clamp diodes to the 5-V clamping rails (V_{CCA}, V_{CCB}, and V_{CCP}).
2. Remove the clamp voltage.
3. Remove the 3.3-V power from V_{CC}.

NOTE: The clamp voltage can be ramped up or ramped down along with the 3.3-V power. The voltage difference between V_{CC} and the clamp voltage must remain within 3.6 V.

3.2 I/O Characteristics

Figure 3–2 shows a 3-state bidirectional buffer. Section 7.2, *Recommended Operating Conditions*, provides the electrical characteristics of the inputs and outputs.

NOTE: The PCI1520 meets the ac specifications of the *1997 PC Card Standard* and *PCI Local Bus Specification.*

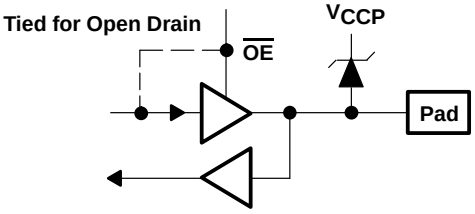


Figure 3–2. 3-State Bidirectional Buffer

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

3.3 Clamping Voltages

The clamping voltages are set to match whatever external environment the PCI1520 is interfaced with, 3.3 V or 5 V. The I/O sites can be pulled through a clamping diode to a voltage rail that protects the core from external signals. The core power supply 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 PCI1520 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, then V_{CCP} can be connected to a 5-V power supply.

The PCI1520 requires three separate clamping voltages because it supports a wide range of features. The three voltages are listed and defined in Section 7.2, *Recommended Operating Conditions*. $\overline{\text{GRST}}$, $\overline{\text{SUSPEND}}$, $\overline{\text{PME}}$, and CSTSCHG are not clamped to any of them.

3.4 Peripheral Component Interconnect (PCI) Interface

The PCI1520 is fully compliant with the *PCI Local Bus Specification*. The PCI1520 provides all required signals for PCI master or slave operation, and may operate in either a 5-V or 3.3-V signaling environment by connecting the V_{CCP} terminal to the desired voltage level. In addition to the mandatory PCI signals, the PCI1520 provides the optional interrupt signals $\overline{\text{INTA}}$ and $\overline{\text{INTB}}$.

3.4.1 PCI $\overline{\text{GRST}}$ Signal

During the power-up sequence, $\overline{\text{GRST}}$ and $\overline{\text{PRST}}$ must be asserted. $\overline{\text{GRST}}$ can only be deasserted 100 µs after PCLK is stable. $\overline{\text{PRST}}$ can be deasserted at the same time as $\overline{\text{GRST}}$ or any time thereafter.

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

The bus-locking protocol defined in the *PCI Local Bus Specification* is not highly recommended, but is provided on the PCI1520 as an additional compatibility feature. The PCI $\overline{\text{LOCK}}$ signal can be routed to the MFUNC4 terminal by setting the appropriate values in bits 19–16 of the multifunction routing register. See Section 4.30, *Multifunction Routing Register*, for details. 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 $\overline{\text{LOCK}}$ protocol defined by the *PCI Local Bus Specification* allows a resource lock without interfering with nonexclusive real-time data transfer, such as video.

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

The PCI1520 supports all $\overline{\text{LOCK}}$ protocol associated with PCI-to-PCI bridges, as also defined for PCI-to-CardBus bridges. This includes disabling write posting while a locked operation is in progress, which can solve a potential deadlock when using devices such as PCI-to-PCI bridges. The potential deadlock can occur if a CardBus target supports delayed transactions and blocks access to the target until it completes a delayed read. This target characteristic is prohibited by the *PCI Local Bus Specification*, and the issue is resolved by the PCI master using $\overline{\text{LOCK}}$.

3.4.3 Loading Subsystem Identification

The subsystem vendor ID register (PCI offset 40h, see Section 4.26) and subsystem ID register (PCI offset 42h, see Section 4.27) make up a doubleword of PCI configuration space for functions 0 and 1. This doubleword register is used for system and option card (mobile dock) identification purposes and is required by some operating systems. Implementation of this unique identifier register is a *PC 99/PC 2001* requirement.

The PCI1520 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 bit 5 (SUBSYSRW) in the system control register (PCI offset 80h, see Section 4.29). Once this bit is cleared, the BIOS can write a subsystem identification value into the registers at PCI offset 40h. The BIOS must set the SUBSYSRW bit such that the subsystem vendor ID register and subsystem ID register is limited to read-only access. This approach saves the added cost of implementing the serial electrically erasable programmable ROM (EEPROM).

In some conditions, such as in a docking environment, the subsystem vendor ID register and subsystem ID register must be loaded with a unique identifier via a serial EEPROM. The PCI1520 loads the 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 PCI1520 core, including the serial-bus state machine (see Section 3.8.5, *Suspend Mode*, for details on using $\overline{\text{SUSPEND}}$).

The PCI1520 provides a two-line serial-bus host controller that can interface to a serial EEPROM. See Section 3.6, *Serial-Bus Interface*, for details on the two-wire serial-bus controller and applications.

3.5 PC Card Applications

This section describes the PC Card interfaces of the PCI1520.

- Card insertion/removal and recognition
- P^2C power-switch interface
- Zoomed video support
- Speaker and audio applications
- LED socket activity indicators
- CardBus socket registers

3.5.1 PC Card Insertion/Removal and Recognition

The *PC Card Standard* (release 7.1) addresses the card-detection and recognition process through an interrogation procedure that the socket must initiate on card insertion into a cold, nonpowered socket. Through this interrogation, card voltage requirements and interface (16-bit versus CardBus) are determined.

The scheme uses the card-detect and voltage-sense signals. The configuration of these four terminals identifies the card type and voltage requirements of the PC Card interface. The encoding scheme is defined in the *PC Card Standard* (release 7.1) and in Table 3–1.

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

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

3.5.2 P^2C Power-Switch Interface (TPS222X)

The PCI1520 provides a PCMCIA peripheral control (P^2C) interface for control of the PC Card power switch. The CLOCK, DATA, and LATCH terminals interface with the TI TPS222X dual-slot PC Card power interface switches to provide power switch support. Figure 3–3 illustrates a typical application where the PCI1520 represents the PCMCIA controller. Table 3–2 shows the available power switch options compatible with the PCI1520.

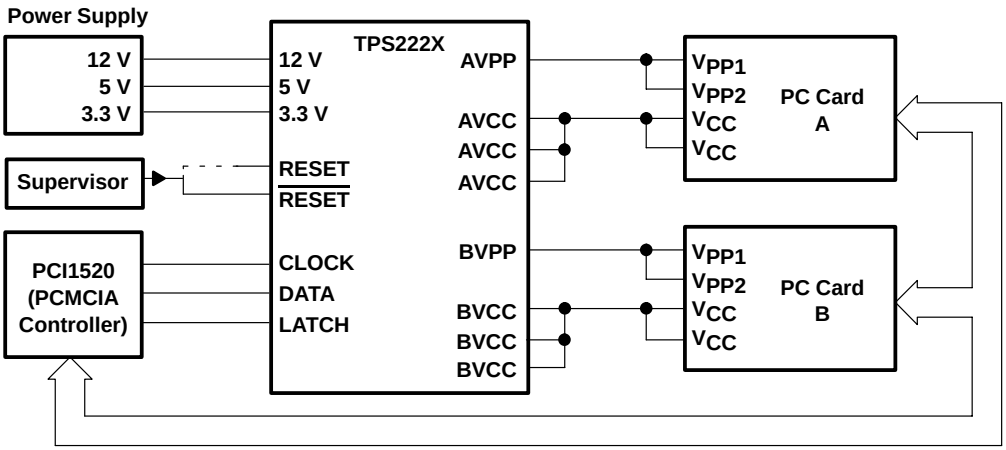


Figure 3–3. TPS222X Typical Application

Table 3–2. Power Switch Options

DEVICE	PIN-COMPATIBLE REPLACEMENT(S)
TPS2206	TPS2226IDB† – 30-pin SSOP TPS2216ADAP – 32-pin TSSOP
TPS2214(A)	TPS2224IDB† – 24-pin SSOP
TPS2216(A)	TPS2226IDB† – 30 pin SSOP
TPS2223†‡	N/A – Check for newer device
TPS2224†	N/A – Check for newer device
TPS2226†	N/A – Check for newer device

† Recommended for new designs
‡ For applications not requiring 12 volts

The CLOCK terminal on the PCI1520 can be an input or an output. The PCI1520 defaults the CLOCK terminal as an input to control the serial interface and the internal state machine. Bit 27 (P2CCLK) in the system control register (offset 80h, see Section 4.29) can be set by the platform BIOS or the serial EEPROM to enable the PCI1520 to generate and drive CLOCK internally from the PCI clock. When the system design implements CLOCK as an output from the PCI1520, an external pulldown resistor is required.

3.5.3 Zoomed Video Support

The PCI1520 allows for the implementation of zoomed video (ZV) for PC Cards. Zoomed video is supported by setting bit 6 (ZVENABLE) in the card control register (PCI offset 91h, see Section 4.32) on a per-socket function basis. Setting this bit puts 16-bit PC Card address lines A25–A4 of the PC Card interface in the high-impedance state. These lines can then transfer video and audio data directly to the appropriate controller. Card address lines A3–A0 can still access PC Card CIS registers for PC Card configuration. Figure 3–4 illustrates a PCI1520 ZV implementation.

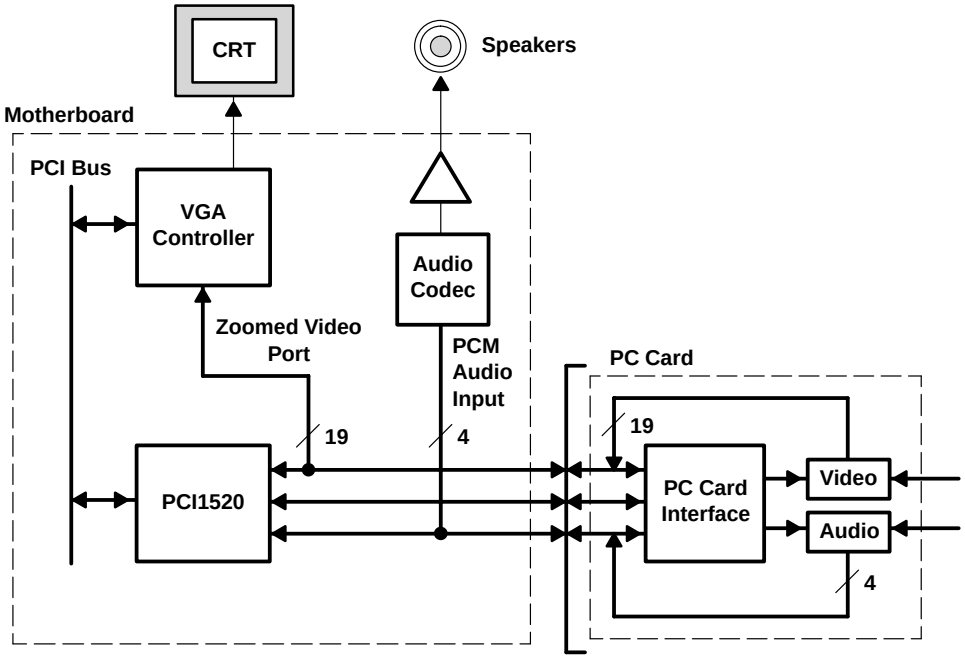


Figure 3–4. Zoomed Video Implementation Using the PCI1520

Not shown in Figure 3–4 is the multiplexing scheme used to route either socket 0 or socket 1 ZV source to the graphics controller. The PCI1520 provides ZVSTAT, $\overline{\text{ZVSEL0}}$, and $\overline{\text{ZVSEL1}}$ signals on the multifunction terminals to switch external bus drivers. Figure 3–5 shows an implementation for switching between three ZV streams using external logic.

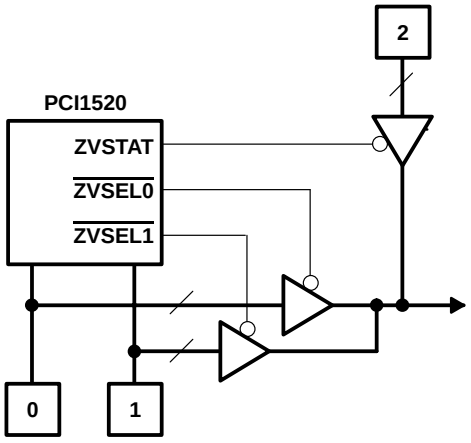


Figure 3–5. Zoomed Video Switching Application

Figure 3–5 illustrates an implementation using standard three-state bus drivers with active-low output enables. $\overline{\text{ZVSEL0}}$ is an active-low output indicating that the socket 0 ZV mode is enabled, and $\overline{\text{ZVSEL1}}$ is an active-low output indicating that socket 1 ZV is enabled. When both sockets have ZV mode enabled, the PCI1520 by defaults indicates socket 0 enabled through $\overline{\text{ZVSEL0}}$; however, bit 5 (PORT_SEL) in the card control register (see Section 4.32) allows software to select the socket ZV source priority. Table 3–3 illustrates the functionality of the ZV output signals.

Table 3–3. Functionality of the ZV Output Signals

INPUTS			OUTPUTS		
PORTSEL	SOCKET 0 ENABLE	SOCKET 1 ENABLE	$\overline{\text{ZVSEL0}}$	$\overline{\text{ZVSEL1}}$	ZVSTAT
X	0	0	1	1	0
0	1	X	0	1	1
0	0	1	1	0	1
1	X	1	1	0	1
1	1	0	0	1	1

Also shown in Figure 3–5 is a third ZV input that can be provided from a source such as a high-speed serial bus like IEEE 1394. The ZVSTAT signal provides a mechanism to switch the third ZV source. ZVSTAT is an active-high output indicating that one of the PCI1520 sockets is enabled for ZV mode. The implementation shown in Figure 3–5 can be used if PC Card ZV is prioritized over other sources.

3.5.4 Standardized Zoomed-Video Register Model

The standardized zoomed-video register model is defined for the purpose of standardizing the ZV port control for PC Card controllers across the industry. The following list summarizes the standardized zoomed-video register model changes to the existing PC Card register set.

- Socket present state register (CardBus socket address + 08h, see Section 6.3)
  Bit 27 (ZVSUPPORT) has been added. The platform BIOS can set this bit via the socket force event register (CardBus socket address + 0Ch, see Section 6.4) to define whether zoomed video is supported on that socket by the platform.
- Socket force event register (CardBus socket address + 0Ch, see Section 6.4)
  Bit 27 (FZVSUPPORT) has been added. The platform BIOS can use this bit to set the ZVSUPPORT bit in the socket present state register (CardBus socket address + 08h, see Section 6.3) to define whether zoomed video is supported on that socket by the platform.
- Socket control register (CardBus socket address +10h, see Section 6.5)
  Bit 11 (ZV_ACTIVITY) has been added. This bit is set when zoomed video is enabled for either of the PC Card sockets.

  Bit 10 (STDZVREG) has been added. This bit defines whether the PC Card controller supports the standardized zoomed-video register model.

  Bit 9 (ZVEN) is provided for software to enable or disable zoomed video, per socket.

If the STDZVEN bit (bit 0) in the diagnostic register (PCI offset 93h, see Section 4.34) is 1, then the standardized zoomed video register model is disabled. For backward compatibility, even if the STDZVEN bit is 0 (enabled), the PCI1520 allows software to access zoomed video through the legacy address in the card control register (PCI offset 91h, see Section 4.32), or through the new register model in the socket control register (CardBus socket address + 10h, see Section 6.5).

3.5.4.1 Zoomed-Video Card Insertion and Configuration Procedure

1. A zoomed-video PC Card is inserted into an empty slot.
2. The card is detected and interrogated appropriately.

There are two types of PC Card controllers to consider.

- Legacy controller not using the standardized ZV register model
  Software reads bit 10 (STDZVREG) of the socket control register (CardBus socket address + 10h) to determine if the standardized zoomed-video register model is supported. If the bit returns 0, then software must use legacy code to enable zoomed video.
- Newer controller that uses the standardized ZV register model
  Software reads bit 10 (STDZVREG) of the socket control register (CardBus socket address + 10h) to determine if the standardized zoomed-video register model is supported. If the bit returns 1, then software can use the process/register model detailed in Table 3–4 to enable zoomed video.

Table 3–4. Zoomed-Video Card Interrogation

ZVSUPPORT (this socket)	ZVSUPPORT (other socket)	ZV_ACTIVITY	ACTION
1	X	0	Set ZVEN to enable zoomed video.
1	X	1	Display a user message such as, “The zoomed video protocol required by this PC Card application is already in use by another card.”
0	0	X	Display a user message such as, “This platform does not support the zoomed-video protocol required by this PC Card application.”
0	1	X	Display a user message such as, “This platform does not support the zoomed-video protocol required by this PC Card application in this PC Card socket. Please remove the card and re-insert in the other PC Card socket.”

3.5.5 Internal Ring Oscillator

The internal ring oscillator provides an internal clock source for the PCI1520 so that neither the PCI clock nor an external clock is required in order for the PCI1520 to power down a socket or interrogate a PC Card. This internal oscillator, operating nominally at 16 kHz, can be enabled by setting bit 27 (P2CCLK) of the system control register (PCI offset 80h, see Section 4.29) to 1. This function is disabled by default.

3.5.6 Integrated Pullup Resistors

The *PC Card Standard* (release 7.1) requires pullup resistors on various terminals to support both CardBus and 16-bit card configurations. Unlike the PCI12XX, PCI1450, and PCI4450 which required external pullup resistors, the PCI1520 has integrated all of these pullup resistors. The I/O buffer on the BVD1($\overline{\text{STSCHG}}$)/CSTSCHG terminal has the capability to switch either pullup or pulldown. The pullup resistor is turned on when a 16-bit PC Card is inserted, and the pulldown resistor is turned on when a CardBus PC Card is inserted. This prevents unexpected CSTSCHG signal assertion. The integrated pullup resistors are listed in Table 3–5.

Table 3–5. Integrated Pullup Resistors

SIGNAL NAME	TERM. NUMBER SOCKET A		TERM. NUMBER SOCKET B	
	PDV	GHK	PDV	GHK
A14/$\overline{\text{CPERR}}$	109	R18	42	N06
A15/$\overline{\text{CIRDY}}$	117	N18	50	P05
A19/$\overline{\text{CBLOCK}}$	108	N14	41	N03
A20/$\overline{\text{CSTOP}}$	111	P17	44	P02
A21/$\overline{\text{CDEVSEL}}$	113	N15	46	P03
A22/$\overline{\text{CTRDY}}$	116	N17	49	R02
BVD1($\overline{\text{STSCHG}}$)/CSTSCHG	141	H14	73	U09
BVD2($\overline{\text{SPKR}}$)/CAUDIO	140	H17	72	V09
$\overline{\text{CD1}}$/$\overline{\text{CCD1}}$	83	U11	15	H05
$\overline{\text{CD2}}$/$\overline{\text{CCD2}}$	144	G18	75	P09
$\overline{\text{INPACK}}$/$\overline{\text{CREQ}}$	130	K17	61	R07
READY/$\overline{\text{CINT}}$	138	H19	69	V08
RESET/$\overline{\text{CRST}}$	126	L15	58	W05
$\overline{\text{VS1}}$/CVS1	137	J15	68	U08
$\overline{\text{VS2}}$/CVS2	124	L18	56	P07
$\overline{\text{WAIT}}$/$\overline{\text{CSERR}}$	139	H18	71	W09
WP($\overline{\text{IOIS16}}$)/$\overline{\text{CCLKRUN}}$	142	H15	74	R09

3.5.7 SPKROUT and CAUDPWM Usage

SPKROUT carries the digital audio signal from the PC Card to the system. When a 16-bit PC Card is configured for I/O mode, the BVD2 terminal becomes $\overline{\text{SPKR}}$. This terminal is also used in CardBus binary audio applications, and is referred to as CAUDIO. $\overline{\text{SPKR}}$ passes a TTL-level digital audio signal to the PCI1520. 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 PCI1520 to produce SPKROUT. This output is enabled by bit 1 (SPKROUTEN) in the card control register (PCI offset 91h, see Section 4.32).

Older controllers support CAUDIO in binary or PWM mode but use the same terminal (SPKROUT). Some audio chips may not support both modes on one terminal and may have a separate terminal for binary and PWM. The PCI1520 implementation includes a signal for PWM, CAUDPWM, which can be routed to an MFUNC terminal. Bit 2 (AUD2MUX), located in the card control register, is programmed on a per-socket function basis to route a CardBus CAUDIO PWM terminal to CAUDPWM. If both CardBus functions enable CAUDIO PWM routing to CAUDPWM, then socket 0 audio takes precedence. See Section 4.30, *Multifunction Routing Register*, for details on configuring the MFUNC terminals.

Figure 3–6 provides an illustration of a sample application using SPKROUT and CAUDPWM.

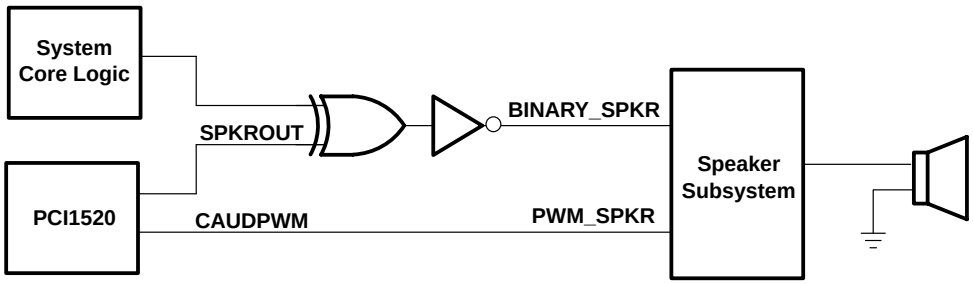


Figure 3–6. Sample Application of SPKROUT and CAUDPWM

3.5.8 LED Socket Activity Indicators

The socket activity LEDs are provided to indicate when a PC Card is being accessed. The LEDA1 and LEDA2 signals can be routed to the multifunction terminals. 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_SKT output indicates socket activity to either socket 0 or socket 1. See Section 4.30, *Multifunction Routing Register*, for details on configuring the multifunction terminals.

The active-high LED signal is driven for 64-ms. When the LED is not being driven high, it is driven to a low state. Either of the two circuits shown in Figure 3–7 can be implemented to provide LED signaling, and the board designer must implement the circuit that best fits the application.

The LED activity signals are valid when a card is inserted, powered, and not in reset. For PC Card-16, the LED activity signals are pulsed when READY/$\overline{\text{IREQ}}$ is low. For CardBus cards, the LED activity signals are pulsed if $\overline{\text{CFRAME}}$, $\overline{\text{IRDY}}$, or $\overline{\text{CREQ}}$ are active.

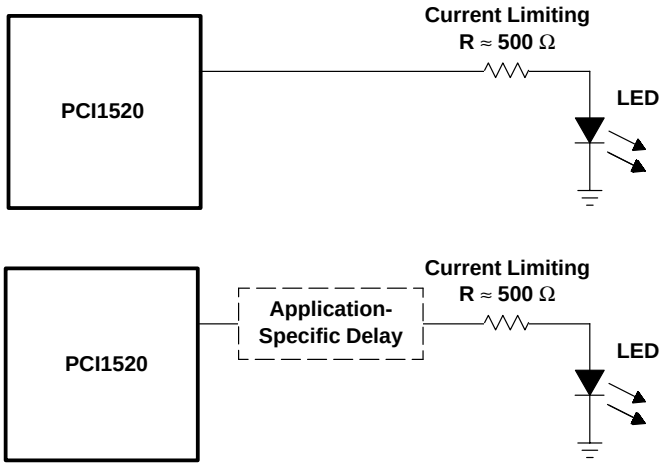


Figure 3–7. Two Sample LED Circuits

As indicated, the LED signals are driven for a period of 64 ms by a counter circuit. To avoid the possibility of the LEDs appearing to be stuck when the PCI clock is stopped, the LED signaling is cut off when the $\overline{\text{SUSPEND}}$ signal is asserted, when the PCI clock is to be stopped during the clock run protocol, or when in the D2 or D1 power state.

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

3.5.9 CardBus Socket Registers

The PCI1520 contains all registers for compatibility with the *1997 PC Card Standard*. These registers exist as the CardBus socket registers and are listed in Table 3–6.

Table 3–6. CardBus Socket Registers

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

3.6 Serial-Bus Interface

The PCI1520 provides a serial-bus interface to load subsystem identification information and selected register defaults from a serial EEPROM, and to provide a PC Card power-switch interface alternative to P^2C. See Section 3.5.2, *P^2C Power-Switch Interface (TPS222X)*, for details. The PCI1520 serial-bus interface is compatible with various I^2C and SMBus components.

3.6.1 Serial-Bus Interface Implementation

The PCI1520 defaults to serial bus interface are disabled. To enable the serial interface, a pulldown resistor must be implemented on the LATCH terminal and the appropriate pullup resistor must be implemented on the SDA and SCL signals, that is, the MFUNC1 and MFUNC4 terminals. When the interface is detected, bit 3 (SBDETECT) in the serial bus control and status register (see Section 4.48) is set. The SBDETECT bit is cleared by a writeback of 1.

The PCI1520 implements a two-terminal serial interface with one clock signal (SCL) and one data signal (SDA). When a pulldown resistor is provided on the LATCH terminal, the SCL signal is mapped to the MFUNC4 terminal and the SDA signal is mapped to the MFUNC1 terminal. The PCI1520 drives SCL at nearly 100 kHz during data transfers, which is the maximum specified frequency for standard mode I^2C. The serial EEPROM must be located at address A0h. Figure 3–8 illustrates an example application implementing the two-wire serial bus.

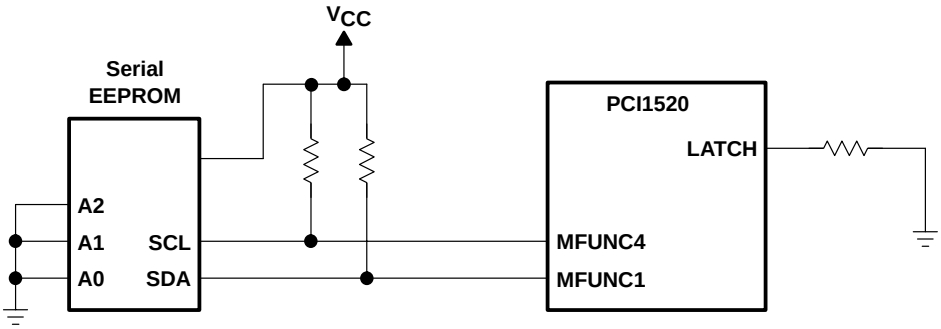


Figure 3–8. Serial EEPROM Application

Some serial device applications may include PC Card power switches, ZV source switches, card ejectors, or other devices that may enhance the user's PC Card experience. The serial EEPROM device and PC Card power switches are discussed in the sections that follow.

3.6.2 Serial-Bus Interface Protocol

The SCL and SDA signals are bidirectional, open-drain signals and require pullup resistors as shown in Figure 3–8. The PCI1520, which supports up to 100-Kb/s data-transfer rate, is compatible with standard mode I^2C using 7-bit addressing.

All data transfers are initiated by the serial bus master. The beginning of a data transfer is indicated by a start condition, which is signaled when the SDA line transitions to low state while SCL is in the high state, as illustrated

in Figure 3–9. The end of a requested data transfer is indicated by a stop condition, which is signaled by a low-to-high transition of SDA while SCL is in the high state, as shown in Figure 3–9. Data on SDA must remain stable during the high state of the SCL signal, as changes on the SDA signal during the high state of SCL are interpreted as control signals, that is, a start or a stop condition.

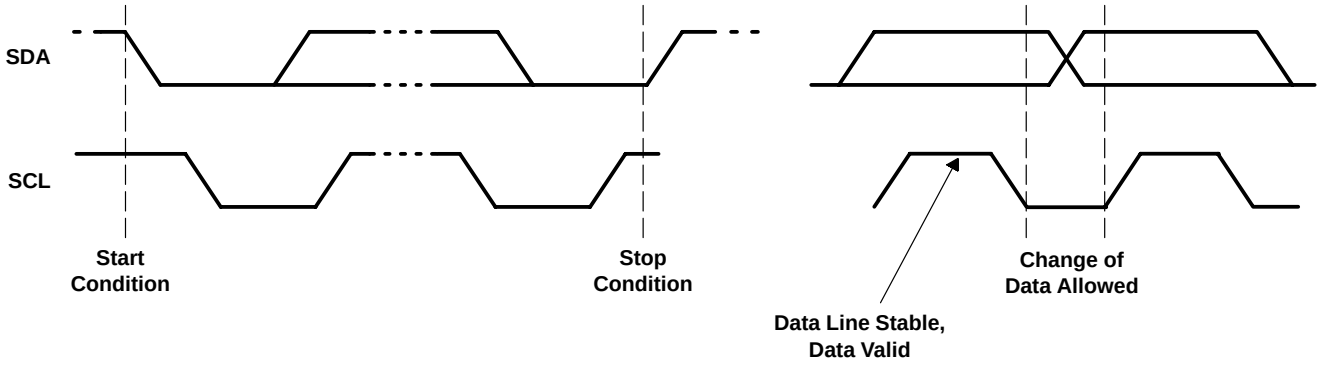


Figure 3–9. Serial-Bus Start/Stop Conditions and Bit Transfers

Data is transferred serially in 8-bit bytes. The number of bytes that may be transmitted during a data transfer is unlimited; however, each byte must be completed with an acknowledge bit. An acknowledge (ACK) is indicated by the receiver pulling the SDA signal low, so that it remains low during the high state of the SCL signal. Figure 3–10 illustrates the acknowledge protocol.

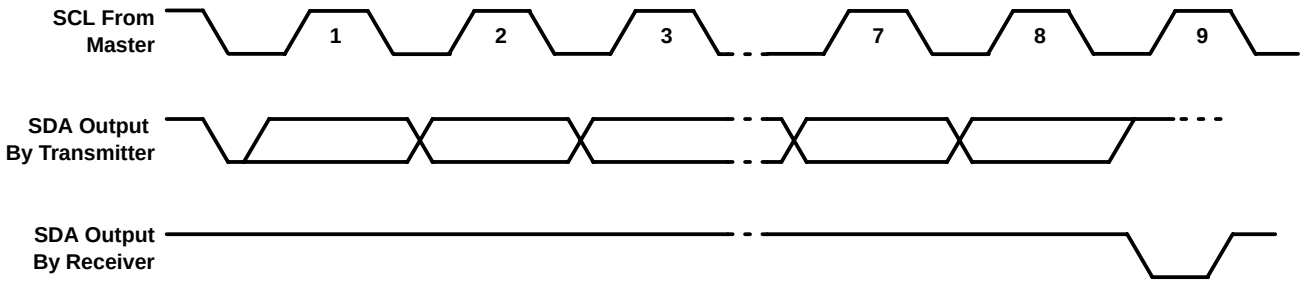


Figure 3–10. Serial-Bus Protocol Acknowledge

The PCI1520 is a serial bus master; all other devices connected to the serial bus external to the PCI1520 are slave devices. As the bus master, the PCI1520 drives the SCL clock at nearly 100 kHz during bus cycles and places SCL in a high-impedance state (zero frequency) during idle states.

Typically, the PCI1520 masters byte reads and byte writes under software control. Doubleword reads are performed by the serial EEPROM initialization circuitry upon a PCI reset and may not be generated under software control. See Section 3.6.3, *Serial-Bus EEPROM Application*, for details on how the PCI1520 automatically loads the subsystem identification and other register defaults through a serial-bus EEPROM.

Figure 3–11 illustrates a byte write. The PCI1520 issues a start condition and sends the 7-bit slave device address and the command bit zero. A 0 in the R/$\overline{W}$ command bit indicates that the data transfer is a write. The slave device acknowledges if it recognizes the address. If no acknowledgment is received by the PCI1520, then an appropriate status bit is set in the serial-bus control and status register (PCI offset B3h, see Section 4.48). The word address byte is then sent by the PCI1520, and another slave acknowledgment is expected. Then the PCI1520 delivers the data byte MSB first and expects a final acknowledgment before issuing the stop condition.

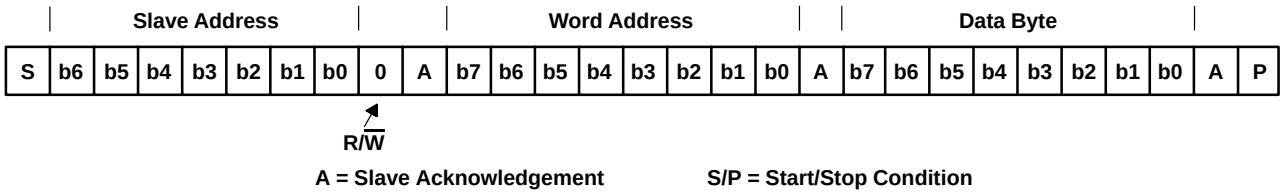


Figure 3–11. Serial-Bus Protocol – Byte Write

Figure 3–12 illustrates a byte read. The read protocol is very similar to the write protocol, except the R/$\overline{W}$ command bit must be set to 1 to indicate a read-data transfer. In addition, the PCI1520 master must acknowledge reception of the read bytes from the slave transmitter. The slave transmitter drives the SDA signal during read data transfers. The SCL signal remains driven by the PCI1520 master.

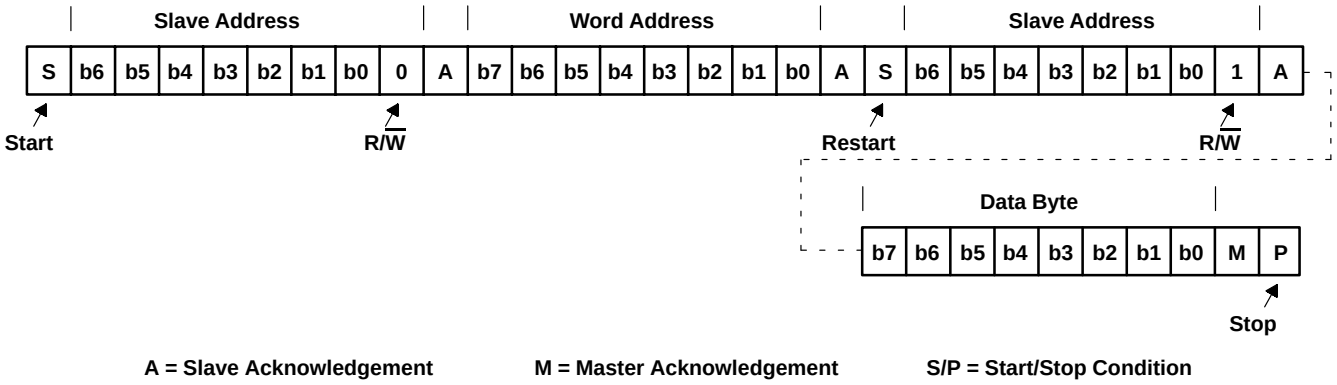


Figure 3–12. Serial-Bus Protocol – Byte Read

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

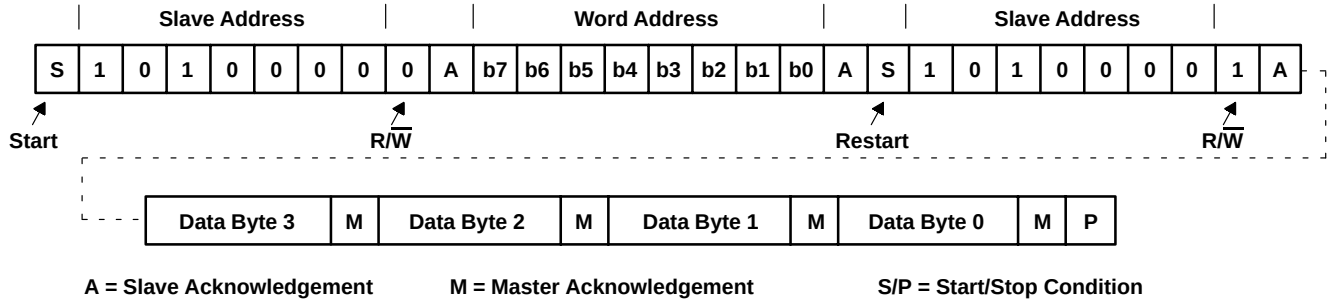


Figure 3–13. EEPROM Interface Doubleword Data Collection

3.6.3 Serial-Bus EEPROM Application

When the PCI bus is reset and the serial-bus interface is detected, the PCI1520 attempts to read the subsystem identification and other register defaults from a serial EEPROM. The registers and corresponding bits that can be loaded with defaults through the EEPROM are provided in Table 3–7.

Table 3–7. Register- and Bit-Loading Map

EEPROM OFFSET	REGISTER OFFSET	REGISTER BITS LOADED FROM EEPROM
00h	Flag	01h: Load / FFh: do not load
01h	PCI 04h	Command register, bits 8, 6–5, 2–0 Note: bits loaded per following: b8 ← b7 b6 ← b6 b5 ← b5 b2 ← b2 b1 ← b1 b0 ← b0
02h	PCI 40h	Subsystem vendor ID bits 7–0 ← bits 7–0
03h	PCI 40h	Subsystem vendor ID bits 15–8 ← bit 7–0
04h	PCI 42h	Subsystem ID bits 7–0 ← bits 7–0
05h	PCI 42h	Subsystem ID bits 15–8 ← bits 7–0
06h	PCI 44h	PC Card 16-bit I/F legacy-mode base address bits 7–1 ← bits 7–1
07h	PCI 44h	PC Card 16-bit I/F legacy-mode base address bits 15–8 ← bits 7–0
08h	PCI 44h	PC Card 16-bit I/F legacy-mode base address bit 23:16 ← bit 7:0
09h	PCI 44h	PC Card 16-bit I/F legacy-mode base address bits 31–24 ← bits 7–0
0Ah	PCI 80h	System control bits 7–0 ← bits 7–0
0Bh	PCI 80h	System control bits 15–8 ← bits 7–0
0Ch	PCI 80h	System control byte bits 31–24 ← bits 7–0
0Dh	PCI 8Ch	Multifunction routing bits 7–0 ← bits 7–0
0Eh	PCI 8Ch	Multifunction routing bits 15–8 ← bits 7–0
0Fh	PCI 8Ch	Multifunction routing bits 23–16 ← bits 7–0
10h	PCI 8Ch	Multifunction routing bits 27–24 ← bits 3–0
11h	PCI 90h	Retry status bits 7, 6 ← bits 7, 6
12h	PCI 91h	Card control bits 7, 5 ← bits 7, 6
13h	PCI 92h	Device control bits 6, 3–0 ← bits 6, 3–0
14h	PCI 93h	Diagnostic bits 7, 4–0 ← bits 7, 4–0
15h	PCI A2h	Power management capabilities bit 15 ← bit 7
16h	ExCA 00h	ExCA identification and revision bits 7–0 ← bits 7–0
17h	CB Socket + 0Ch (function 0)	Function 0 socket force event, bit 27 ← bit 3
18h	CB Socket + 0Ch (function 1)	Function 1 socket force event, bit 27 ← bit 3

This format must be followed for the PCI1520 to load initializations from a serial EEPROM. All bit fields must be considered when programming the EEPROM.

The serial EEPROM is addressed at slave address 1010 000b by the PCI1520. All hardware address bits for the EEPROM should be tied to the appropriate level to achieve this address. The serial EEPROM chip in the sample application circuit (Figure 3–8) assumes the 1010b high-address nibble. The lower three address bits are terminal inputs to the chip, and the sample application shows these terminal inputs tied to GND.

3.6.4 Accessing Serial-Bus Devices Through Software

The PCI1520 provides a programming mechanism to control serial bus devices through software. The programming is accomplished through a doubleword of PCI configuration space at offset B0h. Table 3–8 lists the registers used to program a serial-bus device through software.

Table 3–8. PCI1520 Registers Used to Program Serial-Bus Devices

PCI OFFSET	REGISTER NAME	DESCRIPTION
B0h	Serial-bus data	Contains the data byte to send on write commands or the received data byte on read commands.
B1h	Serial-bus index	The content of this register is sent as the word address on byte writes or reads. This register is not used in the quick command protocol.
B2h	Serial-bus slave address	Write transactions to this register initiate a serial-bus transaction. The slave device address and the R/$\overline{W}$ command selector are programmed through this register.
B3h	Serial-bus control and status	Read data valid, general busy, and general error status are communicated through this register. In addition, the protocol-select bit is programmed through this register.

3.7 Programmable Interrupt Subsystem

Interrupts provide a way for I/O devices to let the microprocessor know that they require servicing. The dynamic nature of PC Cards and the abundance of PC Card I/O applications require substantial interrupt support from the PCI1520. The PCI1520 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 PCI1520 is, therefore, backward compatible with existing interrupt control register definitions, and new registers have been defined where required.

The PCI1520 detects PC Card interrupts and events at the PC Card interface and notifies the host controller using one of several interrupt signaling protocols. To simplify the discussion of interrupts in the PCI1520, PC Card interrupts are classified either as card status change (CSC) or as functional interrupts.

The method by which any type of PCI1520 interrupt is communicated to the host interrupt controller varies from system to system. The PCI1520 offers system designers the choice of using parallel PCI interrupt signaling, parallel ISA-type IRQ interrupt signaling, or the IRQSER serialized ISA and/or PCI interrupt protocol. It is possible to use the parallel PCI interrupts in combination with either parallel IRQs or serialized IRQs, as detailed in the sections that follow. All interrupt signaling is provided through the seven multifunction terminals, MFUNC0–MFUNC6.

3.7.1 PC Card Functional and Card Status Change Interrupts

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

Card status change (CSC)-type interrupts are defined as events at the PC Card interface that are detected by the PCI1520 and may warrant notification of host card and socket services software for service. CSC events include both card insertion and removal from PC Card sockets, as well as transitions of certain PC Card signals.

Table 3–9 summarizes the sources of PC Card interrupts and the type of card associated with them. CSC and functional interrupt sources are dependent on the type of card inserted in the PC Card socket. The three types of cards that can be inserted into any PC Card socket are:

- 16-bit memory card
- 16-bit I/O card
- CardBus cards

Table 3–9. 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

Functional interrupt events are valid only for 16-bit I/O and CardBus cards; that is, the functional interrupts are not valid for 16-bit memory cards. Furthermore, card insertion and removal-type CSC interrupts are independent of the card type.

Table 3–10. 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.
All PC Cards	Card insertion or removal	CSC	$\overline{\text{CD1}}$//$\overline{\text{CCD1}}$, $\overline{\text{CD2}}$//$\overline{\text{CCD2}}$	A transition on either $\overline{\text{CD1}}$//$\overline{\text{CCD1}}$ or $\overline{\text{CD2}}$//$\overline{\text{CCD2}}$ indicates an insertion or removal of a 16-bit or CardBus PC Card.
	Power cycle complete	CSC	N/A	An interrupt is generated when a PC Card power-up cycle has completed.

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

The *1997 PC Card Standard* describes the power-up sequence that must be followed by the PCI1520 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 PCI1520 interrupt scheme can be used to notify the host system (see Table 3–10), denoted by the power cycle complete event. This interrupt source is considered a PCI1520 internal event, because it depends on the completion of applying power to the socket rather than on a signal change at the PC Card interface.

3.7.2 Interrupt Masks and Flags

Host software may individually mask (or disable) most of the potential interrupt sources listed in Table 3–10 by setting the appropriate bits in the PCI1520. By individually masking the interrupt sources listed, software can control those events that cause a PCI1520 interrupt. Host software has some control over the system interrupt the PCI1520 asserts by programming the appropriate routing registers. The PCI1520 allows host software to route PC Card CSC and PC Card functional interrupts to separate system interrupts. Interrupt routing somewhat specific to the interrupt signaling method used is discussed in more detail in the following sections.

When an interrupt is signaled by the PCI1520, the interrupt service routine must determine which of the events listed in Table 3–9 caused the interrupt. Internal registers in the PCI1520 provide flags that report the source of an interrupt. By reading these status bits, the interrupt service routine can determine the action to be taken.

Table 3–9 details the registers and bits associated with masking and reporting potential interrupts. All interrupts can be masked except the functional PC Card interrupts, and an interrupt status flag is available for all types of interrupts.

Notice that there is not a mask bit to stop the PCI1520 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.

Table 3–9 lists the various methods of clearing the interrupt flag bits. The flag bits in the ExCA registers (16-bit PC Card-related interrupt flags) can be cleared using two different methods. One method is an explicit write of 1 to the flag bit to clear and the other is by reading the flag bit register. The selection of flag bit clearing methods is made by bit 2 (IFCMODE) in the ExCA global control register (ExCA offset 1Eh/5Eh/81Eh, see Section 5.20), 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 (see Section 6.1). Although some of the functionality is shared between the CardBus registers and the ExCA registers, software should not program the chip through both register sets when a CardBus card is functioning.

3.7.3 Using Parallel IRQ Interrupts

The seven multifunction terminals, MFUNC6–MFUNC0, implemented in the PCI1520 can be routed to obtain a subset of the ISA IRQs. The IRQ choices provide ultimate flexibility in PC Card host interruptions. To use the parallel ISA-type IRQ interrupt signaling, software must program the device control register (PCI offset 92h, see Section 4.33), to select the parallel IRQ signaling scheme. See Section 4.30, *Multifunction Routing Register*, for details on configuring the multifunction terminals.

A system using parallel IRQs requires (at a minimum) one PCI terminal, $\overline{\text{INTA}}$, to signal CSC events. This requirement is dictated by certain card and socket-services software. The $\overline{\text{INTA}}$ requirement calls for routing the MFUNC0 terminal for $\overline{\text{INTA}}$ signaling. The INTRTIE bit is used, in this case, to route socket B interrupt events to $\overline{\text{INTA}}$. This leaves (at a maximum) six different IRQs to support legacy 16-bit PC Card functions.

As an example, suppose the six IRQs used by legacy PC Card applications are IRQ3, IRQ4, IRQ5, IRQ10, IRQ11, and IRQ15. The multifunction routing register must be programmed to a value of 0FBA 5432h. This value routes the MFUNC0 terminal to $\overline{\text{INTA}}$ signaling and routes the remaining terminals as illustrated in Figure 3–14. Not shown is that $\overline{\text{INTA}}$ must also be routed to the programmable interrupt controller (PIC), or to some circuitry that provides parallel PCI interrupts to the host.

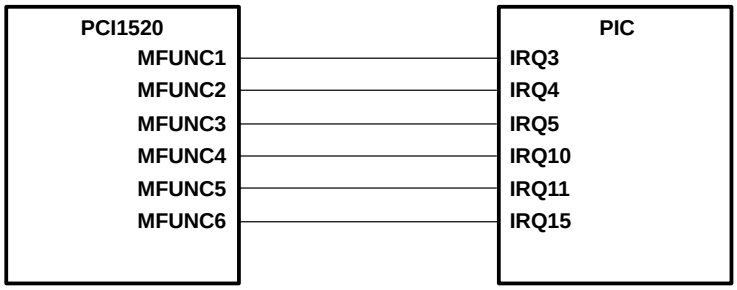


Figure 3–14. IRQ Implementation

Power-on software is responsible for programming the multifunction routing register to reflect the IRQ configuration of a system implementing the PCI1520. The multifunction routing register is shared between the two PCI1520 functions, and only one write to function 0 or 1 is necessary to configure the MFUNC6–MFUNC0 signals. Writing to function 0 only is recommended. See Section 4.30, *Multifunction Routing Register*, for details on configuring the multifunction terminals.

The parallel ISA-type IRQ signaling from the MFUNC6–MFUNC0 terminals is compatible with the input signal requirements of the 8259 PIC. The parallel IRQ option is provided for system designs that require legacy ISA IRQs. Design constraints may demand more MFUNC6–MFUNC0 IRQ terminals than the PCI1520 makes available.

3.7.4 Using Parallel PCI Interrupts

Parallel PCI interrupts are available when exclusively in parallel PCI interrupt/parallel ISA IRQ signaling mode, and when only IRQs are serialized with the IRQSER protocol. Both $\overline{\text{INTA}}$ and $\overline{\text{INTB}}$ can be routed to MFUNC terminals (MFUNC0 and MFUNC1). However, interrupts of both socket functions can be routed to $\overline{\text{INTA}}$ (MFUNC0) if bit 29 (INTRTIE) is set in the system control register (PCI offset 80h, see Section 4.29).

The INTRTIE bit affects the read-only value provided through accesses to the interrupt pin register (PCI offset 3Dh, see Section 4.24). When the INTRTIE bit is set, both functions return a value of 01h on reads from the interrupt pin register for both parallel and serial PCI interrupts. Table 3–11 summarizes the interrupt signaling modes.

Table 3–11. Interrupt Pin Register Cross Reference

INTRTIE BIT	INTPIN	
	FUNCTION 0	FUNCTION 1
0	01h	02h
1	01h	01h

3.7.5 Using Serialized IRQSER Interrupts

The serialized interrupt protocol implemented in the PCI1520 uses a single terminal to communicate all interrupt status information to the host controller. The protocol defines a serial packet consisting of a start cycle, multiple interrupt indication cycles, and a stop cycle. All data in the packet is synchronous with the PCI clock. The packet data describes 16 parallel ISA IRQ signals and the optional 4 PCI interrupts $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, $\overline{\text{INTC}}$, and $\overline{\text{INTD}}$. For details on the IRQSER protocol, refer to the document *Serialized IRQ Support for PCI Systems*.

3.7.6 SMI Support in the PCI1520

The PCI1520 provides a mechanism for interrupting the system when power changes have been made to the PC Card socket interfaces. The interrupt mechanism is designed to fit into a system maintenance interrupt (SMI) scheme. SMI interrupts are generated by the PCI1520, when enabled, after a write cycle to either the socket control register (CB offset 10h, see Section 6.5) of the CardBus register set, or the ExCA power control register (ExCA offset 02h/42h/802h, see Section 5.3) causes a power cycle change sequence to be sent on the power switch interface.

The SMI control is programmed through three bits in the system control register (PCI offset 80h, see Section 4.29). These bits are SMIROUTE (bit 26), SMISTATUS (bit 25), and SMIENB (bit 24). Table 3–12 describes the SMI control bits function.

Table 3–12. SMI Control

BIT NAME	FUNCTION
SMIROUTE	This shared bit controls whether the SMI interrupts are sent as a CSC interrupt or as IRQ2.
SMISTAT	This socket dependent bit is set when an SMI interrupt is pending. This status flag is cleared by writing back a 1.
SMIENB	When set, SMI interrupt generation is enabled. This bit is shared by functions 0 and 1.

If CSC SMI interrupts are selected, then the SMI interrupt is sent as the CSC on a per-socket basis. The CSC interrupt can be either level or edge mode, depending upon the CSCMODE bit in the ExCA global control register (ExCA offset 1Eh/5Eh/81Eh, see Section 5.20).

If IRQ2 is selected by SMIROUTE, then the IRQSER signaling protocol supports SMI signaling in the IRQ2 IRQ/Data slot. In a parallel ISA IRQ system, the support for an active low IRQ2 is provided only if IRQ2 is routed to either MFUNC3 or MFUNC6 through the multifunction routing register (PCI offset 8Ch, see Section 4.30).

3.8 Power Management Overview

In addition to the low-power CMOS technology process used for the PCI1520, various features are designed into the device to allow implementation of popular power-saving techniques. These features and techniques are discussed in this section.

3.8.1 Integrated Low-Dropout Voltage Regulator (LDO-VR)

The PCI1520 requires 2.5-V core voltage. The core power can be supplied by the PCI1520 itself using the internal LDO-VR. The core power can alternatively be supplied by an external power supply through the VR_PORT terminal. Table 3–13 lists the requirements for both the internal core power supply and the external core power supply.

Table 3–13. Requirements for Internal/External 2.5-V Core Power Supply

SUPPLY	V_{CC}	$\overline{VR\_EN}$	VR_PORT	NOTE
Internal	3.3 V	GND	2.5-V output	Internal 2.5-V LDO-VR is enabled. A 1.0 μF bypass capacitor is required on the VR_PORT terminal for decoupling. This output is not for external use.
External	3.3 V	V_{CC}	2.5-V input	Internal 2.5-V LDO-VR is disabled. An external 2.5-V power supply, of minimum 50-mA capacity, is required. A 0.1 μF bypass capacitor on the VR_PORT terminal is required.

3.8.2 Clock Run Protocol

The PCI $\overline{\text{CLKRUN}}$ feature is the primary method of power management on the PCI interface of the PCI1520. $\overline{\text{CLKRUN}}$ signaling is provided through the MFUNC6 terminal. Since some chip sets do not implement $\overline{\text{CLKRUN}}$, this is not always available to the system designer, and alternate power-saving features are provided. For details on the $\overline{\text{CLKRUN}}$ protocol see the *PCI Mobile Design Guide*.

The PCI1520 does not permit the central resource to stop the PCI clock under any of the following conditions:

- Bit 1 (KEEPCLK) in the system control register (PCI offset 80h, see Section 4.29) is set.
- The 16-bit PC Card- resource manager is busy.
- The PCI1520 CardBus master state machine is busy. A cycle may be in progress on CardBus.
- The PCI1520 master is busy. There may be posted data from CardBus to PCI in the PCI1520.
- Interrupts are pending.
- The CardBus CCLK for either socket has not been stopped by the PCI1520 $\overline{\text{CCLKRUN}}$ manager.

The PCI1520 restarts the PCI clock using the $\overline{\text{CLKRUN}}$ protocol under any of the following conditions:

- A 16-bit PC Card $\overline{\text{IREQ}}$ or a CardBus $\overline{\text{CINT}}$ has been asserted by either card.
- A CardBus CBWAKE (CSTSCHG) or 16-bit PC Card $\overline{\text{STSCHG}}$/$\overline{\text{RI}}$ event occurs in either socket.
- A CardBus attempts to start the CCLK using $\overline{\text{CCLKRUN}}$.
- A CardBus card arbitrates for the CardBus bus using $\overline{\text{CREQ}}$.

3.8.3 CardBus PC Card Power Management

The PCI1520 implements its own card power-management engine that can turn off the CCLK to a socket when there is no activity to the CardBus PC Card. The PCI clock-run protocol is followed on the CardBus $\overline{\text{CCLKRUN}}$ interface to control this clock management.

3.8.4 16-Bit PC Card Power Management

The COE bit (bit 7) of the ExCA power control register (ExCA offset 02h/42h/802h, see Section 5.3) and PWRDWN bit (bit 0) of the ExCA global control register (ExCA offset 1Eh/5Eh/81Eh, see Section 5.20) bits are provided for 16-bit PC Card power management. The COE bit places the card interface in a high-impedance state to save power. The power savings when using this feature are minimal. The COE bit resets the PC Card when used, and the PWRDWN bit does not. Furthermore, the PWRDWN bit is an automatic COE, that is, the PWRDWN performs the COE function when there is no card activity.

> **NOTE:** The 16-bit PC Card must implement the proper pullup resistors for the COE and PWRDWN modes.

3.8.5 Suspend Mode

The $\overline{\text{SUSPEND}}$ signal, provided for backward compatibility, gates the $\overline{\text{PRST}}$ (PCI reset) signal and the $\overline{\text{GRST}}$ (global reset) signal from the PCI1520. Besides gating $\overline{\text{PRST}}$ and $\overline{\text{GRST}}$, $\overline{\text{SUSPEND}}$ also gates PCLK inside the PCI1520 in order to minimize power consumption.

Gating PCLK does not create any issues with respect to the power switch interface in the PCI1520. This is because the PCI1520 does not depend on the PCI clock to clock the power switch interface. There are two methods to clock the power switch interface in the PCI1520:

- Use an external clock to the PCI1520 CLOCK terminal
- Use the internal oscillator

It should also be noted that asynchronous signals, such as card status change interrupts and $\overline{\text{RI\_OUT}}$, can be passed to the host system without a PCI clock. However, if card status change interrupts are routed over the serial interrupt stream, then the PCI clock must be restarted in order to pass the interrupt, because neither the internal oscillator nor an external clock is routed to the serial-interrupt state machine. Figure 3–15 is a signal diagram of the suspend function.

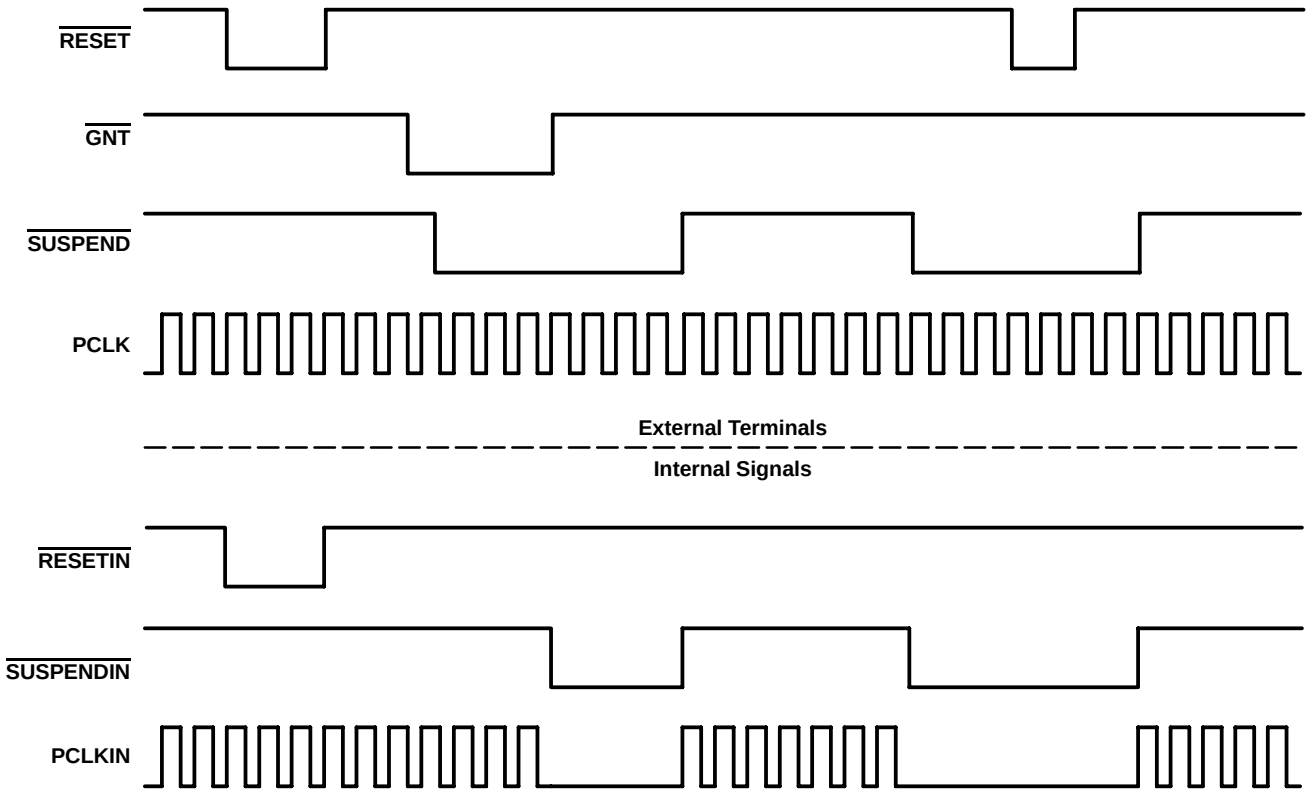


Figure 3–15. Signal Diagram of Suspend Function

3.8.6 Requirements for Suspend Mode

The suspend mode prevents the clearing of all register contents on the assertion of reset ($\overline{\text{PRST}}$ or $\overline{\text{GRST}}$) which would require the reconfiguration of the PCI1520 by software. Asserting the $\overline{\text{SUSPEND}}$ signal places the PCI outputs of the controller in a high-impedance state and gates the PCLK signal internally to the controller unless a PCI transaction is currently in process ($\overline{\text{GNT}}$ is asserted). It is important that the PCI bus not be parked on the PCI1520 when $\overline{\text{SUSPEND}}$ is asserted because the outputs are in a high-impedance state.

The GPIOs, MFUNC signals, and $\overline{\text{RI\_OUT}}$ signal are all active during $\overline{\text{SUSPEND}}$, unless they are disabled in the appropriate PCI1520 registers.

3.8.7 Ring Indicate

The $\overline{\text{RI\_OUT}}$ output is an important feature in power management, allowing a system to go into a suspended mode and wake up on modem rings and other card events. TI-designed flexibility permits this signal to fit wide platform requirements. $\overline{\text{RI\_OUT}}$ on the PCI1520 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 powered CardBus card asserts CSTSCHG from the insertion/removal of cards or change in battery voltage levels.

Figure 3–16 shows various enable bits for the PCI1520 $\overline{\text{RI\_OUT}}$ function; however, it does not show the masking of CSC events. See Table 3–9 for a detailed description of CSC interrupt masks and flags.

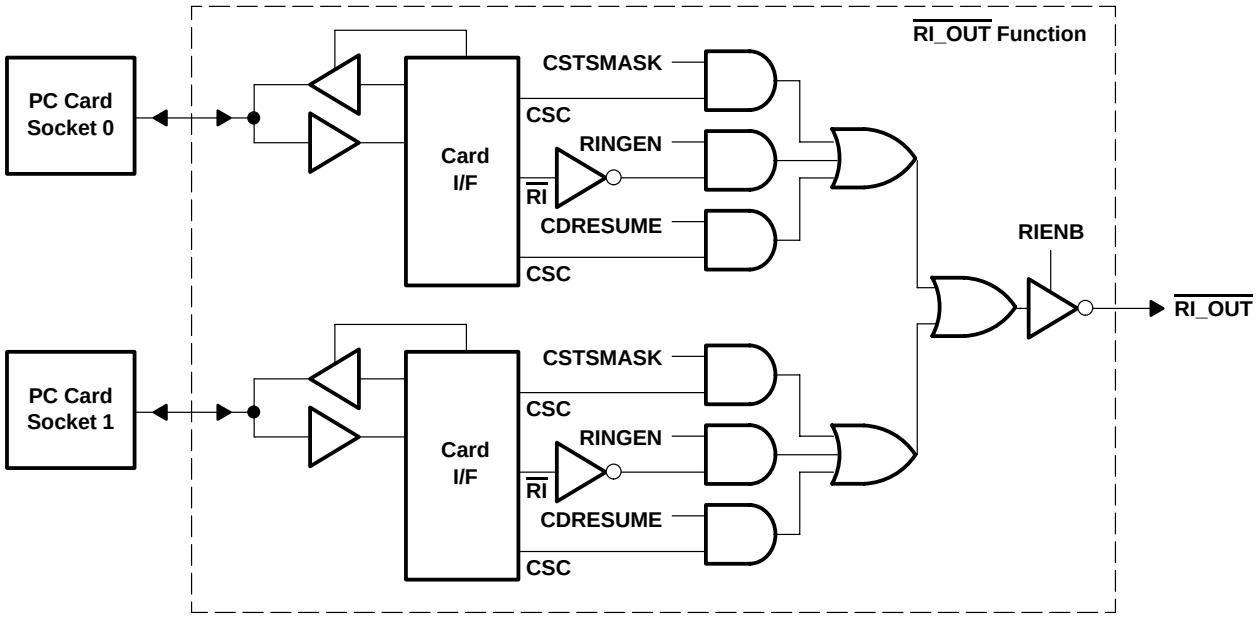


Figure 3–16. $\overline{\text{RI\_OUT}}$ Functional Diagram

$\overline{\text{RI}}$ from the 16-bit PC Card interface is masked by bit 7 (RINGEN) in the ExCA interrupt and general control register (ExCA offset 03h/43h/803h, see Section 5.4). 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 (bit 0, CSTSMASK) is programmed through the socket mask register (CB offset 04h, see Section 6.2) in the CardBus socket registers.

$\overline{\text{RI\_OUT}}$ can be routed through any of three different pins, $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$, MFUNC2, or MFUNC4. The $\overline{\text{RI\_OUT}}$ function is enabled by setting RIENB in the card control register (PCI offset 91h, see Section 4.32). The $\overline{\text{PME}}$ function is enabled by setting PMEEN in the power management control/status register (PCI offset A4h, see Section 4.38). When RIMUX in the system control register (PCI offset 80h, see Section 4.29) is set to 0, both the $\overline{\text{RI\_OUT}}$ function and the $\overline{\text{PME}}$ function are routed to the $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ terminal. If both functions are enabled and RIMUX is set to 0, the $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ terminal becomes $\overline{\text{RI\_OUT}}$ only and $\overline{\text{PME}}$ assertions will never be seen. Therefore, in a system using both the $\overline{\text{RI\_OUT}}$ function and the $\overline{\text{PME}}$ function, RIMUX must be set to 1 and $\overline{\text{RI\_OUT}}$ must be routed to either MFUNC2 or MFUNC4.

3.8.8 PCI Power Management

The *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges* 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 seven power-management states, resulting in varying levels of power savings.

The seven power-management states of PCI functions are:

- D0-uninitialized – Before device configuration, device not fully functional
- D0-active – Fully functional state
- D1 – Low-power state
- D2 – Low-power state
- $D3_{hot}$ – Low-power state. Transition state before $D3_{cold}$
- $D3_{cold}$ – $\overline{\text{PME}}$ signal-generation capable. Main power is removed and VAUX is available.
- $D3_{off}$ – No power and completely nonfunctional

NOTE 1: In the D0-uninitialized state, the PCI1520 does not generate $\overline{\text{PME}}$ and/or interrupts. When the IO_EN and MEM_EN bits (bits 0 and 1) of the command register (PCI offset 04h, see Section 4.4) are both set, the PCI1520 switches the state to D0-active. Transition from $D3_{cold}$ to the D0-uninitialized state happens at the deassertion of $\overline{\text{PRST}}$. The assertion of $\overline{\text{GRST}}$ forces the controller to the D0-uninitialized state immediately.

NOTE 2: The PWR_STATE bits (bits 0–1) of the power-management control/status register (PCI offset A4h, see Section 4.38) only code for four power states, D0, D1, D2, and $D3_{hot}$. The differences between the three D3 states is invisible to the software because the controller is not accessible in the $D3_{cold}$ or $D3_{off}$ state.

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

For the operating system (OS) to manage the device power states on the PCI bus, the PCI function should support four power-management operations. These operations are:

- Capabilities reporting
- Power status reporting
- Setting the power state
- System wake up

The OS identifies the capabilities of the PCI function by traversing the new capabilities list. The presence of capabilities in addition to the standard PCI capabilities is indicated by a 1 in bit 4 (CAPLIST) of the status register (PCI offset 06h, see Section 4.5).

The capabilities pointer provides access to the first item in the linked list of capabilities. For the PCI1520, 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, then the next item pointer must be set to 0. The registers following the next item pointer are specific to the capability of the function. The PCI power-management capability implements the register block outlined in Table 3–14.

Table 3–14. Power-Management Registers

REGISTER NAME				OFFSET
Power-management capabilities		Next item pointer	Capability ID	A0h
Data	Power-management control/ status register bridge support extensions	Power-management control/status (CSR)		A4h

The power management capabilities register (PCI offset A2h, see Section 4.37) provides information on the capabilities of the function related to power management. The power-management control/status register (PCI offset A4h, see Section 4.38) enables control of power-management states and enables/monitors power-management events. The data register is an optional register that can provide dynamic data.

For more information on PCI power management, see the *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges*.

3.8.9 CardBus Bridge Power Management

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

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

- Preservation of device context. The specification states that a reset must occur during the transition from D3 to D0. Some method to preserve wake-up context must be implemented so that the reset does not clear the $\overline{\text{PME}}$ context registers.
- Power source in $D3_{cold}$ if wake-up support is required from this state.

The Texas Instruments PCI1520 addresses these D3 wake-up issues in the following manner:

- Two resets are provided to handle preservation of $\overline{\text{PME}}$ context bits:
  - Global reset ($\overline{\text{GRST}}$) is used only on the initial boot up of the system after power up. It places the PCI1520 in its default state and requires BIOS to configure the device before becoming fully functional.
  - PCI reset ($\overline{\text{PRST}}$) has dual functionality based on whether $\overline{\text{PME}}$ is enabled or not. If $\overline{\text{PME}}$ is enabled, then $\overline{\text{PME}}$ context is preserved. If $\overline{\text{PME}}$ is not enabled, then $\overline{\text{PRST}}$ acts the same as a normal PCI reset. Please see the master list of $\overline{\text{PME}}$ context bits in Section 3.8.11.
- Power source in $D3_{cold}$ if wake-up support is required from this state. Since V_{CC} is removed in $D3_{cold}$, an auxiliary power source must be supplied to the PCI1520 V_{CC} terminals. Consult the *PCI14xx Implementation Guide for D3 Wake-Up* or the *PCI Power Management Interface Specification for PCI to CardBus Bridges* for further information.

3.8.10 ACPI Support

The *Advanced Configuration and Power Interface (ACPI) Specification* provides a mechanism that allows unique pieces of hardware to be described to the ACPI driver. The PCI1520 offers a generic interface that is compliant with ACPI design rules.

Two doublewords of general-purpose ACPI programming bits reside in PCI1520 PCI configuration space at offset A8h. The programming model is broken into status and control functions. In compliance with ACPI, the top level event status and enable bits reside in the general-purpose event status register (PCI offset A8h, see Section 4.41) and general-purpose event enable register (PCI offset AAh, see Section 4.42). The status and enable bits are implemented as defined by ACPI and illustrated in Figure 3–17.

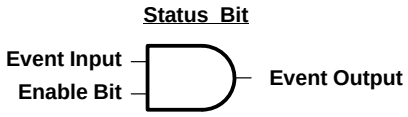


Figure 3–17. Block Diagram of a Status/Enable Cell

The status and enable bits generate an event that allows the ACPI driver to call a control method associated with the pending status bit. The control method can then control the hardware by manipulating the hardware control bits or by investigating child status bits and calling their respective control methods. A hierarchical implementation would be somewhat limiting, however, as upstream devices would have to remain in some level of power state to report events.

For more information of ACPI, see the *Advanced Configuration and Power Interface (ACPI) Specification.*

3.8.11 Master List of $\overline{\text{PME}}$ Context Bits and Global Reset-Only Bits

If the $\overline{\text{PME}}$ enable bit (bit 8) of the power-management control/status register (PCI offset A4h, see section 4.38) is asserted, then the assertion of $\overline{\text{PRST}}$ will not clear the following $\overline{\text{PME}}$ context bits. If the $\overline{\text{PME}}$ enable bit is not asserted, then the $\overline{\text{PME}}$ context bits are cleared with $\overline{\text{PRST}}$. The $\overline{\text{PME}}$ context bits are:

- Bridge control register (PCI offset 3Eh): bit 6
- System control register (PCI offset 80h): bits 10, 9, 8
- Power-management control/status register (PCI offset A4h): bits 15, 8
- ExCA power control register (ExCA offset 802h): bits 7, 5†, 4–3, 1–0 († 82365SL mode only)
- ExCA interrupt and general control register (ExCA offset 803h): bits 6–5
- ExCA card status change register (ExCA offset 804h): bits 11–8, 3–0
- ExCA card status-change-interrupt configuration register (ExCA offset 805h): bits 3–0
- CardBus socket event register (CardBus offset 00h): bits 3–0
- CardBus socket mask register (CardBus offset 04h): bits 3–0
- CardBus socket present state register (CardBus offset 08h): bits 13–7, 5–1
- CardBus socket control register (CardBus offset 10h): bits 6–4, 2–0

Global reset places all registers in their default state regardless of the state of the $\overline{\text{PME}}$ enable bit. The $\overline{\text{GRST}}$ signal is gated only by the $\overline{\text{SUSPEND}}$ signal. This means that assertion of $\overline{\text{SUSPEND}}$ blocks the $\overline{\text{GRST}}$ signal internally, thus preserving all register contents. The registers cleared only by $\overline{\text{GRST}}$ are:

- Status register (PCI offset 06h): bits 15–11, 8
- Secondary status register (PCI offset 16h): bits 15–11, 8
- Interrupt pin register (PCI offset 3Dh): bits 1,0 (function 1 only)
- Subsystem vendor ID register (PCI offset 40h): bits 15–0
- Subsystem ID register (PCI offset 42h): bits 15–0
- PC Card 16-bit legacy mode base address register (PCI offset 44h): bits 31–1
- System control register (PCI offset 80h): bits 31–29, 27–13, 11, 6–0
- Multifunction routing register (PCI offset 8Ch): bits 27–0
- Retry status register (PCI offset 90h): bits 7–5, 3, 1
- Card control register (PCI offset 91h): bits 7–5, 2–0
- Device control register (PCI offset 92h): bits 7–5, 3–0
- Diagnostic register (PCI offset 93h): bits 7–0
- Power management capabilities register (PCI offset A2h): bit 15
- General-purpose event status register (PCI offset A8h): bits 15–14
- General-purpose event enable register (PCI offset AAh): bits 15–14, 11, 8, 4–0
- General-purpose output (PCI offset AEh): bits 4–0
- Serial bus data (PCI offset B0h): bits 7–0
- Serial bus index (PCI offset B1h): bits 7–0
- Serial bus slave address register (PCI offset B2h): bits 7–0
- Serial bus control and status register (PCI offset B3h): bits 7, 5–0
- ExCA identification and revision register (ExCA offset 00h): bits 7–0
- ExCA global control register (ExCA offset 1Eh): bits 2–0
- Socket present state register (CardBus offset 08h): bit 29
- Socket power management register (CardBus offset 20h): bits 25–24

4 PC Card Controller Programming Model

This section describes the PCI1520 PCI configuration registers that make up the 256-byte PCI configuration header for each PCI1520 function. As noted, some bits are global in nature and are accessed only through function 0.

4.1 PCI Configuration Registers (Functions 0 and 1)

The PCI1520 is a multifunction PCI device, and the PC Card controller is integrated as PCI functions 0 and 1. The configuration header is compliant with the *PCI Local Bus Specification* as a CardBus bridge header and is *PC 99* compliant as well. Table 4–1 shows the PCI configuration header, which includes both the predefined portion of the configuration space and the user-definable registers.

Table 4–1. PCI Configuration Registers (Functions 0 and 1)

REGISTER NAME				OFFSET
Device ID		Vendor ID		00h
Status		Command		04h
Class code			Revision ID	08h
BIST	Header type	Latency timer	Cache line size	0Ch
CardBus socket/ExCA base address				10h
Secondary status		Reserved	Capability pointer	14h
CardBus latency timer	Subordinate bus number	CardBus bus number	PCI bus number	18h
CardBus Memory base register 0				1Ch
CardBus Memory limit register 0				20h
CardBus Memory base register 1				24h
CardBus Memory limit register 1				28h
CardBus I/O base register 0				2Ch
CardBus I/O limit register 0				30h
CardBus I/O base register 1				34h
CardBus I/O limit register 1				38h
Bridge control		Interrupt pin	Interrupt line	3Ch
Subsystem ID		Subsystem vendor ID		40h
PC Card 16-bit I/F legacy-mode base address				44h
Reserved				48h–7Ch
System control				80h
Reserved				84h–88h
Multifunction routing				8Ch
Diagnostic	Device control	Card control	Retry status	90h
Reserved				94h–9Ch
Power-management capabilities		Next-item pointer	Capability ID	A0h
Power-management data	Power-management control/status bridge support extensions	Power-management control/status		A4h
General-purpose event enable		General-purpose event status		A8h
General-purpose output		General-purpose input		ACh
Serial bus control/status	Serial bus slave address	Serial bus index	Serial bus data	B0h
Reserved				B4h–FCh

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

Table 4–2. Bit Field Access Tag Descriptions

ACCESS TAG	NAME	MEANING
R	Read	Field may be read by software.
W	Write	Field may be written by software to any value.
S	Set	Field may be set by a write of 1. Writes of 0 have no effect.
C	Clear	Field may be cleared by a write of 1. Writes of 0 have no effect.
U	Update	Field may be autonomously updated by the PCI1520.

4.2 Vendor ID Register

This 16-bit register contains a value allocated by the PCI Special Interest Group (SIG) and identifies the manufacturer of the PCI device. The vendor ID assigned to TI is 104Ch.

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

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

4.3 Device ID Register

This 16-bit register contains a value assigned to the PCI1520 by TI. The device identification for the PCI1520 is AC55h.

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

Register: **Device ID**
Offset: 02h (functions 0, 1)
Type: Read-only
Default: AC55h

4.4 Command Register

The command register provides control over the PCI1520 interface to the PCI bus. All bit functions adhere to the definitions in *PCI Local Bus Specification*. None of the bit functions in this register is shared between the two PCI1520 PCI functions. Two command registers exist in the PCI1520, one for each function. Software must manipulate the two PCI1520 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. See Table 4–3 for a complete description of the register contents.

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

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

Table 4–3. Command Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–10	RSVD	R	Reserved. Bits 15–10 return 0s when read.
9	FBB_EN	R	Fast back-to-back enable. The PCI1520 does not generate fast back-to-back transactions; therefore, bit 9 returns 0 when read.
8	SERR_EN	RW	System error ($\overline{\text{SERR}}$) enable. Bit 8 controls the enable for the $\overline{\text{SERR}}$ driver on the PCI interface. $\overline{\text{SERR}}$ can be asserted after detecting an address parity error on the PCI bus. Both bits 8 and 6 must be set for the PCI1520 to report address parity errors. 0 = Disable $\overline{\text{SERR}}$ output driver (default) 1 = Enable $\overline{\text{SERR}}$ output driver
7	STEP_EN	R	Address/data stepping control. The PCI1520 does not support address/data stepping; therefore, bit 7 is hardwired to 0.
6	PERR_EN	RW	Parity error response enable. Bit 6 controls the PCI1520 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 = PCI1520 ignores detected parity error (default) 1 = PCI1520 responds to detected parity errors
5	VGA_EN	RW	VGA palette snoop. Bit 5 controls how PCI devices handle accesses to video graphics array (VGA) palette registers.
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 PCI1520 controller does not support memory write-and-invalidate commands, but uses memory write commands instead; therefore, this bit is hardwired to 0.
3	SPECIAL	R	Special cycles. Bit 3 controls whether or not a PCI device ignores PCI special cycles. The PCI1520 does not respond to special cycle operations; therefore, this bit is hardwired to 0.
2	MAST_EN	RW	Bus master control. Bit 2 controls whether or not the PCI1520 can act as a PCI bus initiator (master). The PCI1520 can take control of the PCI bus only when this bit is set. 0 = Disables the PCI1520 from generating PCI bus accesses (default) 1 = Enables the PCI1520 to generate PCI bus accesses
1	MEM_EN	RW	Memory space enable. Bit 1 controls whether or not the PCI1520 can claim cycles in PCI memory space. 0 = Disables the PCI1520 from responding to memory space accesses (default) 1 = Enables the PCI1520 to respond to memory space accesses
0	IO_EN	RW	I/O space control. Bit 0 controls whether or not the PCI1520 can claim cycles in PCI I/O space. 0 = Disables the PCI1520 from responding to I/O space accesses (default) 1 = Enables the PCI1520 to respond to I/O space accesses

4.5 Status Register

The status register provides device information to the host system. Bits in this register can be read normally. A bit in the status register is reset when a 1 is written to that bit location; a 0 written to a bit location has no effect. All bit functions adhere to the definitions in the *PCI Local Bus Specification*. PCI bus status is shown through each function. See Table 4–4 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Status															
Type	RC	RC	RC	RC	RC	R	R	RC	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**
Offset: 06h (functions 0, 1)
Type: Read-only, Read/Clear
Default: 0210h

Table 4–4. Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	PAR_ERR	RC	Detected parity error. Bit 15 is set when a parity error is detected (either address or data).
14	SYS_ERR	RC	Signaled system error. Bit 14 is set when $\overline{\text{SERR}}$ is enabled and the PCI1520 signals a system error to the host.
13	MABORT	RC	Received master abort. Bit 13 is set when a cycle initiated by the PCI1520 on the PCI bus is terminated by a master abort.
12	TABT_REC	RC	Received target abort. Bit 12 is set when a cycle initiated by the PCI1520 on the PCI bus is terminated by a target abort.
11	TABT_SIG	RC	Signaled target abort. Bit 11 is set by the PCI1520 when it terminates a transaction on the PCI bus with a target abort.
10–9	PCI_SPEED	R	$\overline{\text{DEVSEL}}$ timing. These bits encode the timing of $\overline{\text{DEVSEL}}$ and are hardwired 01b, indicating that the PCI1520 asserts PCI_SPEED at a medium speed on nonconfiguration cycle accesses.
8	DATAPAR	RC	Data parity error detected. 0 = The conditions for setting bit 8 have not been met. 1 = A data parity error occurred, and the following conditions were met: a. $\overline{\text{PERR}}$ was asserted by any PCI device including the PCI1520. b. The PCI1520 was the bus master during the data parity error. c. The parity error response bit is set in the command register (PCI offset 04h, see Section 4.4).
7	FBB_CAP	R	Fast back-to-back capable. The PCI1520 cannot accept fast back-to-back transactions; therefore, bit 7 is hardwired to 0.
6	UDF	R	User-definable feature support. The PCI1520 does not support the user-definable features; therefore, bit 6 is hardwired to 0.
5	66MHZ	R	66-MHz capable. The PCI1520 operates at a maximum PCLK frequency of 33 MHz; therefore, bit 5 is hardwired to 0.
4	CAPLIST	R	Capabilities list. Bit 4 returns 1 when read. This bit indicates that capabilities in addition to standard PCI capabilities are implemented. The linked list of PCI power-management capabilities is implemented in this function.
3–0	RSVD	R	Reserved. Bits 3–0 return 0s when read.

4.6 Revision ID Register

The revision ID register indicates the silicon revision of the PCI1520.

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**
Offset: 08h (functions 0, 1)
Type: Read-only
Default: 01h

4.7 PCI Class Code Register

The class code register recognizes PCI1520 functions 0 and 1 as a bridge device (06h) and a CardBus bridge device (07h), with a 00h programming interface.

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

Register: **PCI class code**
Offset: 09h (functions 0, 1)
Type: Read-only
Default: 06 0700h

4.8 Cache Line Size Register

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

Bit	7	6	5	4	3	2	1	0
Name	Cache line size							
Type	RW	RW	RW	RW	RW	RW	RW	RW
Default	0	0	0	0	0	0	0	0

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

4.9 Latency Timer Register

The latency timer register specifies the latency time for the PCI1520 in units of PCI clock cycles. When the PCI1520 is a PCI bus initiator and asserts $\overline{\text{FRAME}}$, the latency timer begins counting from zero. If the latency timer expires before the PCI1520 transaction has terminated, then the PCI1520 terminates the transaction when its $\overline{\text{GNT}}$ is deasserted. This register is separate for each of the two PCI1520 functions. This allows platforms to prioritize use of the PCI bus by the two PCI1520 functions.

Bit	7	6	5	4	3	2	1	0
Name	Latency timer							
Type	RW	RW	RW	RW	RW	RW	RW	RW
Default	0	0	0	0	0	0	0	0

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

4.10 Header Type Register

This register returns 82h when read, indicating that the PCI1520 function 0 and 1 configuration spaces adhere to the CardBus bridge PCI header. The CardBus bridge PCI header ranges from PCI register 000h to 7Fh, and 80h to FFh is user-definable extension registers.

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

Register: **Header type**
Offset: 0Eh (functions 0, 1)
Type: Read/Write
Default: 82h

4.11 BIST Register

Because the PCI1520 does not support a built-in self-test (BIST), this register returns the value of 00h when read.

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

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

4.12 CardBus Socket/ExCA Base-Address Register

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

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

4.13 Capability Pointer Register

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

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

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

4.14 Secondary Status Register

The secondary status register is compatible with the PCI-to-PCI bridge secondary status register and indicates CardBus-related device information to the host system. This register is very similar to the status register (offset 06h, see Section 4.5); status bits are cleared by writing a 1. See Table 4–5 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Secondary status															
Type	RC	RC	RC	RC	RC	R	R	RC	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**
Offset: 16h
Type: Read-only, Read/Clear
Default: 0200h

Table 4–5. Secondary Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	CBPARITY	RC	Detected parity error. Bit 15 is set when a CardBus parity error is detected (either address or data).
14	CBSERR	RC	Signaled system error. Bit 14 is set when $\overline{\text{CSERR}}$ is signaled by a CardBus card. The PCI1520 does not assert $\overline{\text{CSERR}}$.
13	CBMABORT	RC	Received master abort. Bit 13 is set when a cycle initiated by the PCI1520 on the CardBus bus has been terminated by a master abort.
12	REC_CBTA	RC	Received target abort. Bit 12 is set when a cycle initiated by the PCI1520 on the CardBus bus is terminated by a target abort.
11	SIG_CBTA	RC	Signaled target abort. Bit 11 is set by the PCI1520 when it terminates a transaction on the CardBus bus with a target abort.
10–9	CB_SPEED	R	CDEVSEL timing. These bits encode the timing of $\overline{\text{CDEVSEL}}$ and are hardwired 01b, indicating that the PCI1520 asserts CB_SPEED at a medium speed.
8	CB_DPAR	RC	CardBus data parity error detected. 0 = The conditions for setting bit 8 have not been met. 1 = A data parity error occurred and the following conditions were met: a. $\overline{\text{CPERR}}$ was asserted on the CardBus interface. b. The PCI1520 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 PCI1520 cannot accept fast back-to-back transactions; therefore, bit 7 is hardwired to 0.
6	CB_UDF	R	User-definable feature support. The PCI1520 does not support user-definable features; therefore, bit 6 is hardwired to 0.
5	CB66MHZ	R	66-MHz capable. The PCI1520 CardBus interface operates at a maximum CCLK frequency of 33 MHz; therefore, bit 5 is hardwired to 0.
4–0	RSVD	R	Reserved. Bits 4–0 return 0s when read.

4.15 PCI Bus Number Register

This register is programmed by the host system to indicate the bus number of the PCI bus to which the PCI1520 is connected. The PCI1520 uses this register in conjunction with the CardBus bus number and subordinate bus number registers to determine when to forward PCI configuration cycles to its secondary buses.

Bit	7	6	5	4	3	2	1	0
Name	PCI bus number							
Type	RW	RW	RW	RW	RW	RW	RW	RW
Default	0	0	0	0	0	0	0	0

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

4.16 CardBus Bus Number Register

This register is programmed by the host system to indicate the bus number of the CardBus bus to which the PCI1520 is connected. The PCI1520 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 PCI1520 controller function.

Bit	7	6	5	4	3	2	1	0
Name	CardBus bus number							
Type	RW	RW	RW	RW	RW	RW	RW	RW
Default	0	0	0	0	0	0	0	0

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

4.17 Subordinate Bus Number Register

This register is programmed by the host system to indicate the highest-numbered bus below the CardBus bus. The PCI1520 uses this register in conjunction with the PCI bus number and CardBus bus number registers to determine when to forward PCI configuration cycles to its secondary buses. This register is separate for each CardBus controller function.

Bit	7	6	5	4	3	2	1	0
Name	Subordinate bus number							
Type	RW	RW	RW	RW	RW	RW	RW	RW
Default	0	0	0	0	0	0	0	0

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

4.18 CardBus Latency Timer Register

This register is programmed by the host system to specify the latency timer for the PCI1520 CardBus interface in units of CCLK cycles. When the PCI1520 is a CardBus initiator and asserts $\overline{\text{CFRAME}}$, the CardBus latency timer begins counting. If the latency timer expires before the PCI1520 transaction has terminated, then the PCI1520 terminates the transaction at the end of the next data phase. A recommended minimum value for this register is 40h, which allows most transactions to be completed.

Bit	7	6	5	4	3	2	1	0
Name	CardBus latency timer							
Type	RW	RW	RW	RW	RW	RW	RW	RW
Default	0	0	0	0	0	0	0	0

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

4.19 Memory Base Registers 0, 1

The memory base registers indicate the lower address of a PCI memory address range. These registers are used by the PCI1520 to determine when to forward a memory transaction to the CardBus bus and when to forward a CardBus cycle to PCI. Bits 31–12 of these registers are read/write and allow the memory base to be located anywhere in the 32-bit PCI memory space on 4-Kbyte boundaries. Bits 11–0 are read-only and always return 0s. Write transactions to these bits have no effect. Bits 8 and 9 of the bridge control register 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 PCI1520 to claim any memory transactions through CardBus memory windows (that is, these windows are not enabled by default to pass the first 4 Kbytes of memory to CardBus).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Memory base registers 0, 1															
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
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	RW	RW	RW	RW	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**
Offset: 1Ch, 24h
Type: Read-only, Read/Write
Default: 0000 0000h

4.20 Memory Limit Registers 0, 1

The memory limit registers indicate the upper address of a PCI memory address range. These registers are used by the PCI1520 to determine when to forward a memory transaction to the CardBus bus and when to forward a CardBus cycle to PCI. Bits 31–12 of these registers are read/write and allow the memory base to be located anywhere in the 32-bit PCI memory space on 4-Kbyte boundaries. Bits 11–0 are read-only and always return 0s. Write transactions to these bits have no effect. Bits 8 and 9 of the bridge control register 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 PCI1520 to claim any memory transactions through CardBus memory windows; that is, these windows are not enabled by default to pass the first 4 Kbytes of memory to CardBus.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Memory limit registers 0, 1															
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
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	RW	RW	RW	RW	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**
Offset: 20h, 28h
Type: Read-only, Read/Write
Default: 0000 0000h

4.21 I/O Base Registers 0, 1

The I/O base registers indicate the lower address of a PCI I/O address range. These registers are used by the PCI1520 to determine when to forward an I/O transaction to the CardBus bus and when to forward a CardBus cycle to the PCI bus. The lower 16 bits of this register locate the bottom of the I/O window within a 64-Kbyte page, and the upper 16 bits (31–16) are a page register which locates this 64-Kbyte page in 32-bit PCI I/O address space. Bits 31–2 are read/write. Bits 1 and 0 are read-only and always return 0s, forcing I/O windows to be aligned on a natural doubleword boundary.

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

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	I/O base registers 0, 1															
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
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	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	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**
Offset: 2Ch, 34h
Type: Read-only, Read/Write
Default: 0000 0000h

4.22 I/O Limit Registers 0, 1

The I/O limit registers indicate the upper address of a PCI I/O address range. These registers are used by the PCI1520 to determine when to forward an I/O transaction to the CardBus bus and when to forward a CardBus cycle to PCI. The lower 16 bits of this register locate the top of the I/O window within a 64-Kbyte page, and the upper 16 bits are a page register that locates this 64-Kbyte page in 32-bit PCI I/O address space. Bits 15–2 are read/write and allow the I/O limit address to be located anywhere in the 64-Kbyte page (indicated by bits 31–16 of the appropriate I/O base) on doubleword boundaries.

Bits 31–16 are read-only and always return 0s when read. The page is set in the I/O base register. Bits 1 and 0 are read-only and always return 0s, forcing I/O windows to be aligned on a natural doubleword boundary. Write transactions to read-only bits have no effect. The PCI1520 assumes that the lower 2 bits of the limit address are 1s.

NOTE: The I/O base or the I/O limit register must be nonzero to enable an I/O transaction.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	I/O limit registers 0, 1															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	I/O limit registers 0, 1															
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	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**
Offset: 30h, 38h
Type: Read-only, Read/Write
Default: 0000 0000h

4.23 Interrupt Line Register

The interrupt line register communicates interrupt line routing information. Each PCI1520 function has an interrupt line register.

Bit	7	6	5	4	3	2	1	0
Name	Interrupt line							
Type	RW	RW	RW	RW	RW	RW	RW	RW
Default	1	1	1	1	1	1	1	1

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

4.24 Interrupt Pin Register

The value read from the interrupt pin register is function dependent and depends on the interrupt signaling mode, selected through bits 2–1 (INTMODE field) of the device control register (PCI offset 92h, see Section 4.33) and the state of bit 29 (INTRTIE) in the system control register (PCI offset 80h, see Section 4.29). When the INTRTIE bit is set, this register reads 01h ($\overline{\text{INTA}}$) for both functions. See Table 4–6 for a complete description of the register contents.

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

Register: **Interrupt pin**
Offset: 3Dh
Type: Read-only
Default: 0Xh

Table 4–6. Interrupt Pin Register Cross Reference

INTRTIE BIT	INTPIN	
	FUNCTION 0	FUNCTION 1
0	01h	02h
1	01h	01h

4.25 Bridge Control Register

The bridge control register provides control over various PCI1520 bridging functions. Some bits in this register are global and are accessed only through function 0. See Table 4–7 for a complete description of the register contents.

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

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

Table 4–7. Bridge Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–11	RSVD	R	Reserved. Bits 15–11 return 0s when read.
10	POSTEN	RW	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 burst 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	RW	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	RW	Memory window 0 type. Bit 8 specifies whether or not memory window 0 is prefetchable. This bit is encoded as: 0 = Memory window 0 is nonprefetchable. 1 = Memory window 0 is prefetchable (default).
7	INTR	RW	PCI interrupt – IREQ routing enable. Bit 7 selects 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 ExCAs
6	CRST	RW	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	RW	Master abort mode. Bit 5 controls how the PCI1520 responds to a master abort when the PCI1520 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	RW	VGA enable. Bit 3 affects how the PCI1520 responds to VGA addresses. When this bit is set, accesses to VGA addresses are forwarded.
2	ISAEN	RW	ISA mode enable. Bit 2 affects how the PCI1520 passes I/O cycles within the 64-Kbyte ISA range. This bit is not common between sockets. When this bit is set, the PCI1520 does not forward the last 768 bytes of each 1K I/O range to CardBus.
1†	CSERREN	RW	$\overline{\text{CSERR}}$ enable. Bit 1 controls the response of the PCI1520 to $\overline{\text{CSERR}}$ signals on the CardBus bus. This bit is common between the two sockets. 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	RW	CardBus parity error response enable. Bit 0 controls the response of the PCI1520 to CardBus parity errors. This bit is common between the two sockets. 0 = CardBus parity errors are ignored. 1 = CardBus parity errors are reported using $\overline{\text{CPERR}}$.

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

4.26 Subsystem Vendor ID Register

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

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

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

4.27 Subsystem ID Register

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

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

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

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

The PCI1520 supports the index/data scheme of accessing the ExCA registers, which are 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. See Section 5, *ExCA Compatibility Registers*, for register offsets.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PC Card 16-bit I/F legacy-mode base address															
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
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	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	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**
Offset: 44h (functions 0, 1)
Type: Read-only, Read/Write
Default: 0000 0001h

4.29 System Control Register

System-level initializations are performed by programming this doubleword register. Some of the bits are global and are written only through function 0. See Table 4–8 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	System control															
Type	RW	RW	RW	R	RW	RW	RC	RW	R	RW	RW	RW	RW	RW	RW	RW
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	RW	RW	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW
Default	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0

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

Table 4–8. System Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–30†	SER_STEP	RW	Serialized PCI interrupt routing step. Bits 31 and 30 configure the serialized PCI interrupt stream signaling and accomplish an even distribution of interrupts signaled on the four PCI interrupt slots. Bits 31 and 30 are global to all PCI1520 functions. 00 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ signal in $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ IRQSER slots 01 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ signal in $\overline{\text{INTB}}$/$\overline{\text{INTC}}$ IRQSER slots 10 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ signal in $\overline{\text{INTC}}$/$\overline{\text{INTD}}$ IRQSER slots 11 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ signal in $\overline{\text{INTD}}$/$\overline{\text{INTA}}$ IRQSER slots
29†	INTRTIE	RW	Tie internal PCI interrupts. When this bit is set, the $\overline{\text{INTA}}$ and $\overline{\text{INTB}}$ signals are tied together internally and are signaled as $\overline{\text{INTA}}$. $\overline{\text{INTA}}$ can then be shifted by using bits 31–30 (SER_STEP). This bit is global to all PCI1520 functions. When configuring the PCI1520 functions to share PCI interrupts, multifunction terminal MFUNC3 must be configured as IRQSER prior to setting the INTRTIE bit.
28	RSVD	R	Reserved. Bit 28 returns 0 when read.
27†	P2CCLK	RW	P2C power switch clock. The PCI1520 CLOCK signal is used to clock the serial interface power switch and the internal state machine. The default state for bit 27 is 0, requiring an external clock source provided to the CLOCK terminal (terminal number F15 for the GHK package or terminal number 154 for the PDV package). Bit 27 can be set to 1, allowing the internal oscillator to provide the clock signal. 0 = CLOCK provided externally, input to PCI1520 (default) 1 = CLOCK generated by internal oscillator and driven by PCI1520.
26	SMIROUTE	RW	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	RC	SMI interrupt status. This socket-dependent bit is set when bit 24 (SMIENB) is set and a write occurs to set the socket power. Writing a 1 to bit 25 clears the status. 0 = SMI interrupt signaled (default) 1 = SMI interrupt not signaled
24†	SMIENB	RW	SMI interrupt mode enable. When bit 24 is set and a write to the socket power control occurs, the SMI interrupt signaling is enabled and generates an interrupt. This bit is shared and defaults to 0 (disabled).
23	RSVD	R	Reserved. Bit 23 returns 0 when read.
22	CBRSVD	RW	CardBus reserved terminals signaling. When a CardBus card is inserted and bit 22 is set, the RSVD CardBus terminals are driven low. When this bit is 0, these terminals are placed in a high-impedance state. 0 = Place CardBus RSVD terminals in a high-impedance state. 1 = Drive Cardbus RSVD terminals low (default).
21	VCCPROT	RW	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	REDUCEZV	RW	Reduced zoomed video enable. When this bit is enabled, terminals A25–A22 of the card interface for PC Card-16 cards are placed in the high-impedance state. This bit should not be set for normal ZV operation. This bit is encoded as: 0 = Reduced zoomed video disabled (default) 1 = Reduced zoomed video enabled
19–16	RSVD	RW	Reserved. Do not change the default value.
15†	MRBURSTDN	RW	Memory read burst enable downstream. When bit 15 is set, memory read transactions are allowed to burst downstream. 0 = Downstream memory read burst is disabled. 1 = Downstream memory read burst is enabled (default).
14†	MRBURSTUP	RW	Memory read burst enable upstream. When bit 14 is set, the PCI1520 allows memory read transactions to burst upstream. 0 = Upstream memory read burst is disabled (default). 1 = Upstream memory read burst is enabled.
13	SOCACTIVE	R	Socket activity status. When set, bit 13 indicates access has been performed to or from a PC card and is cleared upon read of this status bit. This bit is socket-dependent. 0 = No socket activity (default) 1 = Socket activity
12	RSVD	R	Reserved. Bit 12 returns 1 when read.

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

Table 4–8. System Control Register Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION
11†	PWRSTREAM	R	Power stream in progress status bit. When set, bit 11 indicates that a power stream to the power switch is in progress and a powering change has been requested. This bit is cleared when the power stream is complete. 0 = Power stream is complete and delay has expired. 1 = Power stream is in progress.
10†	DELAYUP	R	Power-up delay in progress status. When set, bit 10 indicates that a power-up stream has been sent to the power switch and proper power may not yet be stable. This bit is cleared when the power-up delay has expired.
9†	DELAYDOWN	R	Power-down delay in progress status. When set, bit 9 indicates that a power-down stream has been sent to the power switch and proper power may not yet be stable. This bit is cleared when the power-down delay has expired.
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 returns 0 when read.
6	PWRSAVINGS	RW	Power savings mode enable. When this bit is set, if a CB card is inserted, idle, and without a CB clock, then the applicable CB state machine will not be clocked.
5†	SUBSYSRW	RW	Subsystem ID (PCI offset 42h, see Section 4.27), subsystem vendor ID (PCI offset 40H, see Section 4.26), ExCA identification and revision (ExCA offset 00h/40h/800h, see Section 5.1) registers read/write enable. Bit 5 is shared by functions 0 and 1. 0 = Subsystem ID, subsystem vendor ID, ExCA identification and revision registers are read/write. 1 = Subsystem ID, subsystem vendor ID, ExCA identification and revision registers are read-only (default).
4†	CB_DPAR	RW	CardBus data parity $\overline{\text{SERR}}$ signaling enable 0 = CardBus data parity not signaled on PCI $\overline{\text{SERR}}$ 1 = CardBus data parity signaled on PCI $\overline{\text{SERR}}$
3	RSVD	RW	Reserved. Do not change the default value.
2	EXCAPOWER	RW	ExCA power-control bit. 0 = Enables 3.3 V 1 = Enables 5 V
1†	KEEPCLK	RW	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	RW	$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ multiplex enable. 0 = $\overline{\text{RI\_OUT}}$ and $\overline{\text{PME}}$ are both routed to the $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ terminal. If both functions are are enabled at the same time, the terminal becomes $\overline{\text{RI\_OUT}}$ only and $\overline{\text{PME}}$ assertions are not seen. 1 = Only $\overline{\text{PME}}$ is routed to the $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ terminal.

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

4.30 Multifunction Routing Register

The multifunction routing register is used to configure the MFUNC0–MFUNC6 terminals. These terminals may be configured for various functions. All multifunction terminals default to the general-purpose input configuration. This register is intended to be programmed once at power-on initialization. The default value for this register can also be loaded through a serial bus EEPROM. See Table 4–9 for a complete description of the register contents.

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

Register: **Multifunction routing**
Offset: 8Ch (functions 0, 1)
Type: Read-only, Read/Write
Default: 0000 1000h

Table 4–9. Multifunction Routing Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–28	RSVD	R	Bits 31–28 return 0s when read.
27–24	MFUNC6	RW	Multifunction terminal 6 configuration. These bits control the internal signal mapped to the MFUNC6 terminal as follows: 0000 = RSVD† 0100 = IRQ4 1000 = IRQ8 1100 = IRQ12 0001 = $\overline{\text{CLKRUN}}$ 0101 = IRQ5 1001 = IRQ9 1101 = IRQ13 0010 = IRQ2 0110 = IRQ6 1010 = IRQ10 1110 = IRQ14 0011 = IRQ3 0111 = IRQ7 1011 = IRQ11 1111 = IRQ15
23–20	MFUNC5	RW	Multifunction terminal 5 configuration. These bits control the internal signal mapped to the MFUNC5 terminal as follows: 0000 = GPI4† 0100 = IRQ4 1000 = $\overline{\text{CAUDPWM}}$ 1100 = LEDA1 0001 = $\overline{\text{GPO4}}$ 0101 = IRQ5 1001 = $\overline{\text{D3\_STAT}}$ 1101 = $\overline{\text{LED\_SKT}}$ 0010 = $\overline{\text{PCGNT}}$ 0110 = $\overline{\text{ZVSTAT}}$ 1010 = IRQ10 1110 = $\overline{\text{GPE}}$ 0011 = IRQ3 0111 = $\overline{\text{ZVSEL1}}$ 1011 = IRQ11 1111 = IRQ15
19–16	MFUNC4	RW	Multifunction terminal 4 configuration. These bits control the internal signal mapped to the MFUNC4 terminal as follows: NOTE: When the serial bus mode is implemented by pulling down the LATCH terminal, the MFUNC4 terminal provides the SCL signaling. 0000 = GPI3† 0100 = IRQ4 1000 = CAUDPWM 1100 = $\overline{\text{RI\_OUT}}$ 0001 = $\overline{\text{GPO3}}$ 0101 = IRQ5 1001 = IRQ9 1101 = $\overline{\text{LED\_SKT}}$ 0010 = $\overline{\text{LOCK}}$ PCI 0110 = $\overline{\text{ZVSTAT}}$ 1010 = IRQ10 1110 = $\overline{\text{GPE}}$ 0011 = IRQ3 0111 = $\overline{\text{ZVSEL1}}$ 1011 = IRQ11 1111 = $\overline{\text{D3\_STAT}}$
15–12	MFUNC3	RW	Multifunction terminal 3 configuration. These bits control the internal signal mapped to the MFUNC3 terminal as follows: 0000 = RSVD 0100 = IRQ4 1000 = IRQ8 1100 = IRQ12 0001 = IRQSER† 0101 = IRQ5 1001 = IRQ9 1101 = IRQ13 0010 = IRQ2 0110 = IRQ6 1010 = IRQ10 1110 = IRQ14 0011 = IRQ3 0111 = IRQ7 1011 = IRQ11 1111 = IRQ15
11–8	MFUNC2	RW	Multifunction terminal 2 configuration. These bits control the internal signal mapped to the MFUNC2 terminal as follows: 0000 = GPI2† 0100 = IRQ4 1000 = CAUDPWM 1100 = $\overline{\text{RI\_OUT}}$ 0001 = $\overline{\text{GPO2}}$ 0101 = IRQ5 1001 = IRQ9 1101 = $\overline{\text{LEDA2}}$ 0010 = $\overline{\text{PCREQ}}$ 0110 = $\overline{\text{ZVSTAT}}$ 1010 = $\overline{\text{IRQ10}}$ 1110 = $\overline{\text{GPE}}$ 0011 = IRQ3 0111 = $\overline{\text{ZVSEL0}}$ 1011 = $\overline{\text{D3\_STAT}}$ 1111 = IRQ7

Table 4–9. Multifunction Routing Register Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION
7–4	MFUNC1	RW	Multifunction terminal 1 configuration. These bits control the internal signal mapped to the MFUNC1 terminal as follows: NOTE: When the serial bus mode is implemented by pulling down the LATCH terminal, the MFUNC1 terminal provides the SDA signaling. 0000 = GPI1† 0100 = IRQ4 1000 = CAUDPWM 1100 = LEDA1 0001 = $\overline{\text{GPO1}}$ 0101 = IRQ5 1001 = IRQ9 1101 = $\overline{\text{LEDA2}}$ 0010 = $\overline{\text{INTB}}$ 0110 = $\overline{\text{ZVSTAT}}$ 1010 = IRQ10 1110 = $\overline{\text{GPE}}$ 0011 = IRQ3 0111 = $\overline{\text{ZVSEL0}}$ 1011 = IRQ11 1111 = IRQ15
3–0	MFUNC0	RW	Multifunction terminal 0 configuration. These bits control the internal signal mapped to the MFUNC0 terminal as follows: 0000 = GPI0† 0100 = IRQ4 1000 = CAUDPWM 1100 = LEDA1 0001 = $\overline{\text{GPO0}}$ 0101 = IRQ5 1001 = IRQ9 1101 = $\overline{\text{LEDA2}}$ 0010 = $\overline{\text{INTA}}$ 0110 = $\overline{\text{ZVSTAT}}$ 1010 = IRQ10 1110 = $\overline{\text{GPE}}$ 0011 = IRQ3 0111 = $\overline{\text{ZVSEL0}}$ 1011 = IRQ11 1111 = IRQ15

† Default value

4.31 Retry Status Register

The retry status register enables the retry timeout counters and displays the retry expiration status. The flags are set when the PCI1520 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. See Table 4–10 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Retry status							
Type	RW	RW	RC	R	RC	R	RC	R
Default	1	1	0	0	0	0	0	0

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

Table 4–10. Retry Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	PCIRETRY	RW	PCI retry timeout counter enable. Bit 7 is encoded: 0 = PCI retry counter disabled 1 = PCI retry counter enabled (default)
6†	CBRETRY	RW	CardBus retry timeout counter enable. Bit 6 is encoded: 0 = CardBus retry counter disabled 1 = CardBus retry counter enabled (default)
5	TEXP_CBB	RC	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	RC	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	RC	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.

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

4.32 Card Control Register

The card control register is provided for PCI1130 compatibility. $\overline{\text{RI\_OUT}}$ is enabled through this register, and the enable bit is shared between functions 0 and 1. See Table 4–11 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Card control							
Type	RW	RW	RW	R	R	RW	RW	RC
Default	0	0	0	0	0	0	0	0

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

Table 4–11. Card Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7†	RIENB	RW	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 $\overline{\text{RI\_OUT}}/\overline{\text{PME}}$ terminal, when RIMUX is set to 0, and for routing to MFUNC2 or MFUNC4
6	ZVENABLE	RW	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	PORT_SEL	RW	Port select. This bit controls the priority for the $\overline{\text{ZVSEL0}}$ and $\overline{\text{ZVSEL1}}$ signaling if bit 6 (ZVENABLE) is set in both functions. 0 = Socket 0 takes priority, as signaled through $\overline{\text{ZVSEL0}}$, when both sockets are in ZV mode. 1 = Socket 1 takes priority, as signaled through $\overline{\text{ZVSEL1}}$, when both sockets are in ZV mode.
4–3	RSVD	R	Reserved. Bits 4 and 3 return 0 when read.
2	AUD2MUX	RW	CardBus audio-to-IRQMUX. When set, the CAUDIO CardBus signal is routed to the corresponding multifunction terminal which may be configured for CAUDPWM. When both socket 0 and 1 functions have AUD2MUX set, socket 0 takes precedence.
1	SPKROUTEN	RW	Speaker out enable. When bit 1 is set, $\overline{\text{SPKR}}$ on the PC Card is enabled and is routed to SPKROUT. The $\overline{\text{SPKR}}$ signal from socket 0 is XORed with the $\overline{\text{SPKR}}$ signal from socket 1 and sent to SPKROUT. The SPKROUT terminal drives data only when the SPKROUTEN bit of either function is set. This bit is encoded as: 0 = $\overline{\text{SPKR}}$ to SPKROUT not enabled 1 = $\overline{\text{SPKR}}$ to SPKROUT enabled
0	IFG	RC	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 (that is, not global). Write back a 1 to clear this bit. 0 = No PC Card functional interrupt detected (default). 1 = PC Card functional interrupt detected.

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

4.33 Device Control Register

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 PCI1520 global bits. The socket-capable force bits are also programmed through this register. See Table 4–12 for a complete description of the register contents.

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

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

Table 4–12. Device Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	SKTPWR_LOCK	RW	Socket power lock bit. When this bit is set to 1, software cannot power down the PC Card socket while in D3. This may be necessary to support wake on LAN or RING if the operating system is programmed to power down a socket when the CardBus controller is placed in the D3 state.
6†	3VCAPABLE	RW	3-V socket capable force 0 = Not 3-V capable 1 = 3-V capable (default)
5	IO16V2	RW	Diagnostic bit. This bit defaults to 1.
4	RSVD	R	Reserved. Bit 4 returns 0 when read.
3†	TEST	RW	TI test. Only a 0 should be written to bit 3.
2–1	INTMODE	RW	Interrupt signaling mode. Bits 2 and 1 select the interrupt signaling mode. The interrupt signaling mode bits are encoded: 00 = Parallel PCI interrupts only 01 = Parallel IRQ and parallel PCI interrupts 10 = IRQ serialized interrupts and parallel PCI interrupt 11 = IRQ and PCI serialized interrupts (default)
0†	RSVD	RW	Reserved. Bit 0 is reserved for test purposes. Only 0 should be written to this bit.

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

4.34 Diagnostic Register

The diagnostic register is provided for internal TI test purposes. It is a read/write register, but only 0s should be written to it. See Table 4–13 for a complete description of the register contents.

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

Register: **Diagnostic**
Offset: 93h (functions 0, 1)
Type: Read/Write
Default: 60h

Table 4–13. Diagnostic Register Description

BIT	SIGNAL	TYPE	FUNCTION
7†	TRUE_VAL	RW	This bit defaults to 0. This bit is encoded as: 0 = Reads true values in PCI vendor ID and PCI device ID registers (default) 1 = Reads all 1s in reads from the PCI vendor ID and PCI device ID registers
6	RSVD	R	Reserved. Bit 6 returns 1 when read.
5	CSC	RW	CSC interrupt routing control 0 = CSC interrupts routed to PCI if ExCA 803 bit 4 = 1 1 = CSC interrupts routed to PCI if ExCA 805 bits 7–4 = 0000b (default) In this case, the setting of ExCA 803 bit 4 is a don't care.
4†	DIAG4	RW	Diagnostic RETRY_DIS. Delayed transaction disable.
3†	DIAG3	RW	Diagnostic RETRY_EXT. Extends the latency from 16 to 64.
2†	DIAG2	RW	Diagnostic DISCARD_TIM_SEL_CB. Set = 2^{10}, reset = 2^{15}.
1†	DIAG1	RW	Diagnostic DISCARD_TIM_SEL_PCI. Set = 2^{10}, reset = 2^{15}.
0	STDZVEN	RW	Standardized zoomed video register model enable. 0 = Enable the standardized zoomed video register model (default). 1 = Disable the standardized zoomed video register model.

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

4.35 Capability ID Register

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

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

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

4.36 Next-Item Pointer Register

The next-item pointer register indicates the next item in the linked list of the PCI power-management capabilities. Because the PCI1520 functions include only one capabilities item, this register returns 0s when read.

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

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

4.37 Power-Management Capabilities Register

This register contains information on the capabilities of the PC Card function related to power management. Both PCI1520 CardBus bridge functions support D0, D1, D2, and D3 power states. See Table 4–14 for a complete description of the register contents.

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

Register: **Power-management capabilities**
Offset: A2h
Type: Read/Write, Read-only
Default: FE12h

Table 4–14. Power-Management Capabilities Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	PME_Support	RW	$\overline{\text{PME}}$ support. This 5-bit field indicates the power states from which the PCI1520 device functions may assert $\overline{\text{PME}}$. A 0 (zero) for any bit indicates that the function cannot assert the $\overline{\text{PME}}$ signal while in that power state. These five bits return 11111b when read. Each of these bits is described below: Bit 15 defaults to the value 1 indicating the $\overline{\text{PME}}$ signal can be asserted from the $D3_{cold}$ state. This bit is R/W because wake-up support from $D3_{cold}$ is contingent on the system providing an auxiliary power source to the V_{CC} terminals. If the system designer chooses not to provide an auxiliary power source to the V_{CC} terminals for $D3_{cold}$ wake-up support, then BIOS should write a 0 to this bit.
14–11	PME_Support	R	Bit 14 contains the value 1, indicating that the $\overline{\text{PME}}$ signal can be asserted from $D3_{hot}$ state. Bit 13 contains the value 1, indicating that the $\overline{\text{PME}}$ signal can be asserted from D2 state. Bit 12 contains the value 1, indicating that the $\overline{\text{PME}}$ signal can be asserted from D1 state. Bit 11 contains the value 1, indicating that the $\overline{\text{PME}}$ signal can be asserted from the D0 state.
10	D2_Support	R	D2 support. Bit 10 returns a 1 when read, indicating that the CardBus function supports the D2 device power state.
9	D1_Support	R	D1 support. Bit 9 returns a 1 when read, indicating that the CardBus function supports the D1 device power state.
8–6	RSVD	R	Reserved. Bits 8–6 return 0s when read.
5	DSI	R	Device-specific initialization. Bit 5 returns 1 when read, indicating that the CardBus controller function requires special initialization (beyond the standard PCI configuration header) before the generic-class device driver is able to use it.
4	AUX_PWR	R	Auxiliary power source. Bit 4 is meaningful only if bit 15 (PME_Support, $D3_{cold}$) is set. When bit 4 is set, it indicates that support for $\overline{\text{PME}}$ in $D3_{cold}$ requires auxiliary power supplied by the system by way of a proprietary delivery vehicle. When bit 4 is 0, it indicates that the function supplies its own auxiliary power source.
3	PMECLK	R	$\overline{\text{PME}}$ clock. Bit 3 returns 0 when read, indicating that no host bus clock is required for the PCI1520 to generate $\overline{\text{PME}}$.
2–0	VERSION	R	Version. Bits 2–0 return 010b when read, indicating that the power-management registers (PCI offsets A4h–A7h, see Sections 4.38–4.40) are defined in the *PCI Bus Power Management Interface Specification* version 1.1.

4.38 Power-Management Control/Status Register

The power-management control/status register determines and changes the current power state of the PCI1520 CardBus function. The contents of this register are not affected by the internally-generated reset caused by the transition from $D3_{hot}$ to D0 state. All PCI, ExCA, and CardBus registers are reset as a result of a $D3_{hot}$ to D0 state transition. TI-specific registers, PCI power-management registers, and the PC Card 16-bit legacy-mode base address register (PCI offset 44h, see Section 4.28) are not reset. See Table 4–15 for a complete description of the register contents.

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

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

Table 4–15. Power-Management Control/Status Register Description

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

4.39 Power-Management Control/Status Register Bridge Support Extensions

The power-management control/status register bridge support extensions support PCI bridge specific functionality. See Table 4–16 for a complete description of the register contents.

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

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

Table 4–16. Power-Management Control/Status Register Bridge Support Extensions Description

BIT	SIGNAL	TYPE	FUNCTION
7	BPCC_EN	R	BPCC_Enable. Bus power/clock control enable. This bit returns 1 when read. This bit is encoded as: 0 = Bus power/clock control is disabled. 1 = Bus power/clock control is enabled (default). A 0 indicates that the bus power/clock control policies defined in the *PCI Bus Power Management Interface Specification* are disabled. When the bus power/clock control enable mechanism is disabled, the bridge power-management control/status register power state field (see Section 4.38, bits 1–0) cannot be used by the system software to control the power or the clock of the bridge secondary bus. A 1 indicates that the bus power/clock control mechanism is enabled.
6	B2_$\overline{B3}$	R	B2/B3 support for $D3_{hot}$. The state of this bit determines the action that is to occur as a direct result of programming the function to $D3_{hot}$. This bit is only meaningful if bit 7 (BPCC_EN) is a 1. This bit is encoded as: 0 = When the bridge is programmed to $D3_{hot}$, its secondary bus has its power removed (B3). 1 = When the bridge function is programmed to $D3_{hot}$, its secondary bus PCI clock is stopped (B2) (default).
5–0	RSVD	R	Reserved. Bits 5–0 return 0s when read.

4.40 Power-Management Data Register

The power-management data register returns 0s when read, because the CardBus functions do not report dynamic data.

Bit	7	6	5	4	3	2	1	0
Name	Power-management data							
Type	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **Power-management data**
Offset: A7h (functions 0, 1)
Type: Read-only
Default: 00h

4.41 General-Purpose Event Status Register

The general-purpose event status register contains status bits that are set when events occur that are controlled by the general-purpose control register. The bits in this register and the corresponding $\overline{GPE}$ are cleared by writing a 1 to the corresponding bit location. The status bits in this register do not depend upon the states of corresponding bits in the general-purpose enable register. Access this register only through function 0. See Table 4–17 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	General-purpose event status															
Type	RC	RC	R	R	RC	R	R	RC	R	R	R	RC	RC	RC	RC	RC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **General-purpose event status**
Offset: A8h (function 0)
Type: Read-only, Read/Clear
Default: 0000h

Table 4–17. General-Purpose Event Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	ZV0_STS	RC	PC Card socket 0 ZV status. Bit 15 is set on a change in status of bit 6 (ZVENABLE) in the function 0 card control register (PCI offset 91h, see Section 4.32).
14	ZV1_STS	RC	PC Card socket 1 ZV status. Bit 14 is set on a change in status of bit 6 (ZVENABLE) in the function 1 card control register (PCI offset 91h, see Section 4.32).
13–12	RSVD	R	Reserved. Bits 13 and 12 return 0s when read.
11	PWR_STS	RC	Power-change status. Bit 11 is set when software has changed the power state of either socket. A change in either V_{CC} or V_{PP} for either socket causes this bit to be set.
10–9	RSVD	R	Reserved. Bits 10 and 9 return 0s when read.
8	VPP12_STS	RC	12-V V_{PP} request status. Bit 8 is set when software has changed the requested V_{PP} level to or from 12 V for either of the two PC Card sockets.
7–5	RSVD	R	Reserved. Bits 7–5 return 0s when read.
4	GP4_STS	RC	GPI4 Status. Bit 4 is set on a change in status of the MFUNC5 terminal input level.
3	GP3_STS	RC	GPI3 Status. Bit 3 is set on a change in status of the MFUNC4 terminal input level .
2	GP2_STS	RC	GPI2 Status. Bit 2 is set on a change in status of the MFUNC2 terminal input level.
1	GP1_STS	RC	GPI1 Status. Bit 1 is set on a change in status of the MFUNC1 terminal input level.
0	GP0_STS	RC	GPI0 Status. Bit 0 is set on a change in status of the MFUNC0 terminal input level.

4.42 General-Purpose Event Enable Register

The general-purpose event enable register contains bits that are set to enable a $\overline{\text{GPE}}$ signal. The $\overline{\text{GPE}}$ signal is driven until the corresponding status bit is cleared and the event is serviced. The $\overline{\text{GPE}}$ can only be signaled if one of the multifunction terminals, MFUNC6–MFUNC0, is configured for $\overline{\text{GPE}}$ signaling. Access this register only through function 0. See Table 4–18 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	General-purpose event enable															
Type	RW	RW	R	R	RW	R	R	RW	R	R	R	RW	RW	RW	RW	RW
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **General-purpose event enable**
Offset: AAh (function 0)
Type: Read-only, Read/Write
Default: 0000h

Table 4–18. General-Purpose Event Enable Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	ZV0_EN	RW	PC Card socket 0 ZV enable. When bit 15 is set, a $\overline{\text{GPE}}$ is signaled on a change in status of bit 6 (ZVENABLE) in the function 0 card control register (PCI offset 91h, see Section 4.32).
14	ZV1_EN	RW	PC Card socket 1 ZV enable. When bit 14 is set, a $\overline{\text{GPE}}$ is signaled on a change in status of bit 6 (ZVENABLE) in the function 1 card control register (PCI offset 91h, see Section 4.32).
13–12	RSVD	R	Reserved. Bits 13 and 12 return 0s when read.
11	PWR_EN	RW	Power change enable. When bit 11 is set, a $\overline{\text{GPE}}$ is signaled on when software has changed the power state of either socket.
10–9	RSVD	R	Reserved. Bits 10 and 9 return 0s when read.
8	VPP12_EN	RW	12-V V_{PP} request enable. When bit 8 is set, a $\overline{\text{GPE}}$ is signaled when software has changed the requested V_{PP} level to or from 12 V for either card socket.
7–5	RSVD	R	Reserved. Bits 7–5 return 0s when read.
4	GP4_EN	RW	GPI4 enable. When bit 4 is set, a $\overline{\text{GPE}}$ is signaled when there has been a change in status of the MFUNC5 terminal input level if configured as GPI4.
3	GP3_EN	RW	GPI3 enable. When bit 3 is set, a $\overline{\text{GPE}}$ is signaled when there has been a change in status of the MFUNC4 terminal input level if configured as GPI3.
2	GP2_EN	RW	GPI2 enable. When bit 2 is set, a $\overline{\text{GPE}}$ is signaled when there has been a change in status of the MFUNC2 terminal input if configured as GPI2.
1	GP1_EN	RW	GPI1 enable. When bit 1 is set, a $\overline{\text{GPE}}$ is signaled when there has been a change in status of the MFUNC1 terminal input if configured as GPI1.
0	GP0_EN	RW	GPI0 enable. When bit 0 is set, a $\overline{\text{GPE}}$ is signaled when there has been a change in status of the MFUNC0 terminal input if configured as GPI0.

4.43 General-Purpose Input Register

The general-purpose input register provides the logical value of the data input from the GPI terminals, MFUNC5, MFUNC4, and MFUNC2–MFUNC0. Access this register only through function 0. See Table 4–19 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	General-purpose input															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X

Register: **General-purpose input**
Offset: ACh (function 0)
Type: Read-only
Default: 00XXh

Table 4–19. General-Purpose Input Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–5	RSVD	R	Reserved. Bits 15–5 return 0s when read.
4	GPI4_DATA	R	GPI4 data bit. The value read from bit 4 represents the logical value of the data input from the MFUNC5 terminal.
3	GPI3_DATA	R	GPI3 data bit. The value read from bit 3 represents the logical value of the data input from the MFUNC4 terminal.
2	GPI2_DATA	R	GPI2 data bit. The value read from bit 2 represents the logical value of the data input from the MFUNC2 terminal.
1	GPI1_DATA	R	GPI1 data bit. The value read from bit 1 represents the logical value of the data input from the MFUNC1 terminal.
0	GPI0_DATA	R	GPI0 data bit. The value read from bit 0 represents the logical value of the data input from the MFUNC0 terminal.

4.44 General-Purpose Output Register

The general-purpose output register is used for control of the general-purpose outputs. Access this register only through function 0. See Table 4–20 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	General-purpose output															
Type	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **General-purpose output**
Offset: AEh (function 0)
Type: Read-only, Read/Write
Default: 0000h

Table 4–20. General-Purpose Output Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–5	RSVD	R	Reserved. Bits 15–5 return 0s when read.
4	GPO4_DATA	RW	GPO4 data bit. The value written to bit 4 represents the logical value of the data driven to the MFUNC5 terminal if configured as GPO4. Read transactions return the last data value written.
3	GPO3_DATA	RW	GPIO3 data bit. The value written to bit 3 represents the logical value of the data driven to the MFUNC4 terminal if configured as GPO3. Read transactions return the last data value written.
2	GPO2_DATA	RW	GPO2 data bit. The value written to bit 2 represents the logical value of the data driven to the MFUNC2 terminal if configured as GPO2. Read transactions return the last data value written.
1	GPO1_DATA	RW	GPO1 data bit. The value written to bit 1 represents the logical value of the data driven to the MFUNC1 terminal if configured as GPO1. Read transactions return the last data value written.
0	GPO0_DATA	RW	GPO0 data bit. The value written to bit 0 represents the logical value of the data driven to the MFUNC0 terminal if configured as GPO0. Read transactions return the last data value written.

4.45 Serial-Bus Data Register

The serial-bus data register is for programmable serial-bus byte reads and writes. This register represents the data when generating cycles on the serial bus interface. To write a byte, this register must be programmed with the data, the serial bus index register must be programmed with the byte address, the serial-bus slave address must be programmed with the 7-bit slave address, and the read/write indicator bit must be reset.

On byte reads, the byte address is programmed into the serial-bus index register, the serial bus slave address register must be programmed with both the 7-bit slave address and the read/write indicator bit, and bit 5 (REQBUSY) in the serial bus control and status register (PCI offset B3h, see Section 4.48) must be polled until clear. Then the contents of this register are valid read data from the serial bus interface. See Table 4–21 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Serial-bus data							
Type	RW	RW	RW	RW	RW	RW	RW	RW
Default	0	0	0	0	0	0	0	0

Register: **Serial-bus data**
Offset: B0h (function 0)
Type: Read/Write
Default: 00h

Table 4–21. Serial-Bus Data Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–0	SBDATA	RW	Serial-bus data. This bit field represents the data byte in a read or write transaction on the serial interface. On reads, the REQBUSY bit must be polled to verify that the contents of this register are valid.

4.46 Serial-Bus Index Register

The serial-bus index register is for programmable serial-bus byte reads and writes. This register represents the byte address when generating cycles on the serial-bus interface. To write a byte, the serial-bus data register must be programmed with the data, this register must be programmed with the byte address, and the serial-bus slave address register must be programmed with both the 7-bit slave address and the read/write indicator bit.

On byte reads, the word address is programmed into this register, the serial-bus slave address must be programmed with both the 7-bit slave address and the read/write indicator bit, and bit 5 (REQBUSY) in the serial-bus control and status register (see Section 4.48) must be polled until clear. Then the contents of the serial-bus data register are valid read data from the serial-bus interface. See Table 4–22 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Serial-bus index							
Type	RW	RW	RW	RW	RW	RW	RW	RW
Default	0	0	0	0	0	0	0	0

Register: **Serial-bus index**
Offset: B1h (function 0)
Type: Read/Write
Default: 00h

Table 4–22. Serial-Bus Index Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–0	SBINDEX	RW	Serial-bus index. This bit field represents the byte address in a read or write transaction on the serial interface.

4.47 Serial-Bus Slave Address Register

The serial-bus slave address register is for programmable serial-bus byte read and write transactions. To write a byte, the serial-bus data register must be programmed with the data, the serial-bus index register must be programmed with the byte address, and this register must be programmed with both the 7-bit slave address and the read/write indicator bit.

On byte reads, the byte address is programmed into the serial bus index register, this register must be programmed with both the 7-bit slave address and the read/write indicator bit, and bit 5 (REQBUSY) in the serial-bus control and status register (PCI offset B3h, see Section 4.48) must be polled until clear. Then the contents of the serial-bus data register are valid read data from the serial-bus interface. See Table 4–23 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Serial-bus slave address							
Type	RW	RW	RW	RW	RW	RW	RW	RW
Default	0	0	0	0	0	0	0	0

Register: **Serial-bus slave address**
Offset: B2h (function 0)
Type: Read/Write
Default: 00h

Table 4–23. Serial-Bus Slave Address Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–1	SLAVADDR	RW	Serial-bus slave address. This bit field represents the slave address of a read or write transaction on the serial interface.
0	RWCMD	RW	Read/write command. Bit 0 indicates the read/write command bit presented to the serial bus on byte read and write accesses. 0 = A byte write access is requested to the serial bus interface. 1 = A byte read access is requested to the serial bus interface.

4.48 Serial-Bus Control and Status Register

The serial-bus control and status register communicates serial-bus status information and selects the quick command protocol. Bit 5 (REQBUSY) in this register must be polled during serial-bus byte reads to indicate when data is valid in the serial-bus data register. See Table 4–24 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Serial-bus control and status							
Type	RW	R	R	R	RC	RW	RC	RC
Default	0	0	0	0	0	0	0	0

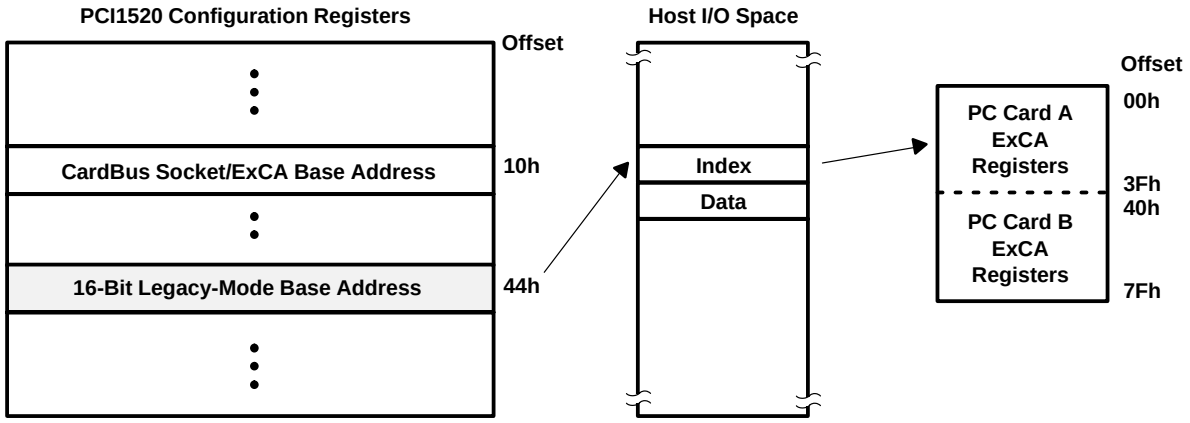
Register: **Serial-bus control and status**
Offset: B3h (function 0)
Type: Read-only, Read/Write, Read/Clear
Default: 00h

Table 4–24. Serial-Bus Control and Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	PROT_SEL	RW	Protocol select. When bit 7 is set, the send-byte protocol is used on write requests and the receive-byte protocol is used on read commands. The word-address byte in the serial-bus index register (PCI offset B1h, see Section 4.46) is not output by the PCI1520 when bit 7 is set.
6	RSVD	R	Reserved. Bit 6 returns 0 when read.
5	REQBUSY	R	Requested serial-bus access busy. Bit 5 indicates that a requested serial-bus access (byte read or write) is in progress. A request is made, and bit 5 is set, by writing to the serial-bus slave address register (PCI offset B2h, see Section 4.47). Bit 5 must be polled on reads from the serial interface. After the byte read access has been requested, the read data is valid in the serial-bus data register.
4	ROMBUSY	R	Serial EEPROM busy status. Bit 4 indicates the status of the PCI1520 serial EEPROM circuitry. Bit 4 is set during the loading of the subsystem ID and other default values from the serial-bus EEPROM. 0 = Serial EEPROM circuitry is not busy 1 = Serial EEPROM circuitry is busy
3	SBDETECT	RC	Serial-bus detect. When bit 3 is set, it indicates that the serial-bus interface is detected. A pulldown resistor must be implemented on the LATCH terminal for bit 3 to be set. If bit 3 is reset, then the MFUNC4 and MFUNC1 terminals can be used for alternate functions such as general-purpose inputs and outputs. 0 = Serial-bus interface not detected 1 = Serial-bus interface detected
2	SBTEST	RW	Serial-bus test. When bit 2 is set, the serial-bus clock frequency is increased for test purposes. 0 = Serial-bus clock at normal operating frequency, ≈ 100 kHz (default) 1 = Serial-bus clock frequency increased for test purposes
1	REQ_ERR	RC	Requested serial-bus access error. Bit 1 indicates when a data error occurs on the serial interface during a requested cycle, and can be set due to a missing acknowledge. Bit 1 is cleared by a writeback of 1. 0 = No error detected during user-requested byte read or write cycle 1 = Data error detected during user-requested byte read or write cycle
0	ROM_ERR	RC	EEPROM data-error status. Bit 0 indicates when a data error occurs on the serial interface during the auto-load from the serial-bus EEPROM, and can be set due to a missing acknowledge. Bit 0 is also set on invalid EEPROM data formats. See Section 3.6.1, *Serial Bus Interface Implementation*, for details on EEPROM data format. Bit 0 is cleared by a writeback of 1. 0 = No error detected during auto-load from serial-bus EEPROM 1 = Data error detected during auto-load from serial-bus EEPROM

5 ExCA Compatibility Registers (Functions 0 and 1)

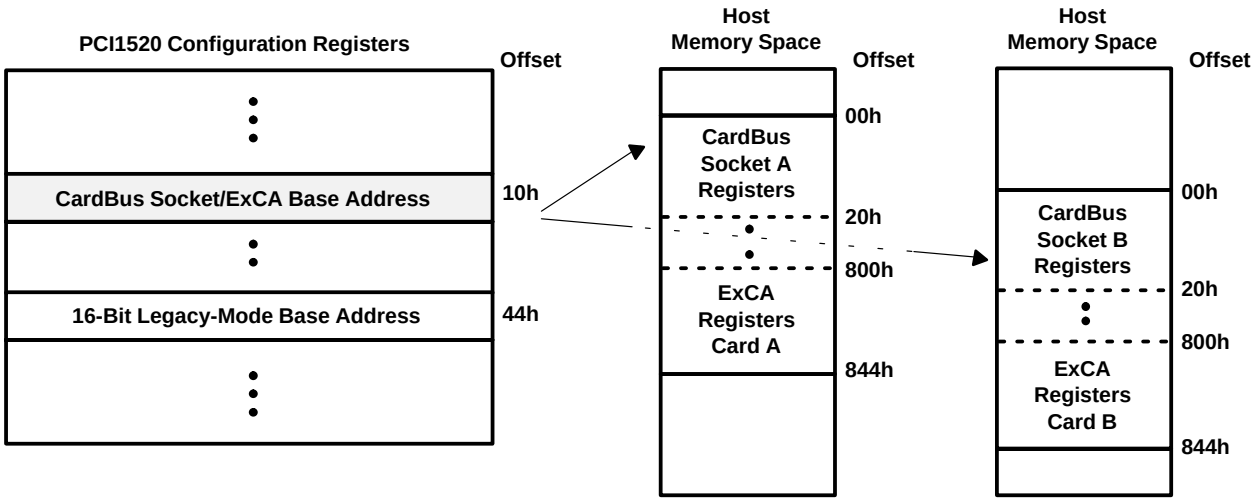
The ExCA registers implemented in the PCI1520 are register-compatible with the Intel 82365SL–DF PCMCIA controller. ExCA registers are identified by an offset value that is compatible with the legacy I/O index/data scheme used on the Intel 82365 ISA controller. The ExCA registers are accessed through 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 (PCI offset 44h, see Section 4.28), which is shared by both card sockets. The offsets from this base address run contiguously from 00h to 3Fh for socket A, and from 40h to 7Fh for socket B. See Figure 5–1 for an ExCA I/O mapping illustration.



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

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

The TI PCI1520 also provides a memory-mapped alias of the ExCA registers by directly mapping them into PCI memory space. They are located through the CardBus socket/ExCA base-address register (PCI offset 10h, see Section 4.12) at memory offset 800h. Each socket has a separate base address programmable by function. See Figure 5–2 for an ExCA memory mapping illustration. Note that memory offsets are 800h–844h for both functions 0 and 1. This illustration also identifies the CardBus socket register mapping, which is mapped into the same 4-K window at memory offset 00h.



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

Figure 5–2. ExCA Register Access Through Memory

The interrupt registers in the ExCA register set, as defined by the 82365SL–DL specification, 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 PCI1520 to ensure that all possible PCI1520 interrupts can potentially be routed to the programmable interrupt controller. The ExCA registers that are critical to the interrupt signaling are the ExCA interrupt and general control register (ExCA offset 03h/43h/803h, see Section 5.4) and the ExCA card status-change interrupt configuration register (05h/45h/805h, see Section 5.6).

Access to I/O mapped 16-bit PC cards is available to the host system via two ExCA I/O windows. These are regions of host I/O address space into which the card I/O space is mapped. These windows are defined by start, end, and offset addresses programmed in the ExCA registers described in this section. I/O windows have byte granularity.

Access to memory mapped 16-bit PC Cards is available to the host system via five ExCA memory windows. These are regions of host memory space into which the card memory space is mapped. These windows are defined by start, end, and offset addresses programmed in the ExCA registers described in this section. Table 5–1 identifies each ExCA register and its respective ExCA offset. Memory windows have 4-Kbyte granularity.

Table 5–1. ExCA Registers and Offsets

ExCA REGISTER NAME	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
Memory window 3 start-address low byte	828	28	68
Memory window 3 start-address high byte	829	29	69
Memory window 3 end-address low byte	82A	2A	6A

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

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

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

5.1 ExCA Identification and Revision Register

The ExCA identification and revision register provides host software with information on 16-bit PC Card support and Intel 82365SL-DF compatibility. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register (PCI offset 80h, see Section 4.29). See Table 5–2 for a complete description of the register contents.

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

Register: **ExCA identification and revision**
Offset: CardBus socket address + 800h; Card A ExCA offset 00h
Card B ExCA offset 40h
Type: Read-only, Read/Write
Default: 84h

Table 5–2. ExCA Identification and Revision Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–6	IFTYPE	R	Interface type. These bits, which are hardwired as 10b, identify the 16-bit PC Card support provided by the PCI1520. The PCI1520 supports both I/O and memory 16-bit PC cards.
5–4	RSVD	RW	Reserved. Bits 5 and 4 can be used for Intel 82365SL-DF emulation.
3–0	365REV	RW	Intel 82365SL-DF revision. This field stores the Intel 82365SL-DF revision supported by the PCI1520. Host software can read this field to determine compatibility to the Intel 82365SL-DF register set. Writing 0010b to this field puts the controller in 82365SL mode.

5.2 ExCA Interface Status Register

The ExCA interface status register provides information on the current status of the PC Card interface. An X in the default bit value indicates that the value of the bit after reset depends on the state of the PC Card interface. See Table 5–3 for a complete description of the register contents.

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

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

Table 5–3. ExCA Interface Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	RSVD	R	Reserved. Bit 7 returns 0 when read.
6	CARDPWR	R	Card Power. Bit 6 indicates the current power status of the PC Card socket. This bit reflects how the ExCA power control register (ExCA offset 02h/42h/802h, see Section 5.3) is programmed. Bit 6 is encoded as: 0 = V_{CC} and V_{PP} to the socket turned off (default) 1 = V_{CC} and V_{PP} to the socket turned on
5	READY	R	Ready. Bit 5 indicates the current status of the READY signal at the PC Card interface. 0 = PC Card not ready for data transfer 1 = PC Card ready for data transfer
4	CARDWP	R	Card write protect (WP). Bit 4 indicates the current status of WP at the PC Card interface. This signal reports to the PCI1520 whether or not the memory card is write protected. Furthermore, write protection for an entire PCI1520 16-bit memory window is available by setting the appropriate bit in the memory window offset-address high-byte register. 0 = WP is 0. PC Card is read/write. 1 = WP is 1. PC Card is read-only.
3	CDETECT2	R	Card detect 2. Bit 3 indicates the status of $\overline{CD2}$ at the PC Card interface. Software may use this and bit 2 (CDETECT1) to determine if a PC Card is fully seated in the socket. 0 = $\overline{CD2}$ is 1. No PC Card is inserted. 1 = $\overline{CD2}$ is 0. PC Card is at least partially inserted.
2	CDETECT1	R	Card detect 1. Bit 2 indicates the status of $\overline{CD1}$ at the PC Card interface. Software may use this and bit 3 (CDETECT2) to determine if a PC Card is fully seated in the socket. 0 = $\overline{CD1}$ is 1. No PC Card is inserted. 1 = $\overline{CD1}$ is 0. PC Card is at least partially inserted.
1–0	BVDSTAT	R	Battery voltage detect. When a 16-bit memory card is inserted, the field indicates the status of the battery voltage detect signals (BVD1, BVD2) at the PC Card interface, where bit 1 reflects the BVD2 status and bit 0 reflects BVD1. 00 = Battery dead 01 = Battery dead 10 = Battery low; warning 11 = Battery good When a 16-bit I/O card is inserted, this field indicates the status of $\overline{SPKR}$ (bit 1) and $\overline{STSCHG}$ (bit 0) at the PC Card interface. In this case, the two bits in this field directly reflect the current state of these card outputs.

5.3 ExCA Power Control Register

The ExCA power control register provides PC Card power control. Bit 7 (COE) of this register controls the 16-bit output enables on the socket interface, and can be used for power management in 16-bit PC Card applications. See Table 5–4 and Table 5–5 for a complete description of the register contents.

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

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

Table 5–4. ExCA Power Control Register Description—82365SL Support

BIT	SIGNAL	TYPE	FUNCTION
7	COE	RW	Card output enable. Bit 7 controls the state of all of the 16-bit outputs on the PCI1520. This bit is encoded as: 0 = 16-bit PC Card outputs disabled (default) 1 = 16-bit PC Card outputs enabled
6	RSVD	R	Reserved. Bit 6 returns 0 when read.
5	AUTOPWRSWEN	RW	Auto power switch enable. 0 = Automatic socket power switching based on card detects is disabled. 1 = Automatic socket power switching based on card detects is enabled.
4	CAPWREN	RW	PC Card power enable. 0 = V_{CC} = No connection 1 = V_{CC} is enabled and controlled by bit 2 (EXCAPOWER) of the system control register (PCI offset 80h, see Section 4.29).
3–2	RSVD	R	Reserved. Bits 3 and 2 return 0s when read.
1–0	EXCAVPP	RW	PC Card V_{PP} power control. Bits 1 and 0 are used to request changes to card V_{PP}. The PCI1520 ignores this field unless V_{CC} to the socket is enabled. This field is encoded as: 00 = No connection (default) 01 = V_{CC} 10 = 12 V 11 = Reserved

Table 5–5. ExCA Power Control Register Description—82365SL-DF Support

BIT	SIGNAL	TYPE	FUNCTION
7	COE	RW	Card output enable. Bit 7 controls the state of all of the 16-bit outputs on the PCI1520. This bit is encoded as: 0 = 16-bit PC Card outputs disabled (default) 1 = 16-bit PC Card outputs enabled
6–5	RSVD	R	Reserved. Bits 6 and 5 return 0s when read.
4–3	EXCAVCC	RW	V_{CC}. Bits 4 and 3 are used to request changes to card V_{CC}. This field is encoded as: 00 = 0 V (default) 01 = 0 V reserved 10 = 5 V 11 = 3.3 V
2	RSVD	R	Reserved. Bit 2 returns 0 when read.
1–0	EXCAVPP	RW	V_{PP}. Bits 1 and 0 are used to request changes to card V_{PP}. The PCI1520 ignores this field unless V_{CC} to the socket is enabled. This field is encoded as: 00 = No connection (default) 01 = V_{CC} 10 = 12 V 11 = Reserved

5.4 ExCA Interrupt and General Control Register

The ExCA interrupt and general control register controls interrupt routing for I/O interrupts, as well as other critical 16-bit PC Card functions. See Table 5–6 for a complete description of the register contents.

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

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

Table 5–6. ExCA Interrupt and General Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	RINGEN	RW	Card ring indicate enable. Bit 7 enables the ring indicate function of BVD1/$\overline{RI}$. This bit is encoded as: 0 = Ring indicate disabled (default) 1 = Ring indicate enabled
6	RESET	RW	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	RW	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	RW	PCI interrupt CSC routing enable bit. When bit 4 is set (high), the card status change interrupts are routed to PCI interrupts. When low, the card status change interrupts are routed using bits 7–4 (CSCSELECT field) in the ExCA card status-change interrupt configuration register (ExCA offset 05h/45h/805h, see Section 5.6). This bit is encoded as: 0 = CSC interrupts are routed by ExCA registers (default). 1 = CSC interrupts are routed to PCI interrupts.
3–0	INTSELECT	RW	Card interrupt select for I/O PC Card functional interrupts. Bits 3–0 select the interrupt routing for I/O PC Card functional interrupts. This field is encoded as: 0000 = No interrupt routing (default). CSC interrupts are routed to PCI interrupts. This bit setting is ORed with bit 4 (CSCROUTE) for backward compatibility. 0001 = IRQ1 enabled 0010 = SMI enabled 0011 = IRQ3 enabled 0100 = IRQ4 enabled 0101 = IRQ5 enabled 0100 = IRQ6 enabled 0111 = IRQ7 enabled 1000 = IRQ8 enabled 1001 = IRQ9 enabled 1010 = IRQ10 enabled 1011 = IRQ11 enabled 1100 = IRQ12 enabled 1101 = IRQ13 enabled 1110 = IRQ14 enabled 1111 = IRQ15 enabled

5.5 ExCA Card Status-Change Register

The ExCA card status-change register controls interrupt routing for I/O interrupts as well as other critical 16-bit PC Card functions. The register enables these interrupt sources to generate an interrupt to the host. When the interrupt source is disabled, the corresponding bit in this register always reads 0. When an interrupt source is enabled, the corresponding bit in this register is set to indicate that the interrupt source is active. After generating the interrupt to the host, the interrupt service routine must read this register to determine the source of the interrupt. The interrupt service routine is responsible for resetting the bits in this register as well. Resetting a bit is accomplished by one of two methods: a read of this register or an explicit write back of 1 to the status bit. The choice of these two methods is based on bit 2 (interrupt flag clear mode select) in the ExCA global control register (ExCA offset 1E/5E/81E, see Section 5.20). See Table 5–7 for a complete description of the register contents.

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

Register: **ExCA card status-change**
Offset: CardBus socket address + 804h; Card A ExCA offset 04h
Card B ExCA offset 44h
Type: Read-only
Default: 00h

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

BIT	SIGNAL	TYPE	FUNCTION
7–4	RSVD	R	Reserved. Bits 7–4 return 0s when read.
3	CDCHANGE	R	Card detect change. Bit 3 indicates whether a change on $\overline{CD1}$ or $\overline{CD2}$ occurred at the PC Card interface. This bit is encoded as: 0 = No change detected on either $\overline{CD1}$ or $\overline{CD2}$ 1 = Change detected on either $\overline{CD1}$ or $\overline{CD2}$
2	READYCHANGE	R	Ready change. When a 16-bit memory is installed in the socket, bit 2 includes whether the source of a PCI1520 interrupt was due to a change on READY at the PC Card interface, indicating that the PC Card is now ready to accept new data. This bit is encoded as: 0 = No low-to-high transition 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 PCI1520 interrupt was due to a battery-low warning condition. This bit is encoded as: 0 = No battery warning condition (default) 1 = Detected battery warning condition When a 16-bit I/O card is installed, bit 1 is always 0.
0	BATDEAD	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 PCI1520 interrupt was due to a battery dead condition. This bit is encoded as: 0 = $\overline{STSCHG}$ deasserted (default) 1 = $\overline{STSCHG}$ asserted Ring indicate. When the PCI1520 is configured for ring indicate operation, bit 0 indicates the status of $\overline{RI}$.

5.6 ExCA Card Status-Change Interrupt Configuration Register

The ExCA card status-change interrupt configuration register controls interrupt routing for card status-change interrupts, as well as masking CSC interrupt sources. See Table 5–8 for a complete description of the register contents.

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

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

Table 5–8. ExCA Card Status-Change Interrupt Configuration Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–4	CSCSELECT	RW	Interrupt select for card status change. Bits 7–4 select the interrupt routing for card status-change interrupts. 0000 = CSC interrupts routed to PCI interrupts if bit 5 (CSC) of the diagnostic register (PCI offset 93h, see Section 4.34) is set to 1. In this case bit 4 (CSCROUTE) of the ExCA interrupt and general control register (ExCA offset 03h/43h/803h, see Section 5.4) is a don't care. This is the default setting. 0000 = No ISA interrupt routing if bit 5 (CSC) of the diagnostic register is set to 0 (see Section 4.34). In this case, CSC interrupts are routed to PCI interrupts by setting bit 4 (CSCROUTE) of the ExCA interrupt and general control register (ExCA offset 03h/43h/803h, see Section 5.4) to 1. This field is encoded as: 0000 = No interrupt routing (default) 1000 = IRQ8 enabled 0001 = IRQ1 enabled 1001 = IRQ9 enabled 0010 = SMI enabled 1010 = IRQ10 enabled 0011 = IRQ3 enabled 1011 = IRQ11 enabled 0100 = IRQ4 enabled 1100 = IRQ12 enabled 0101 = IRQ5 enabled 1101 = IRQ13 enabled 0110 = IRQ6 enabled 1110 = IRQ14 enabled 0111 = IRQ7 enabled 1111 = IRQ15 enabled
3	CDEN	RW	Card detect enable. Bit 3 enables interrupts on $\overline{CD1}$ or $\overline{CD2}$ changes. This bit is encoded as: 0 = Disables interrupts on $\overline{CD1}$ or $\overline{CD2}$ line changes (default) 1 = Enables interrupts on $\overline{CD1}$ or $\overline{CD2}$ line changes
2	READYEN	RW	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	RW	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	RW	Battery dead enable. Bit 0 enables/disables a battery dead condition on a memory PC Card or assertion of the $\overline{STSCHG}$ I/O PC Card signal to generate a CSC interrupt. 0 = Disables host interrupt generation (default) 1 = Enables host interrupt generation

5.7 ExCA Address Window Enable Register

The ExCA address window enable register enables/disables the memory and I/O windows to the 16-bit PC Card. By default, all windows to the card are disabled. The PCI1520 does not acknowledge PCI memory or I/O cycles to the card if the corresponding enable bit in this register is 0, regardless of the programming of the memory or I/O window start/end/offset address registers. See Table 5–9 for a complete description of the register contents.

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

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

Table 5–9. ExCA Address Window Enable Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	IOWIN1EN	RW	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	RW	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 returns 0 when read.
4	MEMWIN4EN	RW	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	RW	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	RW	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	RW	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	RW	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

5.8 ExCA I/O Window Control Register

The ExCA I/O window control register contains parameters related to I/O window sizing and cycle timing. See Table 5–10 for a complete description of the register contents.

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

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

Table 5–10. ExCA I/O Window Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	WAITSTATE1	RW	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	RW	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	RW	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	RW	I/O window 1 data size. Bit 4 controls the I/O window 1 data size. Bit 4 is ignored if bit 5 (IOSIS16W1) is set. This bit is encoded as: 0 = Window data width is 8 bits (default). 1 = Window data width is 16 bits.
3	WAITSTATE0	RW	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	RW	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	RW	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 IOIS16.
0	DATASIZE0	RW	I/O window 0 data size. Bit 0 controls the I/O window 0 data size. Bit 0 is ignored if bit 1 (IOSIS16W0) is set. This bit is encoded as: 0 = Window data width is 8 bits (default). 1 = Window data width is 16 bits.

5.9 ExCA I/O Windows 0 and 1 Start-Address Low-Byte Registers

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

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

5.10 ExCA I/O Windows 0 and 1 Start-Address High-Byte Registers

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

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

5.11 ExCA I/O Windows 0 and 1 End-Address Low-Byte Registers

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

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

5.12 ExCA I/O Windows 0 and 1 End-Address High-Byte Registers

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

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

5.13 ExCA Memory Windows 0–4 Start-Address Low-Byte Registers

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

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

5.14 ExCA Memory Windows 0–4 Start-Address High-Byte Registers

These registers contain the high nibble of the 16-bit memory window start address for memory windows 0, 1, 2, 3, and 4. The lower 4 bits of these registers correspond to bits A23–A20 of the start address. In addition, the memory window data width and wait states are set in this register. See Table 5–11 for a complete description of the register contents.

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

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

BIT	SIGNAL	TYPE	FUNCTION
7	DATASIZE	RW	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	RW	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	RW	Scratch pad bits. Bits 5 and 4 have no effect on memory window operation.
3–0	STAHN	RW	Start-address high nibble. Bits 3–0 represent the upper address bits A23–A20 of the memory window start address.

5.15 ExCA Memory Windows 0–4 End-Address Low-Byte Registers

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

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

5.16 ExCA Memory Windows 0–4 End-Address High-Byte Registers

These registers contain the high nibble of the 16-bit memory window end address for memory windows 0, 1, 2, 3, and 4. The lower 4 bits of these registers correspond to bits A23–A20 of the end address. In addition, the memory window wait states are set in this register. See Table 5–12 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	ExCA memory windows 0–4 end-address high-byte							
Type	RW	RW	R	R	RW	RW	RW	RW
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
Default: 00h

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

BIT	SIGNAL	TYPE	FUNCTION
7–6	MEMWS	RW	Wait state. Bits 7 and 6 specify the number of equivalent ISA wait states to be added to 16-bit memory accesses. The number of wait states added is equal to the binary value of these two bits.
5–4	RSVD	R	Reserved. Bits 5 and 4 return 0s when read.
3–0	ENDHN	RW	End-address high nibble. Bits 3–0 represent the upper address bits A23–A20 of the memory window end address.

5.17 ExCA Memory Windows 0–4 Offset-Address Low-Byte Registers

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

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

5.18 ExCA Memory Windows 0–4 Offset-Address High-Byte Registers

These registers contain the high 6 bits of the 16-bit memory window offset address for memory windows 0, 1, 2, 3, and 4. The lower 6 bits of these registers correspond to bits A25–A20 of the offset address. In addition, the write protection and common/attribute memory configurations are set in this register. See Table 5–13 for a complete description of the register contents.

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

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

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

BIT	SIGNAL	TYPE	FUNCTION
7	WINWP	RW	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	RW	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	RW	Offset-address high byte. Bits 5–0 represent the upper address bits A25–A20 of the memory window offset address.

5.19 ExCA Card Detect and General Control Register

The ExCA card detect and general control register controls how the ExCA registers for the socket respond to card removal, as well as reports the status of $\overline{VS1}$ and $\overline{VS2}$ at the PC Card interface. See Table 5–14 for a complete description of the register contents.

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

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

Table 5–14. ExCA Card Detect and General Control Register Description

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	RW	Software card detect interrupt. If bit 3 (CDEN) in the ExCA card status-change interrupt configuration register (ExCA offset 05h/45h/805, see Section 5.6) is set, then writing a 1 to bit 5 causes a card-detect card-status change interrupt for the associated card socket. If bit 3 (CDEN) in the ExCA card status-change-interrupt configuration register (ExCA offset 05h/45h/805, see Section 5.6) is cleared to 0, then writing a 1 to bit 5 has no effect. A read operation of this bit always returns 0.
4	CDRESUME	RW	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, RI_OUT goes from high to low. RI_OUT remains low until bit 0 (card status change) in the ExCA card status-change register is cleared (see Section 5.5). 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 and 2 return 0s when read.
1	REGCONFIG	RW	Register configuration on card removal. Bit 1 controls how the ExCA registers for the socket react to a card removal event. This bit is encoded as: 0 = No change to ExCA registers on card removal (default) 1 = Reset ExCA registers on card removal
0	RSVD	R	Reserved. Bit 0 returns 0 when read.

5.20 ExCA Global Control Register

The ExCA global control 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. See Table 5–15 for a complete description of the register contents.

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

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

Table 5–15. ExCA Global Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7–5	RSVD	R	Reserved. Bits 7–5 return 0s when read.
4	INTMODEB	RW	Level/edge interrupt mode select – card B. Bit 4 selects the signaling mode for the PCI1520 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	RW	Level/edge interrupt mode select – card A. Bit 3 selects the signaling mode for the PCI1520 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	RW	Interrupt flag clear mode select. Bit 2 selects the interrupt flag clear mechanism for the flags in the ExCA card status change register (ExCA offset 04h/44h/804h, see Section 5.5). This bit is encoded as: 0 = Interrupt flags are cleared by read of CSC register (default). 1 = Interrupt flags are cleared by explicit writeback of 1.
1	CSCMODE	RW	Card status change level/edge mode select. Bit 1 selects the signaling mode for the PCI1520 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	RW	Power-down mode select. When bit 0 is set to 1, the PCI1520 is in power-down mode. In power-down mode, the PCI1520 card outputs are high-impedance until an active cycle is executed on the card interface. Following an active cycle, the outputs are again high-impedance. The PCI1520 still receives functional interrupts and/or card status-change interrupts; however, an actual card access is required to wake up the interface. This bit is encoded as: 0 = Power-down mode is disabled (default). 1 = Power-down mode is enabled.

5.21 ExCA I/O Windows 0 and 1 Offset-Address Low-Byte Registers

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

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

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

5.22 ExCA I/O Windows 0 and 1 Offset-Address High-Byte Registers

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

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

5.23 ExCA Memory Windows 0–4 Page Registers

The upper 8 bits of a 4-byte PCI memory address are compared to the contents of this register when decoding addresses for 16-bit memory windows. Each window has its own page register, all of which default to 00h. By programming this register to a nonzero value, host software can locate 16-bit memory windows in any 1 of 256 16-Mbyte regions in the 4-Gbyte PCI address space. These registers are only accessible when the ExCA registers are memory-mapped; that is, these registers cannot be accessed using the index/data I/O scheme.

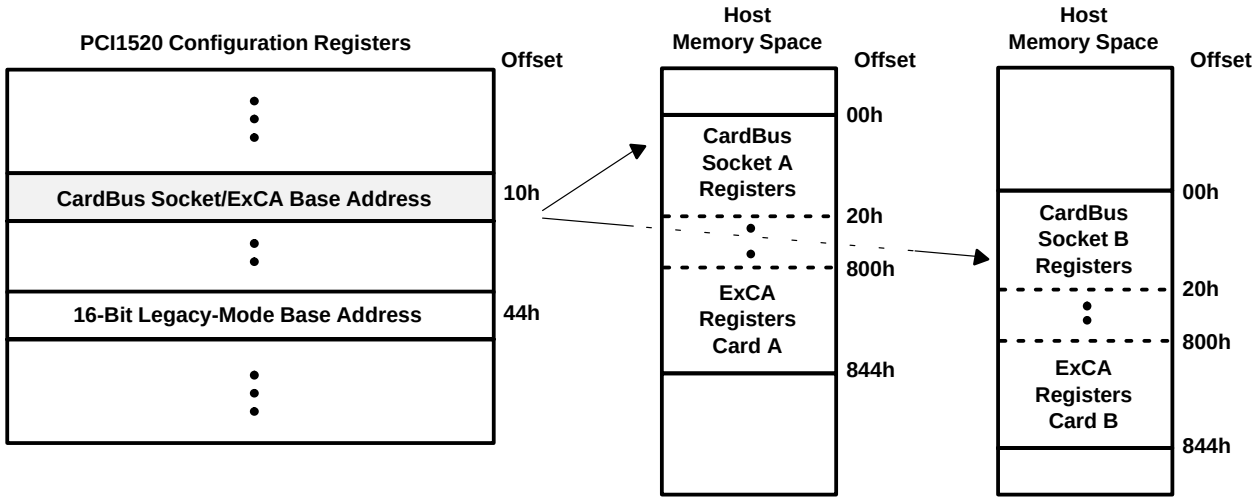
Bit	7	6	5	4	3	2	1	0
Name	ExCA memory windows 0–4 page							
Type	RW	RW	RW	RW	RW	RW	RW	RW
Default	0	0	0	0	0	0	0	0

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

6 CardBus Socket Registers (Functions 0 and 1)

The *1997 PC Card Standard* requires a CardBus socket controller to provide five 32-bit registers that report and control socket-specific functions. The PCI1520 provides the CardBus socket/ExCA base-address register (PCI offset 10h, see Section 4.12) to locate these CardBus socket registers in PCI memory address space. Each socket has a separate base address register for accessing the CardBus socket registers (see Figure 6–1). Table 6–1 gives the location of the socket registers in relation to the CardBus socket/ExCA base address.

The PCI1520 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 6–1. Accessing CardBus Socket Registers Through PCI Memory

Table 6–1. CardBus Socket Registers

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

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

6.1 Socket Event Register

The socket event register indicates a change in socket status has occurred. These bits do not indicate what the change is, only that one has occurred. Software must read the socket present-state register (CB offset 08h, see Section 6.3) for current status. Each bit in this register can be cleared by writing a 1 to that bit. The bits in this register can be set to a 1 by software by writing a 1 to the corresponding bit in the socket force event register (CB offset 0Ch, see Section 6.4). 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 (that is, $\overline{\text{CSTSCHG}}$ reasserted or card detect is still true). Software must clear this register before enabling interrupts. If it is not cleared when interrupts are enabled, then an interrupt is generated (but not masked) based on any bit set. See Table 6–2 for a complete description of the register contents.

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

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

Table 6–2. Socket Event Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–4	RSVD	R	Reserved. Bits 31–4 return 0s when read.
3	PWREVENT	R/C	Power cycle. Bit 3 is set when the PCI1520 detects that bit 3 (PWRCYCLE) in the socket present-state register (CB offset 08h, see Section 6.3) has changed state. This bit is cleared by writing a 1.
2	CD2EVENT	R/C	$\overline{\text{CCD2}}$. Bit 2 is set when the PCI1520 detects that bit 2 (CDETECT2) in the socket present-state register (CB offset 08h, see Section 6.3) has changed state. This bit is cleared by writing a 1.
1	CD1EVENT	R/C	$\overline{\text{CCD1}}$. Bit 1 is set when the PCI1520 detects that bit 1 (CDETECT1) in the socket present-state register (CB offset 08h, see Section 6.3) has changed state. This bit is cleared by writing a 1.
0	CSTSEVENT	R/C	$\overline{\text{CSTSCHG}}$. Bit 0 is set when bit 0 (CARDSTS) in the socket present-state register (CB offset 08h, see Section 6.3) has changed state. For CardBus cards, bit 0 is set on the rising edge of $\overline{\text{CSTSCHG}}$. For 16-bit PC Cards, bit 0 is set on both transitions of $\overline{\text{CSTSCHG}}$. This bit is reset by writing a 1.

6.2 Socket Mask Register

The socket mask register allows software to control the CardBus card events that generate a status change interrupt. The state of these mask bits does not prevent the corresponding bits from reacting in the socket event register (CB offset 00h, see Section 6.1). See Table 6–3 for a complete description of the register contents.

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

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

Table 6–3. Socket Mask Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–4	RSVD	R	Reserved. Bits 31–4 return 0s when read.
3	PWRMASK	RW	Power cycle. Bit 3 masks bit 3 (PWRCYCLE) in the socket present-state register (CB offset 08h, see Section 6.3) from causing a status change interrupt. 0 = PWRCYCLE event does not cause CSC interrupt (default). 1 = PWRCYCLE event causes CSC interrupt.
2–1	CDMASK	RW	Card detect mask. Bits 2 and 1 mask bits 1 and 2 (CDETECT1 and CDETECT2) in the socket present-state register (CB offset 08h, see Section 6.3) from causing a CSC interrupt. 00 = Insertion/removal does not cause CSC interrupt (default). 01 = Reserved (undefined) 10 = Reserved (undefined) 11 = Insertion/removal causes CSC interrupt.
0	CSTSMASK	RW	$\overline{\text{CSTSCHG}}$ mask. Bit 0 masks bit 0 (CARDSTS) in the socket present-state register (CB offset 08h, see Section 6.3) from causing a CSC interrupt. 0 = CARDSTS event does not cause CSC interrupt (default). 1 = CARDSTS event causes CSC interrupt.

6.3 Socket Present-State Register

The socket present-state register reports information about the socket interface. Write transactions to the socket force event register (CB offset 0Ch, see Section 6.4) are reflected here, as well as general socket interface status. Information about PC Card V_{CC} support and card type is only updated at each insertion. Also note that the PCI1520 uses $\overline{CCD1}$ and $\overline{CCD2}$ during card identification, and changes on these signals during this operation are not reflected in this register. See Table 6–4 for a complete description of the register contents.

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

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

Table 6–4. Socket Present-State Register Description

BIT	SIGNAL	TYPE	FUNCTION
31	YVSOCKET	R	YV socket. Bit 31 indicates whether or not the socket can supply V_{CC} = Y.Y V to PC Cards. The PCI1520 does not support Y.Y-V V_{CC}; therefore, this bit is always reset unless overridden by the socket force event register (CB offset 0Ch, see Section 6.4). 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 PCI1520 does not support X.X-V V_{CC}; therefore, this bit is always reset unless overridden by the socket force event register (CB offset 0Ch, see Section 6.4). 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 PCI1520 does support 3.3-V V_{CC}; therefore, this bit is always set unless overridden by the socket force event register (CB offset 0Ch, see Section 6.4).
28	5VSOCKET	R	5-V socket. Bit 28 indicates whether or not the socket can supply V_{CC} = 5 V to PC Cards. The PCI1520 does support 5-V V_{CC}; therefore, this bit is always set unless overridden by the socket force event register (CB offset 0Ch, see Section 6.4).
27	ZVSUPPORT	R	Zoomed-video support. This bit indicates whether or not the socket has support for zoomed video. 0 = Zoomed video is not supported. 1 = Zoomed video is supported.
26–14	RSVD	R	Reserved. Bits 27–14 return 0s when read.
13	YVCARD	R	YV card. Bit 13 indicates whether or not the PC Card inserted in the socket supports V_{CC} = Y.Y V. 0 = Y.Y-V V_{CC} is not supported. 1 = Y.Y-V V_{CC} is supported.
12	XVCARD	R	XV card. Bit 12 indicates whether or not the PC Card inserted in the socket supports V_{CC} = X.X V. 0 = X.X-V V_{CC} is not supported. 1 = X.X-V V_{CC} is supported.
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. 0 = 3.3-V V_{CC} is not supported. 1 = 3.3-V V_{CC} is supported.

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

BIT	SIGNAL	TYPE	FUNCTION
10	5VCARD	R	5-V card. Bit 10 indicates whether or not the PC Card inserted in the socket supports V_{CC} = 5 V. 0 = 5-V V_{CC} is not supported. 1 = 5-V V_{CC} is supported.
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 PCI1520. 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	Power cycle. Bit 3 indicates 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	$\overline{\text{CSTSCHG}}$. Bit 0 reflects the current status of $\overline{\text{CSTSCHG}}$ at the PC Card interface. 0 = $\overline{\text{CSTSCHG}}$ low 1 = $\overline{\text{CSTSCHG}}$ high

6.4 Socket Force Event Register

The socket force event register is used to force changes to the socket event register (CB offset 00h, see Section 6.1) and the socket present-state register (see Section 6.3). Bit 14 (CVSTEST) in this register must be written when forcing changes that require card interrogation. See Table 6–5 for a complete description of the register contents.

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

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

Table 6–5. Socket Force Event Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–28	RSVD	R	Reserved. Bits 31–28 return 0s when read.
27	FZVSUPPORT	W	Zoomed-video support. This bit indicates whether or not the socket has support for zoomed video.
26–15	RSVD	R	Reserved. Bits 26–15 return 0s when read.
14	CVSTEST	W	Card VS test. When bit 14 is set, the PCI1520 re-interrogates the PC Card, updates the socket present-state register (CB offset 08h, see Section 6.3), and enables the socket control register (CB offset 10h, see Section 6.5).
13	FYVCARD	W	Force YV card. Write transactions to bit 13 cause bit 13 (YVCARD) in the socket present-state register (CB offset 08h, see Section 6.3) to be written. When set, this bit disables the socket control register (CB offset 10h, see Section 6.5).
12	FXVCARD	W	Force XV card. Write transactions to bit 12 cause bit 12 (XVCARD) in the socket present-state register (CB offset 08h, see Section 6.3) to be written. When set, this bit disables the socket control register (CB offset 10h, see Section 6.5).
11	F3VCARD	W	Force 3-V card. Write transactions to bit 11 cause bit 11 (3VCARD) in the socket present-state register (CB offset 08h, see Section 6.3) to be written. When set, this bit disables the socket control register (CB offset 10h, see Section 6.5).
10	F5VCARD	W	Force 5-V card. Write transactions to bit 10 cause bit 10 (5VCARD) in the socket present-state register (CB offset 08h, see Section 6.3) to be written. When set, this bit disables the socket control register (CB offset 10h, see Section 6.5).
9	FBADVCCREQ	W	Force bad V_{CC} request. Changes to bit 9 (BADVCCREQ) in the socket present-state register (CB offset 08h, see Section 6.3) can be made by writing to bit 9.
8	FDATALOST	W	Force data lost. Write transactions to bit 8 cause bit 8 (DATALOST) in the socket present-state register (CB offset 08h, see Section 6.3) to be written.
7	FNOTACARD	W	Force not-a-card. Write transactions to bit 7 cause bit 7 (NOTACARD) in the socket present-state register (CB offset 08h, see Section 6.3) to be written.
6	RSVD	R	Reserved. Bit 6 returns 0 when read.
5	FCBCARD	W	Force CardBus card. Write transactions to bit 5 cause bit 5 (CBCARD) in the socket present-state register (CB offset 08h, see Section 6.3) to be written.
4	F16BITCARD	W	Force 16-bit card. Write transactions to bit 4 cause bit 4 (16BITCARD) in the socket present-state register (CB offset 08h, see Section 6.3) to be written.
3	FPWRCYCLE	W	Force power cycle. Write transactions to bit 3 cause bit 3 (PWREVENT) in the socket event register (CB offset 00h, see Section 6.1) to be written, and bit 3 (PWRCYCLE) in the socket present-state register (CB offset 08h, see Section 6.3) is unaffected.
2	FCDETECT2	W	Force $\overline{\text{CCD2}}$. Write transactions to bit 2 cause bit 2 (CD2EVENT) in the socket event register (CB offset 00h, see Section 6.1) to be written, and bit 2 (CDETECT2) in the socket present-state register (CB offset 08h, see Section 6.3) is unaffected.
1	FCDETECT1	W	Force $\overline{\text{CCD1}}$. Write transactions to bit 1 cause bit 1 (CD1EVENT) in the socket event register (CB offset 00h, see Section 6.1) to be written, and bit 1 (CDETECT1) in the socket present-state register (CB offset 08h, see Section 6.3) is unaffected.
0	FCARDSTS	W	Force CSTSCHG. Write transactions to bit 0 cause bit 0 (CSTSEVENT) in the socket event register (CB offset 00h, see Section 6.1) to be written, and bit 0 (CARDSTS) in the socket present-state register (CB offset 08h, see Section 6.3) is unaffected.

6.5 Socket Control Register

The socket control register provides control of the voltages applied to the socket and instructions for the CB $\overline{\text{CLKRUN}}$ protocol. The PCI1520 ensures that the socket is powered up only at acceptable voltages when a CardBus card is inserted. See Table 6–6 for a complete description of the register contents.

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

Register: **Socket control**
Offset: CardBus socket address + 10h
Type: Read-only, Read/Write
Default: 0000 0400h

Table 6–6. Socket Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–12	RSVD	R	Reserved. These bits return 0 when read. A write to these bits has no effect.
11	ZV_ACTIVITY	R	Zoomed video activity. This bit returns 0 when the ZVEN bits for both sockets are 0 (disabled). If either ZVEN bit is set to 1, the ZV_ACTIVITY bit returns 1.
10	STDZVREG	R	Standardized zoomed video register model support. This bit returns 1 by default when the STDZVEN bit (bit 0) in the diagnostic register is cleared (PCI offset 93h, see Section 4.34).
9	ZVEN	RW	Zoomed video enable. This bit enables zoomed video for this socket.
8	RSVD	R	Reserved. ThIs bit returns 0 when read. A write to thIs bit has no effect.
7	STOPCLK	RW	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	RW	V_{CC} control. Bits 6–4 request card V_{CC} changes. 000 = Request power off (default) 100 = Request V_{CC} = X.X V 001 = Reserved 101 = Request V_{CC} = Y.Y V 010 = Request V_{CC} = 5 V 110 = Reserved 011 = Request V_{CC} = 3.3 V 111 = Reserved
3	RSVD	R	Reserved. Bit 3 returns 0 when read.
2–0	VPPCTRL	RW	V_{PP} control. Bits 2–0 request card V_{PP} changes. 000 = Request power off (default) 100 = Request V_{PP} = X.X V 001 = Request V_{PP} = 12 V 101 = Request V_{PP} = Y.Y V 010 = Request V_{PP} = 5 V 110 = Reserved 011 = Request V_{PP} = 3.3 V 111 = Reserved

6.6 Socket Power-Management Register

This register provides power management control over the socket through a mechanism for slowing or stopping the clock on the card interface when the card is idle. See Table 6–7 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Socket power-management															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
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	RW
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Table 6–7. Socket Power-Management Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–26	RSVD	R	Reserved. Bits 31–26 return 0s when read.
25	SKTACCES	R	Socket access status. This bit 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 23–17 return 0s when read.
16	CLKCTRLEN	RW	CardBus clock control enable. When bit 16 is set, bit 0 (CLKCTRL) is enabled. 0 = Clock control is disabled (default). 1 = Clock control is enabled.
15–1	RSVD	R	Reserved. Bits 15–1 return 0s when read.
0	CLKCTRL	RW	CardBus clock control. This bit determines whether the CB $\overline{\text{CLKRUN}}$ protocol stops or slows the CB clock during idle states. Bit 16 (CLKCTRLEN) enables this bit. 0 = Allows CB $\overline{\text{CLKRUN}}$ protocol to stop the CB clock (default). 1 = Allows CB $\overline{\text{CLKRUN}}$ protocol to slow the CB clock by a factor of 16.

7 Electrical Characteristics

7.1 Absolute Maximum Ratings Over Operating Temperature Ranges†

Supply voltage range, V_{CC} . . . . . –0.5 V to 4.6 V
Clamping voltage range, V_{CCP}, V_{CCA}, V_{CCB} . . . . . –0.5 V to 6 V
Input voltage range, V_I: PCI, miscellaneous . . . . . –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
Fail safe . . . . . –0.5 V to V_{CC} + 0.5 V
Output voltage range, V_O: PCI, miscellaneous . . . . . –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
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

† 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 and miscellaneous terminals are measured with respect to V_{CCP} instead of V_{CC}. PC Card terminals are measured with respect to V_{CCA} or V_{CCB}. The limit specified applies for a dc condition.

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

7.2 Recommended Operating Conditions (see Note 3)

			OPERATION	MIN	NOM	MAX	UNIT
V_{CC}	Core voltage	Commercial	3.3 V	3	3.3	3.6	V
V_{CCP}	PCI and miscellaneous I/O clamp voltage	Commercial	3.3 V	3	3.3	3.6	V
			5 V	4.75	5	5.25	
V_{CCA} V_{CCB}	PC Card I/O clamp voltage	Commercial	3.3 V	3	3.3	3.6	V
			5 V	4.75	5	5.25	
V_{IH}†	High-level input voltage	PCI	3.3 V	0.5 V_{CCP}		V_{CCP}	V
			5 V	2		V_{CCP}	
		PC Card	3.3 V	0.475 $V_{CC(A/B)}$		$V_{CC(A/B)}$	
			5 V	2.4		$V_{CC(A/B)}$	
		Miscellaneous‡		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 $V_{CC(A/B)}$	
			5 V	0		0.8	
		Miscellaneous‡		0		0.8	
V_I	Input voltage	PCI		0		V_{CCP}	V
		PC Card		0		$V_{CC(A/B)}$	
		Miscellaneous‡		0		V_{CC}	
V_O§	Output voltage	PCI		0		V_{CC}	V
		PC Card		0		V_{CC}	
		Miscellaneous‡		0		V_{CC}	
t_t	Input transition time (t_r and t_f)	PCI and PC Card		1		4	ns
		Miscellaneous‡		0		6	
T_A	Operating ambient temperature range	PCI1520I		–40	25	85	°C

† Applies to external inputs and bidirectional buffers without hysteresis

‡ Miscellaneous terminals are C11, C15, G18, H05, J15, L18, P07, P09, U08, and U11 for the GHK packaged device ($\overline{\text{SUSPEND}}$, $\overline{\text{GRST}}$, $\overline{\text{CDx}}$, and $\overline{\text{VSx}}$ terminals).

§ Applies to external output buffers

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

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

PARAMETER		TERMINALS	OPERATION	TEST CONDITIONS	MIN	MAX	UNIT
V_{OH}	High-level output voltage	PCI	3.3 V	I_{OH} = –0.5 mA	0.9 V_{CC}		V
			5 V	I_{OH} = –2 mA	2.4		
		SPKROUT	3.3 V	I_{OH} = –0.5 mA	0.9 V_{CC}		
			5 V	I_{OH} = –1 mA	2.4		
		PC Card	3.3 V	I_{OH} = –0.15 mA	0.9 V_{CC}		V
			5 V	I_{OH} = –0.15 mA	2.4		
		Miscellaneous		I_{OH} = –4 mA	V_{CC}–0.6		
V_{OL}	Low-level output voltage	PCI	3.3 V	I_{OL} = 1.5 mA		0.1 V_{CC}	V
			5 V	I_{OL} = 6 mA		0.55	
		PC Card	3.3 V	I_{OL} = 0.7 mA		0.1 V_{CC}	
			5 V	I_{OL} = 0.7 mA		0.55	
		Miscellaneous		I_{OL} = 4 mA		0.5	
		SPKROUT	3.3 V	I_{OL} = 1 mA		0.1 V_{CC}	
			5 V	I_{OL} = 1 mA		0.55	
I_{OZL}	High-impedance, low-level output current	Output terminals	3.6 V	V_I = V_{CC}		–1	µA
			5.25 V	V_I = V_{CC}		–1	
I_{OZH}	High-impedance, high-level output current	Output terminals	3.6 V	V_I = V_{CC}†		10	µA
			5.25 V	V_I = V_{CC}†		25	
I_{IL}	Low-level input current	Input terminals		V_I = GND		–1	µA
		I/O terminals		V_I = GND		–10	
		Pullup terminals		V_I = GND		–330	
I_{IH}	High-level input current	Input terminals	3.6 V	V_I = V_{CC}‡		10	µA
			5.25 V	V_I = V_{CC}‡		20	
		I/O terminals	3.6 V	V_I = V_{CC}‡		10	
			5.25 V	V_I = V_{CC}‡		25	

† For PCI and miscellaneous terminals, V_I = V_{CCP}. For PC Card terminals, V_I = $V_{CC(A/B)}$.
‡ For I/O terminals, input leakage (I_{IL} and I_{IH}) includes I_{OZ} leakage of the disabled output.

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

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

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

This data manual 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 (t_{en}, t_{dis}) = delay time, t_{su} = setup time, and t_h = hold time.

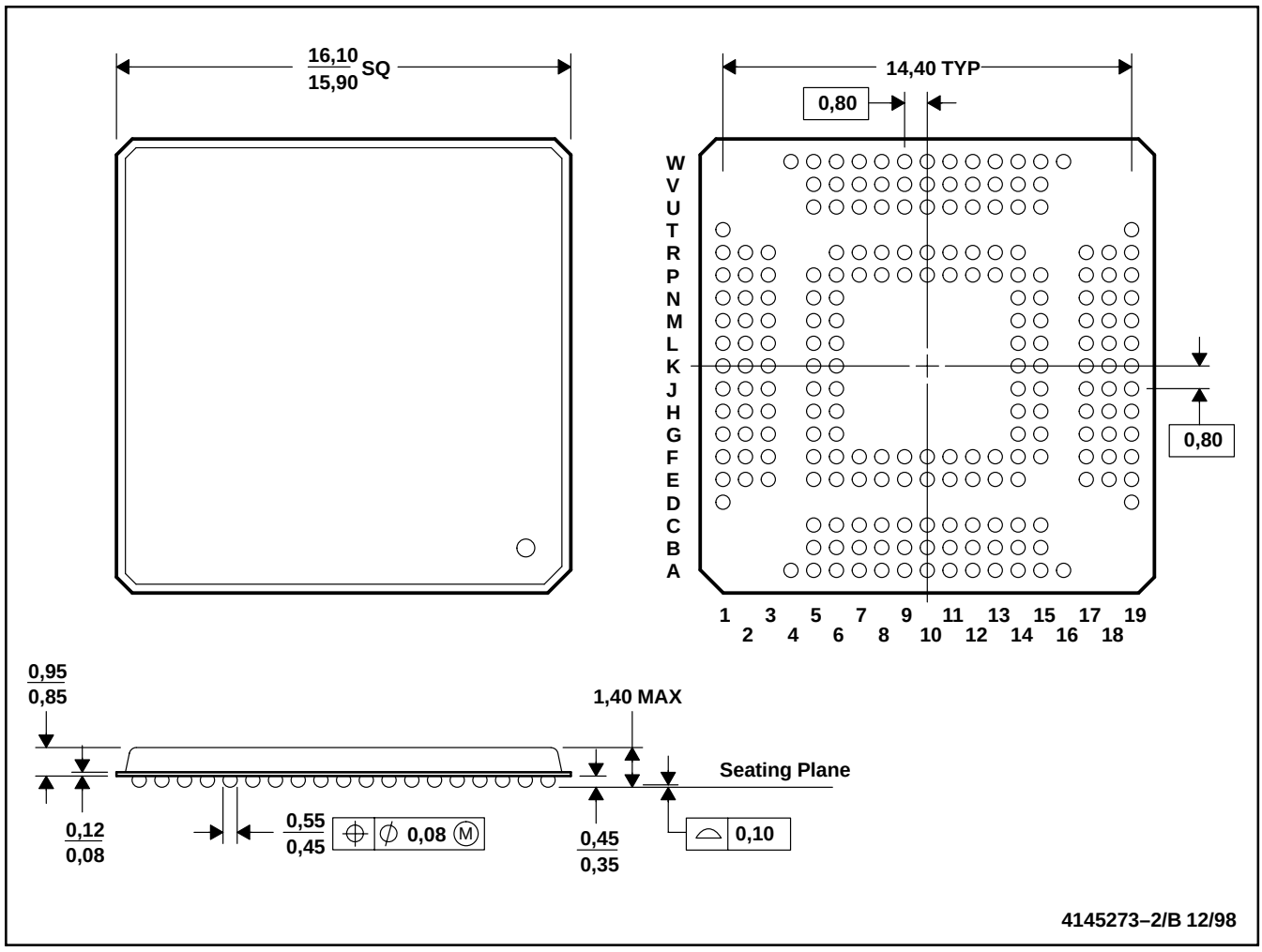
PARAMETER			ALTERNATE SYMBOL	TEST CONDITIONS	MIN	MAX	UNIT
t_{pd}	Propagation delay time, See Note 4	PCLK-to-shared signal valid delay time	t_{val}	C_L = 50 pF, See Note 4		11	ns
		PCLK-to-shared signal invalid delay time	t_{inv}		2		
t_{en}	Enable time, high impedance-to-active delay time from PCLK		t_{on}		2		ns
t_{dis}	Disable time, active-to-high impedance delay time from PCLK		t_{off}			28	ns
t_{su}	Setup time before PCLK valid		t_{su}		7		ns
t_h	Hold time after PCLK high		t_h		0		ns

NOTE 4: PCI shared signals are AD31–AD0, C/$\overline{BE3}$–C/$\overline{BE0}$, $\overline{FRAME}$, $\overline{TRDY}$, $\overline{IRDY}$, $\overline{STOP}$, IDSEL, $\overline{DEVSEL}$, and PAR.

8 Mechanical Information

The PCI1520 is packaged in a 209-ball GHK BGA package. The following shows the mechanical dimensions for the GHK package.

GHK (S-PBGA-N209) PLASTIC BALL GRID ARRAY



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

MicroStar BGA is a trademark of Texas Instruments.