Texas Instruments

PCI1620 PDV/GHK

Dual-Socket PC Card and UltraMedia Controller

Data Manual

August 2003

PCIBus Solutions

SCPS064C

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 PCI1620 is an integrated dual-socket PC Card controller, FlashMedia™ controller (SmartMedia™ Card, MultiMediaCard, SD Card, Memory Stick™ card) and Smart Card controller.

The PCI1620 UltraMedia™ controller is a three-function PCI device compliant with *PCI Local Bus Specification* 2.2. Functions 0 and 1 provide two independent PC Card socket controllers compliant with *PC Card Standard* 8.0. Function 2 is the interface to load the PCI1620 program RAM with firmware. The PCI1620 provides features that make it ideal for bridging between the PCI bus and PC Cards, and supports any combination of 16-bit, CardBus, and UltraMedia PC Cards in the two sockets, powered at 5 V, 3.3 V, or 1.8 V as required.

UltraMedia cards that comply with the latest PCMCIA standard provide for very low-cost flash media and Smart Card adapters, because the control logic is integrated into the PCI1620. The PCI1620 supports a passive 4-in-1 adapter, as well as active PC Card-style Flash media and Smart Card adapters.

No PCMCIA card or socket service software changes are required to move systems from an existing CardBus socket controller to the PCI1620. The FlashMedia UltraMedia applications use existing host ATA drivers, and Texas Instruments provides a qualified Smart Card driver for UltraMedia-based Smart Card adapters. The PCI1620 is register compatible with the Intel 82365SL–DF ExCA controller and implements the host interface defined in the *PC Card Standard*. The PCI1620 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 the pipeline architecture provide a high performance level with sustained bursting. The PCI1620 can be programmed to accept posted writes to improve bus utilization.

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

The PCI1620 is compliant with *PCI Bus Power Management Interface Specification* 1.1, and provides several low-power modes, which enable the host power system to further reduce power consumption. The PCI1620 also has a three-terminal serial interface compatible with both the TI TPS2226 and TPS2228 power switches.

1.2 Features

The PCI1620 supports the following features:

- *PC Card Standard* 8.0 compliant
- *PCI Bus Power Management Interface Specification* 1.1 compliant
- *Advanced Configuration and Power Interface Specification* 1.0 compliant
- *PCI Local Bus Specification* Revision 2.2 compliant
- *PC 98/99* compliant
- Has integrated voltage regulator to use 1.8-V core voltage
- Compliant with the *PCI Bus Interface Specification for PCI-to-CardBus Bridges*
- Advanced filtering on card detect lines provides 90 microseconds of noise immunity.
- Programmable D3 status terminal
- 1.8-V core logic and 3.3-V I/O cells with internal voltage regulator to generate 1.8-V core V_{CC}
- Universal PCI interfaces compatible with 3.3-V and 5-V PCI signaling environments
- Mix-and-match 5-V/3.3-V 16-bit PC Cards and 3.3-V CardBus cards
- Supports two PC Card or CardBus slots with hot insertion and removal

- Uses serial interface to TI TPS2226 and TI TPS2228 dual power switch
- Supports 132-MBps burst transfers to maximize data throughput on both the PCI bus and the CardBus bus
- Supports serialized IRQ with PCI interrupts
- 13 programmable multifunction terminals
- Interrupt modes supported: serial ISA/serial PCI, serial ISA/parallel PCI, parallel PCI only
- Serial EEPROM interface for loading subsystem ID and subsystem vendor ID
- Supports external zoomed video
- Dedicated terminal for PCI CLKRUN
- Four general-purpose event registers
- Multifunction PCI device with separate configuration space for each socket
- Five PCI memory windows and two I/O windows available to each 16-bit PC Card socket
- Two I/O windows and two memory windows available to each CardBus socket
- ExCA-compatible registers are mapped in memory or I/O space
- Intel™ 82365SL–DF register compatible
- Supports ring indicate, suspend, and PCI clock run
- Advanced submicron, low-power CMOS technology
- Provides VGA/palette memory and I/O, and subtractive decoding options
- LED activity terminals
- Supports PCI bus lock (LOCK)

1.3 Related Documents

- *PC Card Controller Device Class Power Management Reference Specification*
- *PC Card Standard* release 7
- *PCI Local Bus Specification* revision 2.2
- *PCI to PCMCIA CardBus Bridge Register Description* (Yenta), revision 2.1
- Texas Instruments TPS2226 and TPS2228 product data sheets
- SmartMedia Specifications, Issued 5/19/99
- MultiMediaCard Specification Version 2.2
- Multimedia Host Specification Version 3.7, Sandisk
- SD Memory Card Specifications, SD Group, 2000.
- Memory Stick Format Specification, Sony
- Memory Stick I/F Specification − Helen/Helious, TI Wireless Japan, Feb. 2000.
- ISO Standards for Identification Cards ISO/IEC 7816
- 3Soft M8052 MegaMacro™ Design Specification
- ANSI AT Attachment (ATA) Specification for Disk Drives x3.221−1994

1.4 Trademarks

Intel is a trademark of Intel Corporation.

MegaMacro is a trademark of MEJ Electronics Ltd., UK.

Memory Stick is a trademark of Sony Kabushiki Kaisha TA Sony Corporation, Japan.

MicroStar BGA and UltraMedia are trademarks of Texas Instruments.

SmartMedia is a trademark of Kabushiki Kaisha Toshiba DBA Toshiba Corporation, Japan.

SmartSocket is a trademark of ControlNet, Incorporated.

Other trademarks are the property of their respective owners.

1.5 Terms and Definitions

Terms and definitions used in this document are given in Table 1−1.

Table 1−1. Terms and Definitions

TERM	DEFINITIONS
ATA	AT (advanced technology, as in PC AT) attachment interface
ATA driver	An existing host software component that loads when a SmartMedia adapter and card is inserted into a PC Card socket. This driver is logically attached to a predefined CIS provided by the PCI1620 when the adapter and media are both inserted.
CIS	Card information structure. Tuple list defined by the PC Card standard to communicate card information to the host computer
CSR	Control and status register
Flash Media	SmartMedia, Memory Stick, or SD Flash operating in an ATA compatible mode
Function 2 firmware loader	A hardware element of the PCI1620 that provides a software interface to the TI firmware loader driver to load the program RAM with firmware
ISO/IEC 7816	The Smart Card standard
Magic Gate	A security technology for Memory Stick promoted and licensed by Sony
Memory Stick™	A small-form-factor flash interface that is defined, promoted, and licensed by Sony
MMC	MultiMediaCard. Specified by the MMC Association, and scope is encompassed by the SD Flash specification.
OHCI	Open host controller interface
PCMCIA	Personal Computer Memory Card International Association. Standards body that governs the PC Card standards
RSVD	Reserved for future use
SD Flash	Secure Digital Flash. Standard governed by the SD Association
Smart Card	The name applied to ID cards containing integrated circuits, as defined by ISO/IEC 7816-1
SmartMedia™	Also known as SSFDC, defined by Toshiba and governed by SSFDC Forum
SPI	Serial peripheral interface, a general-purpose synchronous serial interface. For more information, see the *Multimedia Card System Specification*, version 2.2.
SSFDC	Solid State Floppy Disk Card. The SSFDC Forum specifies SmartMedia
TI firmware loader driver	A qualified software component provided by Texas Instruments that loads the firmware into the PCI1620 on power up and initialization.
TI Smart Card driver	A qualified software component provided by Texas Instruments that loads when an UltraMedia-based Smart Card adapter is inserted into a PC Card slot. This driver is logically attached to a CIS provided by the PCI1620 when the adapter is inserted.
TI SmartSocket™ driver	A qualified software component provided by Texas Instruments that loads when an unsupported UltraMedia-based card is inserted into a PC Card slot. This driver serves to give the user a message that the inserted card is not supported.
UART	Universal asynchronous receiver and transmitter
UltraMedia™	*De facto* industry standard promoted by Texas Instruments that integrates CardBus, Smart Card, Memory Stick, MultiMediaCard/Secure Digital and SmartMedia functionality into one controller.

1.6 Ordering Information

ORDERING NUMBER	NAME	VOLTAGE	PACKAGE
PCI1620	PC Card, Cardbus card, Flash Media and Smart Card controller	3.3-V, 5-V tolerant I/Os	209-terminal GHK, 208-terminal PDV packages

2 Terminal Descriptions

The PCI1620 is available in a 209-terminal MicroStar BGA™ package (GHK), and in a 208-terminal plastic quad flatpack package (PDV).

2.1 Terminal Assignments for PCI1620

Figure 2−1 shows the terminal layout for the 209-terminal MicroStar BGA™ package (GHK). Figure 2−2 shows the terminal assignments for the 208-terminal quad flatpack (PDV) package.

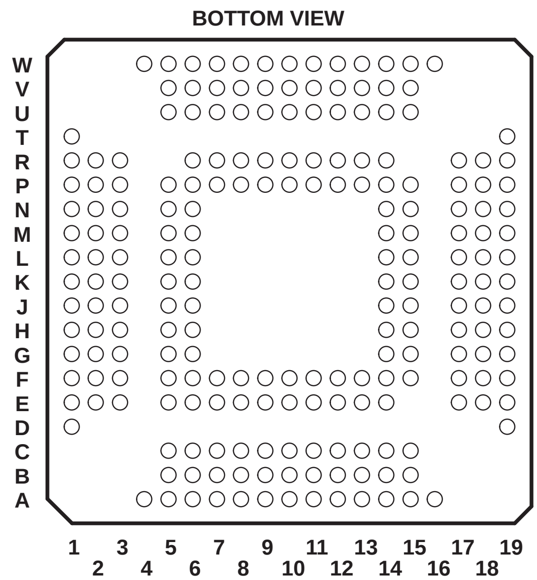


Figure 2−1. PCI1620 GHK Terminal Diagram

PDV LOW-PROFILE QUAD FLAT PACKAGE (LQFP)
TOP VIEW

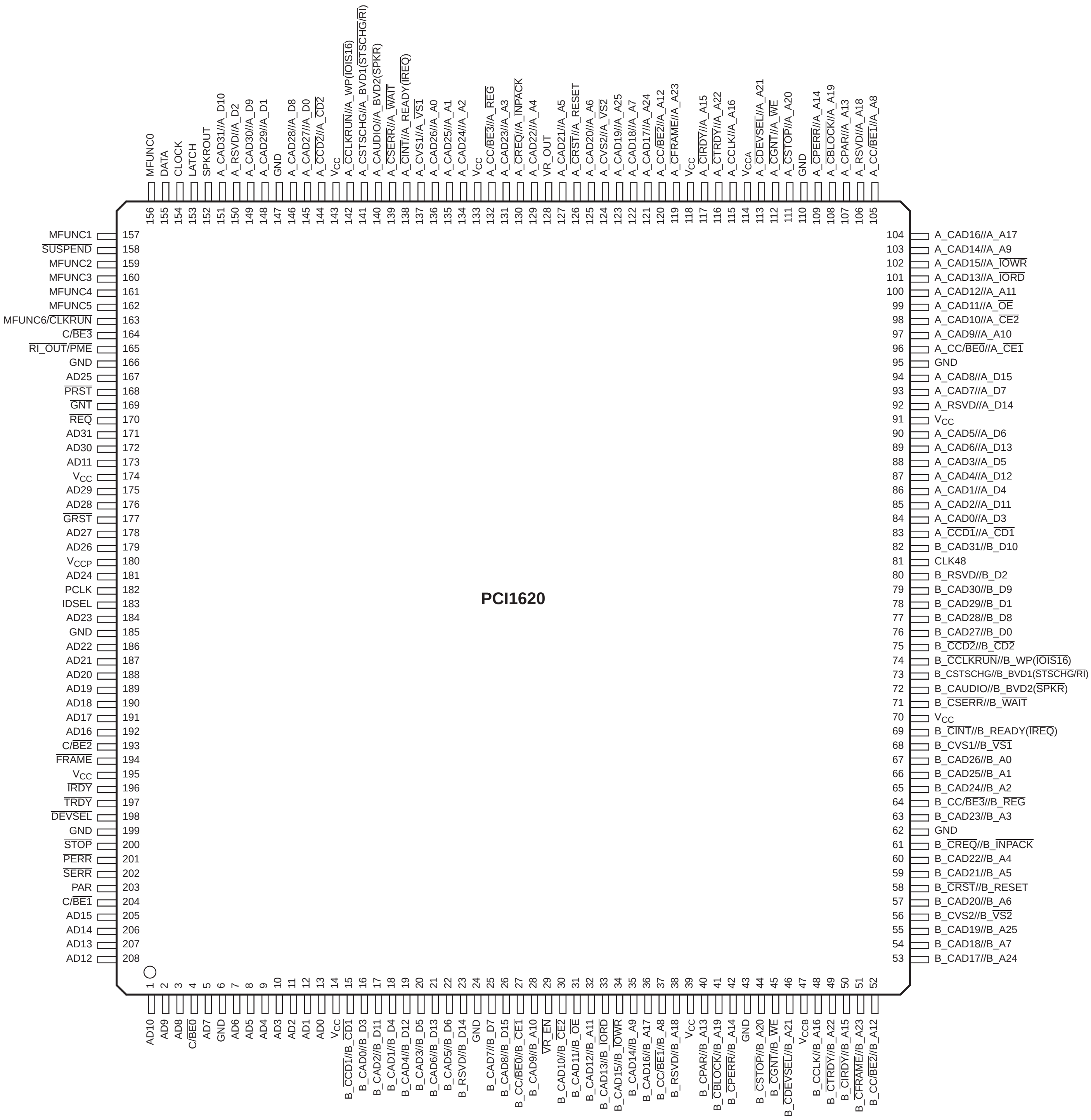


Figure 2−2. PCI1620 PDV Terminal Diagram

The following tables show the correspondence between signal names and their respective terminal assignments. In Table 2−1, PDV-package entries are listed in order by terminal number, with signal names for CardBus PC Cards and 16-bit PC Cards. In Table 2−2, GHK-package entries are listed in alphanumeric order by terminal number, with signal names for CardBus PC Cards and 16-bit PC Cards. In Table 2−3, entries are listed in alphanumeric order by CardBus PC Card signal names, with corresponding terminal numbers. In Table 2−4, entries are listed in alphanumeric order by 16-bit PC Card signal names, with corresponding terminal numbers.

Table 2−1. Signal Names by PDV Terminal Number

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CardBus PC Card	16-Bit PC Card		CardBus PC Card	16-Bit PC Card
1	AD10	AD10	41	B_CBLOCK	B_A19
2	AD9	AD9	42	B_CPERR	B_A14
3	AD8	AD8	43	GND	GND
4	C/BE0	C/BE0	44	B_CSTOP	B_A20
5	AD7	AD7	45	B_CGNT	B_WE
6	GND	GND	46	B_CDEVSEL	B_A21
7	AD6	AD6	47	V_{CCB}	V_{CCB}
8	AD5	AD5	48	B_CCLK	B_A16
9	AD4	AD4	49	B_CTRDY	B_A22
10	AD3	AD3	50	B_CIRDY	B_A15
11	AD2	AD2	51	B_CFRAME	B_A23
12	AD1	AD1	52	B_CC/BE2	B_A12
13	AD0	AD0	53	B_CAD17	B_A24
14	V_{CC}	V_{CC}	54	B_CAD18	B_A7
15	B_CCD1	B_CD1	55	B_CAD19	B_A25
16	B_CAD0	B_D3	56	B_CVS2	B_VS2
17	B_CAD2	B_D11	57	B_CAD20	B_A6
18	B_CAD1	B_D4	58	B_CRST	B_RESET
19	B_CAD4	B_D12	59	B_CAD21	B_A5
20	B_CAD3	B_D5	60	B_CAD22	B_A4
21	B_CAD6	B_D13	61	B_CREQ	B_INPACK
22	B_CAD5	B_D6	62	GND	GND
23	B_RSVD	B_D14	63	B_CAD23	B_A3
24	GND	GND	64	B_CC/BE3	B_REG
25	B_CAD7	B_D7	65	B_CAD24	B_A2
26	B_CAD8	B_D15	66	B_CAD25	B_A1
27	B_CC/BE0	B_CE1	67	B_CAD26	B_A0
28	B_CAD9	B_A10	68	B_CVS1	B_VS1
29	VR_EN	VR_EN	69	B_CINT	B_READY(IREQ)
30	B_CAD10	B_CE2	70	V_{CC}	V_{CC}
31	B_CAD11	B_OE	71	B_CSERR	B_WAIT
32	B_CAD12	B_A11	72	B_CAUDIO	B_BVD2(SPKR)
33	B_CAD13	B_IORD	73	B_CSTSCHG	B_BVD1(STSCHG/RI)
34	B_CAD15	B_IOWR	74	B_CCLKRUN	B_WP(IOIS16)
35	B_CAD14	B_A9	75	B_CCD2	B_CD2
36	B_CAD16	B_A17	76	B_CAD27	B_D0
37	B_CC/BE1	B_A8	77	B_CAD28	B_D8
38	B_RSVD	B_A18	78	B_CAD29	B_D1
39	V_{CC}	V_{CC}	79	B_CAD30	B_D9
40	B_CPAR	B_A13	80	B_RSVD	B_D2

Table 2−1. Signal Names by PDV Terminal Number (Continued)

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CardBus PC Card	16-Bit PC Card		CardBus PC Card	16-Bit PC Card
81	CLK48	CLK48	123	A_CAD19	A_A25
82	B_CAD31	B_D10	124	A_CVS2	A_VS2
83	A_CCD1	A_CD1	125	A_CAD20	A_A6
84	A_CAD0	A_D3	126	A_CRST	A_RESET
85	A_CAD2	A_D11	127	A_CAD21	A_A5
86	A_CAD1	A_D4	128	VR_OUT	VR_OUT
87	A_CAD4	A_D12	129	A_CAD22	A_A4
88	A_CAD3	A_D5	130	A_CREQ	A_INPACK
89	A_CAD6	A_D13	131	A_CAD23	A_A3
90	A_CAD5	A_D6	132	A_CC/BE3	A_REG
91	V_{CC}	V_{CC}	133	V_{CC}	V_{CC}
92	A_RSVD	A_D14	134	A_CAD24	A_A2
93	A_CAD7	A_D7	135	A_CAD25	A_A1
94	A_CAD8	A_D15	136	A_CAD26	A_A0
95	GND	GND	137	A_CVS1	A_VS1
96	A_CC/BE0	A_CE1	138	A_CINT	A_READY(IREQ)
97	A_CAD9	A_A10	139	A_CSERR	A_WAIT
98	A_CAD10	A_CE2	140	A_CAUDIO	A_BVD2(SPKR)
99	A_CAD11	A_OE	141	A_CSTSCHG	A_BVD1(STSCHG/RI)
100	A_CAD12	A_A11	142	A_CCLKRUN	A_WP(IOIS16)
101	A_CAD13	A_IORD	143	V_{CC}	V_{CC}
102	A_CAD15	A_IOWR	144	A_CCD2	A_CD2
103	A_CAD14	A_A9	145	A_CAD27	A_D0
104	A_CAD16	A_A17	146	A_CAD28	A_D8
105	A_CC/BE1	A_A8	147	GND	GND
106	A_RSVD	A_A18	148	A_CAD29	A_D1
107	A_CPAR	A_A13	149	A_CAD30	A_D9
108	A_CBLOCK	A_A19	150	A_RSVD	A_D2
109	A_CPERR	A_A14	151	A_CAD31	A_D10
110	GND	GND	152	SPKROUT	SPKROUT
111	A_CSTOP	A_A20	153	LATCH	LATCH
112	A_CGNT	A_WE	154	CLOCK	CLOCK
113	A_CDEVSEL	A_A21	155	DATA	DATA
114	V_{CCA}	V_{CCA}	156	MFUNC0	MFUNC0
115	A_CCLK	A_A16	157	MFUNC1	MFUNC1
116	A_CTRDY	A_A22	158	SUSPEND	SUSPEND
117	A_CIRDY	A_A15	159	MFUNC2	MFUNC2
118	V_{CC}	V_{CC}	160	MFUNC3	MFUNC3
119	A_CFRAME	A_A23	161	MFUNC4	MFUNC4
120	A_CC/BE2	A_A12	162	MFUNC5	MFUNC5
121	A_CAD17	A_A24	163	MFUNC6/CLKRUN	MFUNC6/CLKRUN
122	A_CAD18	A_A7	164	C/BE3	C/BE3

Table 2−1. Signal Names by PDV Terminal Number (Continued)

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CardBus PC Card	16-Bit PC Card		CardBus PC Card	16-Bit PC Card
165	$\overline{RI\_OUT}/\overline{PME}$	$\overline{RI\_OUT}/\overline{PME}$	187	AD21	AD21
166	GND	GND	188	AD20	AD20
167	AD25	AD25	189	AD19	AD19
168	$\overline{PRST}$	$\overline{PRST}$	190	AD18	AD18
169	$\overline{GNT}$	$\overline{GNT}$	191	AD17	AD17
170	$\overline{REQ}$	$\overline{REQ}$	192	AD16	AD16
171	AD31	AD31	193	C/$\overline{BE2}$	C/$\overline{BE2}$
172	AD30	AD30	194	$\overline{FRAME}$	$\overline{FRAME}$
173	AD11	AD11	195	V_{CC}	V_{CC}
174	V_{CC}	V_{CC}	196	$\overline{IRDY}$	$\overline{IRDY}$
175	AD29	AD29	197	$\overline{TRDY}$	$\overline{TRDY}$
176	AD28	AD28	198	$\overline{DEVSEL}$	$\overline{DEVSEL}$
177	$\overline{GRST}$	$\overline{GRST}$	199	GND	GND
178	AD27	AD27	200	$\overline{STOP}$	$\overline{STOP}$
179	AD26	AD26	201	$\overline{PERR}$	$\overline{PERR}$
180	V_{CCP}	V_{CCP}	202	$\overline{SERR}$	$\overline{SERR}$
181	AD24	AD24	203	PAR	PAR
182	PCLK	PCLK	204	C/$\overline{BE1}$	C/$\overline{BE1}$
183	IDSEL	IDSEL	205	AD15	AD15
184	AD23	AD23	206	AD14	AD14
185	GND	GND	207	AD13	AD13
186	AD22	AD22	208	AD12	AD12

Table 2−2. Signal Names by GHK Terminal Number

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CardBus PC Card	16-Bit PC Card		CardBus PC Card	16-Bit PC Card
A04	AD12	AD12	B12	AD11	AD11
A05	PAR	PAR	B13	$\overline{GNT}$	$\overline{GNT}$
A06	GND	GND	B14	C/$\overline{BE3}$	C/$\overline{BE3}$
A07	V_{CC}	V_{CC}	B15	MFUNC4	MFUNC4
A08	AD18	AD18	C05	AD13	AD13
A09	GND	GND	C06	$\overline{SERR}$	$\overline{SERR}$
A10	V_{CCP}	V_{CCP}	C07	$\overline{TRDY}$	$\overline{TRDY}$
A11	AD29	AD29	C08	AD16	AD16
A12	V_{CC}	V_{CC}	C09	AD21	AD21
A13	$\overline{REQ}$	$\overline{REQ}$	C10	PCLK	PCLK
A14	GND	GND	C11	$\overline{GRST}$	$\overline{GRST}$
A15	MFUNC5	MFUNC5	C12	AD30	AD30
A16	MFUNC1	MFUNC1	C13	$\overline{PRST}$	$\overline{PRST}$
B05	AD15	AD15	C14	MFUNC6/$\overline{CLKRUN}$	MFUNC6/$\overline{CLKRUN}$
B06	$\overline{STOP}$	$\overline{STOP}$	C15	$\overline{SUSPEND}$	$\overline{SUSPEND}$
B07	$\overline{IRDY}$	$\overline{IRDY}$	D01	AD10	AD10
B08	AD17	AD17	D19	MFUNC0	MFUNC0
B09	AD22	AD22	E01	GND	GND
B10	AD24	AD24	E02	AD7	AD7
B11	AD28	AD28	E03	AD9	AD9

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

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CardBus PC Card	16-Bit PC Card		CardBus PC Card	16-Bit PC Card
E05	NC	NC	H05	B_CCD1	B_CD1
E06	AD14	AD14	H06	AD2	AD2
E07	PERR	PERR	H14	A_CSTSCHG	A_BVD1(STSCHG/RI)
E08	FRAME	FRAME	H15	A_CCLKRUN	A_WP(IOIS16)
E09	AD19	AD19	H17	A_CAUDIO	A_BVD2(SPKR)
E10	IDSEL	IDSEL	H18	A_CSERR	A_WAIT
E11	AD27	AD27	H19	A_CINT	A_READY(IREQ)
E12	AD31	AD31	J01	B_CAD4	B_D12
E13	RI_OUT/PME	RI_OUT/PME	J02	B_CAD3	B_D5
E14	MFUNC2	MFUNC2	J03	B_CAD6	B_D13
E17	DATA	DATA	J05	B_CAD5	B_D6
E18	LATCH	LATCH	J06	B_RSVD	B_D14
E19	A_CAD31	A_D10	J14	A_CAD26	A_A0
F01	AD3	AD3	J15	A_CVS1	A_VS1
F02	AD5	AD5	J17	A_CAD25	A_A1
F03	AD6	AD6	J18	A_CAD24	A_A2
F05	AD8	AD8	J19	V_{CC}	V_{CC}
F06	C/BE1	C/BE1	K01	GND	GND
F07	DEVSEL	DEVSEL	K02	B_CAD7	B_D7
F08	C/BE2	C/BE2	K03	B_CAD8	B_D15
F09	AD20	AD20	K05	B_CC/BE0	B_CE1
F10	AD23	AD23	K06	B_CAD9	B_A10
F11	AD26	AD26	K14	A_CC/BE3	A_REG
F12	AD25	AD25	K15	A_CAD23	A_A3
F13	MFUNC3	MFUNC3	K17	A_CREQ	A_INPACK
F14	SPKROUT	SPKROUT	K18	A_CAD22	A_A4
F15	CLOCK	CLOCK	K19	VR_OUT	VR_OUT
F17	A_RSVD	A_D2	L01	VR_EN	VR_EN
F18	A_CAD29	A_D1	L02	B_CAD10	B_CE2
F19	GND	GND	L03	B_CAD11	B_OE
G01	V_{CC}	V_{CC}	L05	B_CAD13	B_IORD
G02	AD0	AD0	L06	B_CAD12	B_A11
G03	AD1	AD1	L14	A_CAD21	A_A5
G05	AD4	AD4	L15	A_CRST	A_RESET
G06	C/BE0	C/BE0	L17	A_CAD20	A_A6
G14	A_CAD28	A_D8	L18	A_CVS2	A_VS2
G15	A_CAD30	A_D9	L19	A_CAD19	A_A25
G17	A_CAD27	A_D0	M01	B_CAD15	B_IOWR
G18	A_CCD2	A_CD2	M02	B_CAD14	B_A9
G19	V_{CC}	V_{CC}	M03	B_CAD16	B_A17
H01	B_CAD1	B_D4	M05	B_RSVD	B_A18
H02	B_CAD2	B_D11	M06	B_CC/BE1	B_A8
H03	B_CAD0	B_D3	M14	A_CCLK	A_A16

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

TERM. NO.	SIGNAL NAME		TERM. NO.	SIGNAL NAME	
	CardBus PC Card	16-Bit PC Card		CardBus PC Card	16-Bit PC Card
M15	A_CFRAME	A_A23	R14	A_CAD15	A_IOWR
M17	A_CC/BE2	A_A12	R17	A_RSVD	A_A18
M18	A_CAD17	A_A24	R18	A_CPERR	A_A14
M19	A_CAD18	A_A7	R19	GND	GND
N01	V_{CC}	V_{CC}	T01	B_CC/BE2	B_A12
N02	B_CPAR	B_A13	T19	A_CC/BE1	A_A8
N03	B_CBLOCK	B_A19	U05	B_CAD18	B_A7
N05	B_CGNT	B_WE	U06	B_CAD21	B_A5
N06	B_CPERR	B_A14	U07	B_CC/BE3	B_REG
N14	A_CBLOCK	A_A19	U08	B_CVS1	B_VS1
N15	A_CDEVSEL	A_A21	U09	B_CSTSCHG	B_BVD1(STSCHG/RI)
N17	A_CTRDY	A_A22	U10	B_CAD29	B_D1
N18	A_CIRDY	A_A15	U11	A_CCD1	A_CD1
N19	V_{CC}	V_{CC}	U12	A_CAD3	A_D5
P01	GND	GND	U13	A_CAD7	A_D7
P02	B_CSTOP	B_A20	U14	A_CAD10	A_CE2
P03	B_CDEVSEL	B_A21	U15	A_CAD14	A_A9
P05	B_CIRDY	B_A15	V05	B_CAD20	B_A6
P06	B_CCLK	B_A16	V06	B_CAD22	B_A4
P07	B_CVS2	B_VS2	V07	B_CAD24	B_A2
P08	B_CAD23	B_A3	V08	B_CINT	B_READY(IREQ)
P09	B_CCD2	B_CD2	V09	B_CAUDIO	B_BVD2(SPKR)
P10	B_RSVD	B_D2	V10	B_CAD28	B_D8
P11	A_CAD0	A_D3	V11	B_CAD31	B_D10
P12	A_CAD6	A_D13	V12	A_CAD4	A_D12
P13	A_CAD8	A_D15	V13	A_RSVD	A_D14
P14	A_CAD12	A_A11	V14	A_CC/BE0	A_CE1
P15	A_CPAR	A_A13	V15	A_CAD13	A_IORD
P17	A_CSTOP	A_A20	W04	B_CAD17	B_A24
P18	A_CGNT	A_WE	W05	B_CRST	B_RESET
P19	V_{CCA}	V_{CCA}	W06	GND	GND
R01	V_{CCB}	V_{CCB}	W07	B_CAD25	B_A1
R02	B_CTRDY	B_A22	W08	V_{CC}	V_{CC}
R03	B_CFRAME	B_A23	W09	B_CSERR	B_WAIT
R06	B_CAD19	B_A25	W10	B_CAD27	B_D0
R07	B_CREQ	B_INPACK	W11	CLK48	CLK48
R08	B_CAD26	B_A0	W12	A_CAD1	A_D4
R09	B_CCLKRUN	B_WP(IOIS16)	W13	V_{CC}	V_{CC}
R10	B_CAD30	B_D9	W14	GND	GND
R11	A_CAD2	A_D11	W15	A_CAD11	A_OE
R12	A_CAD5	A_D6	W16	A_CAD16	A_A17
R13	A_CAD9	A_A10			

Table 2−3. CardBus PC Card Signal Names Sorted Alphabetically

SIGNAL NAME	TERM NO.		SIGNAL NAME	TERM. NO.		SIGNAL NAME	TERM. NO.		SIGNAL NAME	TERM NO.	
	PDV	GHK		PDV	GHK		PDV	GHK		PDV	GHK
A_CAD0	84	P11	A_CC/BE1	105	T19	AD10	1	D01	B_CAD13	33	L05
A_CAD1	86	W12	A_CC/BE2	120	M17	AD11	173	B12	B_CAD14	35	M02
A_CAD2	85	R11	A_CC/BE3	132	K14	AD12	208	A04	B_CAD15	34	M01
A_CAD3	88	U12	A_CCD1	83	U11	AD13	207	C05	B_CAD16	36	M03
A_CAD4	87	V12	A_CCD2	144	G18	AD14	206	E06	B_CAD17	53	W04
A_CAD5	90	R12	A_CCLK	115	M14	AD15	205	B05	B_CAD18	54	U05
A_CAD6	89	P12	A_CCLKRUN	142	H15	AD16	192	C08	B_CAD19	55	R06
A_CAD7	93	U13	A_CDEVSEL	113	N15	AD17	191	B08	B_CAD20	57	V05
A_CAD8	94	P13	A_CFRAME	119	M15	AD18	190	A08	B_CAD21	59	U06
A_CAD9	97	R13	A_CGNT	112	P18	AD19	189	E09	B_CAD22	60	V06
A_CAD10	98	U14	A_CINT	138	H19	AD20	188	F09	B_CAD23	63	P08
A_CAD11	99	W15	A_CIRDY	117	N18	AD21	187	C09	B_CAD24	65	V07
A_CAD12	100	P14	A_CPAR	107	P15	AD22	186	B09	B_CAD25	66	W07
A_CAD13	101	V15	A_CPERR	109	R18	AD23	184	F10	B_CAD26	67	R08
A_CAD14	103	U15	A_CREQ	130	K17	AD24	181	B10	B_CAD27	76	W10
A_CAD15	102	R14	A_CRST	126	L15	AD25	167	F12	B_CAD28	77	V10
A_CAD16	104	W16	A_CSERR	139	H18	AD26	179	F11	B_CAD29	78	U10
A_CAD17	121	M18	A_CSTOP	111	P17	AD27	178	E11	B_CAD30	79	R10
A_CAD18	122	M19	A_CSTSCHG	141	H14	AD28	176	B11	B_CAD31	82	V11
A_CAD19	123	L19	A_CTRDY	116	N17	AD29	175	A11	B_CAUDIO	72	V09
A_CAD20	125	L17	A_CVS1	137	J15	AD30	172	C12	B_CBLOCK	41	N03
A_CAD21	127	L14	A_CVS2	124	L18	AD31	171	E12	B_CC/BE0	27	K05
A_CAD22	129	K18	A_RSVD	106	R17	B_CAD0	16	H03	B_CC/BE1	37	M06
A_CAD23	131	K15	A_RSVD	92	V13	B_CAD1	18	H01	B_CC/BE2	52	T01
A_CAD24	134	J18	A_RSVD	150	F17	B_CAD2	17	H02	B_CC/BE3	64	U07
A_CAD25	135	J17	AD0	13	G02	B_CAD3	20	J02	B_CCD1	15	H05
A_CAD26	136	J14	AD1	12	G03	B_CAD4	19	J01	B_CCD2	75	P09
A_CAD27	145	G17	AD2	11	H06	B_CAD5	22	J05	B_CCLK	48	P06
A_CAD28	146	G14	AD3	10	F01	B_CAD6	21	J03	B_CCLKRUN	74	R09
A_CAD29	148	F18	AD4	9	G05	B_CAD7	25	K02	B_CDEVSEL	46	P03
A_CAD30	149	G15	AD5	8	F02	B_CAD8	26	K03	B_CFRAME	51	R03
A_CAD31	151	E19	AD6	7	F03	B_CAD9	28	K06	B_CGNT	45	N05
A_CAUDIO	140	H17	AD7	5	E02	B_CAD10	30	L02	B_CINT	69	V08
A_CBLOCK	108	N14	AD8	3	F05	B_CAD11	31	L03	B_CIRDY	50	P05
A_CC/BE0	96	V14	AD9	2	E03	B_CAD12	32	L06	B_CPAR	40	N02

Table 2−3. CardBus PC Card Signal Names Sorted Alphabetically (Continued)

SIGNAL NAME	TERM NO.		SIGNAL NAME	TERM NO.	
	PDV	GHK		PDV	GHK
B_CPERR	42	N06	LATCH	153	E18
B_CREQ	61	R07	MFUNC0	156	D19
B_CRST	58	W05	MFUNC1	157	A16
B_CSERR	71	W09	MFUNC2	159	E14
B_CSTOP	44	P02	MFUNC3	160	F13
B_CSTSCHG	73	U09	MFUNC4	161	B15
B_CTRDY	49	R02	MFUNC5	162	A15
B_CVS1	68	U08	MFUNC6/CLKRUN	163	C14
B_CVS2	56	P07	NC	—	E05
B_RSVD	23	J06	PAR	203	A05
B_RSVD	38	M05	PCLK	182	C10
B_RSVD	80	P10	PERR	201	E07
C/BE0	4	G06	PRST	168	C13
C/BE1	204	F06	REQ	170	A13
C/BE2	193	F08	RI_OUT/PME	165	E13
C/BE3	164	B14	SERR	202	C06
CLK48	81	W11	SPKROUT	152	F14
CLOCK	154	F15	STOP	200	B06
DATA	155	E17	SUSPEND	158	C15
DEVSEL	198	F07	TRDY	197	C07
FRAME	194	E08	V_{CC}	14	A07
GND	6	A06	V_{CC}	39	A12
GND	24	A09	V_{CC}	70	G01
GND	43	A14	V_{CC}	91	G19
GND	62	E01	V_{CC}	118	J19
GND	95	K01	V_{CC}	133	N01
GND	110	P01	V_{CC}	143	N19
GND	147	R19	V_{CC}	174	W08
GND	166	W06	V_{CC}	195	W13
GND	185	F19	V_{CCA}	114	P19
GND	199	W14	V_{CCB}	47	R01
GNT	169	B13	V_{CCP}	180	A10
GRST	177	C11	VR_EN	29	L01
IDSEL	183	E10	VR_OUT	128	K19
IRDY	196	B07			

Table 2−4. 16-Bit PC Card Signal Names Sorted Alphabetically

SIGNAL NAME	TERM. NO.		SIGNAL NAME	TERM. NO.		SIGNAL NAME	TERM NO.	
	PDV	GHK		PDV	GHK		PDV	GHK
A_A0	136	J14	A_D10	151	E19	AD24	181	B10
A_A1	135	J17	A_D11	85	R11	AD25	167	F12
A_A2	134	J18	A_D12	87	V12	AD26	179	F11
A_A3	131	K15	A_D13	89	P12	AD27	178	E11
A_A4	129	K18	A_D14	92	V13	AD28	176	B11
A_A5	127	L14	A_D15	94	P13	AD29	175	A11
A_A6	125	L17	A_INPACK	130	K17	AD30	172	C12
A_A7	122	M19	A_IORD	101	V15	AD31	171	E12
A_A8	105	T19	A_IOWR	102	R14	B_A0	67	R08
A_A9	103	U15	A_OE	99	W15	B_A1	66	W07
A_A10	97	R13	A_READY(IREQ)	138	H19	B_A2	65	V07
A_A11	100	P14	A_REG	132	K14	B_A3	63	P08
A_A12	120	M17	A_RESET	126	L15	B_A4	60	V06
A_A13	107	P15	A_VS1	137	J15	B_A5	59	U06
A_A14	109	R18	A_VS2	124	L18	B_A6	57	V05
A_A15	117	N18	A_WAIT	139	H18	B_A7	54	U05
A_A16	115	M14	A_WE	112	P18	B_A8	37	M06
A_A17	104	W16	A_WP(IOIS16)	142	H15	B_A9	35	M02
A_A18	106	R17	AD0	13	G02	B_A10	28	K06
A_A19	108	N14	AD1	12	G03	B_A11	32	L06
A_A20	111	P17	AD2	11	H06	B_A12	52	T01
A_A21	113	N15	AD3	10	F01	B_A13	40	N02
A_A22	116	N17	AD4	9	G05	B_A14	42	N06
A_A23	119	M15	AD5	8	F02	B_A15	50	P05
A_A24	121	M18	AD6	7	F03	B_A16	48	P06
A_A25	123	L19	AD7	5	E02	B_A17	36	M03
A_BVD1(STSCHG/RI)	141	H14	AD8	3	F05	B_A18	38	M05
A_BVD2(SPKR)	140	H17	AD9	2	E03	B_A19	41	N03
A_CD1	83	U11	AD10	1	D01	B_A20	44	P02
A_CD2	144	G18	AD11	173	B12	B_A21	46	P03
A_CE1	96	V14	AD12	208	A04	B_A22	49	R02
A_CE2	98	U14	AD13	207	C05	B_A23	51	R03
A_D0	145	G17	AD14	206	E06	B_A24	53	W04
A_D1	148	F18	AD15	205	B05	B_A25	55	R06
A_D2	150	F17	AD16	192	C08	B_BVD1(STSCHG/RI)	73	U09
A_D3	84	P11	AD17	191	B08	B_BVD2(SPKR)	72	V09
A_D4	86	W12	AD18	190	A08	B_CD1	15	H05
A_D5	88	U12	AD19	189	E09	B_CD2	75	P09
A_D6	90	R12	AD20	188	F09	B_CE1	27	K05
A_D7	93	U13	AD21	187	C09	B_CE2	30	L02
A_D8	146	G14	AD22	186	B09	B_D0	76	W10
A_D9	149	G15	AD23	184	F10	B_D1	78	U10

Table 2−4. 16-Bit PC Card Signal Names Sorted Alphabetically (Continued)

SIGNAL NAME	TERM NO.		SIGNAL NAME	TERM NO.	
	PDV	GHK		PDV	GHK
B_D2	80	P10	GND	147	R19
B_D3	16	H03	GND	166	W06
B_D4	18	H01	GND	199	W14
B_D5	20	J02	$\overline{\text{GNT}}$	169	B13
B_D6	22	J05	$\overline{\text{GRST}}$	177	C11
B_D7	25	K02	IDSEL	183	E10
B_D8	77	V10	$\overline{\text{IRDY}}$	196	B07
B_D9	79	R10	LATCH	153	E18
B_D10	82	V11	MFUNC0	156	D19
B_D11	17	H02	MFUNC1	157	A16
B_D12	19	J01	MFUNC2	159	E14
B_D13	21	J03	MFUNC3/IRQSER	160	F13
B_D14	23	J06	MFUNC4	161	B15
B_D15	26	K03	MFUNC5	162	A15
B_$\overline{\text{INPACK}}$	61	R07	MFUNC6/$\overline{\text{CLKRUN}}$	163	C14
B_$\overline{\text{IORD}}$	33	L05	NC	—	E05
B_$\overline{\text{IOWR}}$	34	M01	PAR	203	A05
B_$\overline{\text{OE}}$	31	L03	PCLK	182	C10
B_READY($\overline{\text{IREQ}}$)	69	V08	$\overline{\text{PERR}}$	201	E07
B_$\overline{\text{REG}}$	64	U07	$\overline{\text{PRST}}$	168	C13
B_RESET	58	W05	$\overline{\text{REQ}}$	170	A13
B_$\overline{\text{VS1}}$	68	U08	$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	165	E13
B_$\overline{\text{VS2}}$	56	P07	$\overline{\text{SERR}}$	202	C06
B_$\overline{\text{WAIT}}$	71	W09	SPKROUT	152	F14
B_$\overline{\text{WE}}$	45	N05	$\overline{\text{STOP}}$	200	B06
B_WP($\overline{\text{IOIS16}}$)	74	R09	$\overline{\text{SUSPEND}}$	158	C15
C/$\overline{\text{BE0}}$	4	G06	$\overline{\text{TRDY}}$	197	C07
C/$\overline{\text{BE1}}$	204	F06	V_{CC}	14	A07
C/$\overline{\text{BE2}}$	193	F08	V_{CC}	39	A12
C/$\overline{\text{BE3}}$	164	B14	V_{CC}	70	G01
CLK48	185	W11	V_{CC}	91	G19
CLOCK	154	F15	V_{CC}	118	J19
DATA	155	E17	V_{CC}	133	N01
$\overline{\text{DEVSEL}}$	198	F07	V_{CC}	143	N19
$\overline{\text{FRAME}}$	194	E08	V_{CC}	174	W08
GND	6	A06	V_{CC}	195	W13
GND	24	A09	V_{CCA}	114	P19
GND	43	A14	V_{CCB}	47	R01
GND	62	E01	V_{CCP}	180	A10
GND	81	F19	$\overline{\text{VR\_EN}}$	29	L01
GND	95	K01	VR_OUT	128	K19
GND	110	P01			

The terminals are grouped in tables by functionality, such as PCI system function and power supply function, for quick reference (see Table 2−5 through Table 2−15). The terminal names and numbers are also listed for convenient reference.

Table 2−5. Power Supply Terminals

TERMINAL			I/O	DESCRIPTION
NAME	NO. PDV	NO. GHK		
GND	6, 24, 43, 62, 95, 110, 147, 166, 185, 199	A06, A09, A14, E01, F19, K01, P01, R19, W06, W14	−	Device ground terminals
V_{CC}	14, 39, 70, 91, 118, 133, 143, 174, 195	A07, A12, G01, G19, J19, N01, N19, W08, W13	−	3.3-V power terminals
V_{CCA}	114	P19	−	PC Card A signaling rail power input; clamped per PC Card specification
V_{CCB}	47	R01	−	PC Card B signaling rail power input; clamped per PC Card specification
V_{CCP}	180	A10	−	PCI signaling clamp rail power input; clamped per PCI specification
$\overline{\text{VR\_EN}}$	29	L01	I	Internal voltage regulator enable. Active-low
VR_OUT	128	K19	O	Internal voltage regulator output (1.8 V) for external bypass capacitor

Table 2−6. PC Card Power Switch Terminals

TERMINAL			I/O	DESCRIPTION
NAME	NO. PDV	NO. GHK		
CLOCK	154	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 PCI1620 output by using bit 27 (P2CCLK) in the system control register (PCI offset 80h, see Section 4.31). For use with the TPS222X, the maximum frequency of this signal is limited to 2 MHz. However, the PCI1620 requires a 16-KHz to 100-KHz frequency range. As an input, this terminal requires an external 32-kHz clock. If a system design defines this terminal as an output, then this terminal requires an external pulldown resistor. The frequency of the PCI1620 output CLOCK is derived from the internal ring oscillator (16 kHz typical).
DATA	155	E17	O	Power switch data. DATA is used to communicate socket power control information serially to the power switch.
LATCH	153	E18	I/O	Power switch latch. LATCH is asserted by the PCI1620 to indicate to the power switch that the data on the DATA line is valid. The LATCH terminal is also used to indicate the presence of an external EEPROM; when a pulldown resistor is implemented on this terminal, the MFUNC1 and MFUNC4 terminals provide the serial EEPROM SDA and SCL interface.

Table 2−7. PCI System Terminals

TERMINAL			I/O	DESCRIPTION
NAME	NO.			
	PDV	GHK		
$\overline{\text{GRST}}$	177	C11	I	Global reset. When the global reset is asserted, the $\overline{\text{GRST}}$ signal causes the PCI1620 to place all output buffers in a high-impedance state and reset all internal registers. When $\overline{\text{GRST}}$ is asserted, the device is completely in its default state. For systems that require wake-up from D3, $\overline{\text{GRST}}$ normally is asserted only during initial boot. $\overline{\text{PRST}}$ should be asserted during $\overline{\text{GRST}}$ and for resets subsequent to the initial $\overline{\text{GRST}}$ so that PME context is retained during the transition from D3 to D0. For systems that do not require wake-up from D3, $\overline{\text{GRST}}$ should be tied to $\overline{\text{PRST}}$. When the $\overline{\text{SUSPEND}}$ mode is enabled together with $\overline{\text{GRST}}$, the device is protected from $\overline{\text{GRST}}$; the internal registers are not reset, but all outputs are placed in a high-impedance state.
PCLK	182	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.
$\overline{\text{PRST}}$	168	C13	I	PCI reset. When the PCI bus reset is asserted, $\overline{\text{PRST}}$ causes the PCI1620 to reset internal registers and place all output buffers in a high-impedance state. When $\overline{\text{PRST}}$ is asserted, the device can generate the $\overline{\text{PME}}$ signal only if it is enabled. After $\overline{\text{PRST}}$ is deasserted, the PCI1620 is in a default state. When the $\overline{\text{SUSPEND}}$ mode is enabled together with $\overline{\text{PRST}}$, the device is protected from $\overline{\text{PRST}}$ and the internal registers are preserved, but all outputs are placed in a high-impedance state.

Table 2−8. PCI Address and Data Terminals

TERMINAL			I/O	DESCRIPTION
NAME	NO.			
	PDV	GHK		
AD31	171	E12	I/O	PCI address/data bus. These signals make up the multiplexed PCI address and data bus on the primary interface. During the address phase of a primary bus PCI cycle, AD31−AD0 contain a 32-bit address or other destination information. During the data phase, AD31−AD0 contain data.
AD30	172	C12		
AD29	175	A11		
AD28	176	B11		
AD27	178	E11		
AD26	179	F11		
AD25	167	F12		
AD24	181	B10		
AD23	184	F10		
AD22	186	B09		
AD21	187	C09		
AD20	188	F09		
AD19	189	E09		
AD18	190	A08		
AD17	191	B08		
AD16	192	C08		
AD15	205	B05		
AD14	206	E06		
AD13	207	C05		
AD12	208	A04		
AD11	173	B12		
AD10	1	D01		
AD9	2	E03		
AD8	3	F05		
AD7	5	E02		
AD6	7	F03		
AD5	8	F02		
AD4	9	G05		
AD3	10	F01		
AD2	11	H06		
AD1	12	G03		
AD0	13	G02		
C/BE3 C/BE2 C/BE1 C/BE0	164 193 204 4	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/BE3−C/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/BE0 applies to byte 0 (AD7−AD0), C/BE1 applies to byte 1 (AD15−AD8), C/BE2 applies to byte 2 (AD23−AD16), and C/BE3 applies to byte 3 (AD31−AD24).
PAR	203	A05	I/O	PCI bus parity. In all PCI bus read and write cycles, the PCI1620 calculates even parity across the AD31−AD0 and C/BE3−C/BE0 buses. As an initiator during PCI cycles, the PCI1620 outputs this parity indicator with a one-PCLK delay. As a target during PCI cycles, the calculated parity is compared to the parity indicator of the initiator. A compare error results in the assertion of a parity error (PERR).

Table 2−9. PCI Interface Control Terminals

TERMINAL			I/O	DESCRIPTION
NAME	NO.			
	PDV	GHK		
$\overline{\text{DEVSEL}}$	198	F07	I/O	PCI device select. The PCI1620 asserts $\overline{\text{DEVSEL}}$ to claim a PCI cycle as the target device. As a PCI initiator on the bus, the PCI1620 monitors $\overline{\text{DEVSEL}}$ until a target responds. If no target responds before timeout occurs, then the PCI1620 terminates the cycle with an initiator abort.
$\overline{\text{FRAME}}$	194	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.
$\overline{\text{GNT}}$	169	B13	I	PCI bus grant. $\overline{\text{GNT}}$ is driven by the PCI bus arbiter to grant the PCI1620 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	183	E10	I	Initialization device select. IDSEL selects the PCI1620 during configuration space accesses. IDSEL can be connected to one of the upper 24 PCI address lines† on the PCI bus.
$\overline{\text{IRDY}}$	196	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.
$\overline{\text{PERR}}$	201	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).
$\overline{\text{REQ}}$	170	A13	O	PCI bus request. $\overline{\text{REQ}}$ is asserted by the PCI1620 to request access to the PCI bus as an initiator.
$\overline{\text{SERR}}$	202	C06	O	PCI system error. $\overline{\text{SERR}}$ is an output that is pulsed from the PCI1620 when enabled through bit 8 of the command register (PCI offset 04h, see Section 4.4), indicating a system error has occurred. The PCI1620 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.
$\overline{\text{STOP}}$	200	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.
$\overline{\text{TRDY}}$	197	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.

† Care must be exercised in selection of the address line that is used for connecting to IDSEL. Check each PCI component to avoid the use of address lines that it may have reserved, because address lines used can vary from one device to another of the same device type. For example, one commonly-used chipset uses lines AD11 and AD12, and assignment of IDSEL to either of those lines in an implementation using that chipset would result in an address conflict.

Table 2−10. Multifunction and Miscellaneous Terminals

TERMINAL			I/O	DESCRIPTION
NAME	NO.			
	PDV	GHK		
CLK48	81	W11	I	48-MHz clock input. This clock is used as a clock source for internal microcontroller.
MFUNC0	156	D19	I/O	Multifunction terminal 0. MFUNC0 can be configured as parallel PCI interrupt INTA, GPI0, GPO0, socket activity LED output, ZV switching outputs, CardBus audio PWM, GPE, or a parallel IRQ. See Section 4.38, *Multifunction Routing Status Register*, for configuration details.
MFUNC1	157	A16	I/O	Multifunction terminal 1. MFUNC1 can be configured as parallel PCI interrupt INTB, GPI1, GPO1, socket activity LED output, ZV switching outputs, CardBus audio PWM, GPE, or a parallel IRQ. See Section 4.38, *Multifunction Routing Status Register*, for configuration details. Serial data (SDA). When LATCH is detected low during 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 3.4.4, *Loading the Subsystem Identification (EEPROM Interface)*, for details on other serial bus applications.
MFUNC2	159	E14	I/O	Multifunction terminal 2. MFUNC2 can be configured as GPI2, GPO2, socket activity LED output, ZV switching outputs, CardBus audio PWM, GPE, RI_OUT, or a parallel IRQ. See Section 4.38, *Multifunction Routing Status Register*, for configuration details.
MFUNC3/ IRQSER	160	F13	I/O	Multifunction terminal 3. MFUNC3 can be configured as a parallel IRQ or the serialized interrupt signal IRQSER. See Section 4.38, *Multifunction Routing Status Register*, for configuration details.
MFUNC4	161	B15	I/O	Multifunction terminal 4. MFUNC4 can be configured as PCI LOCK, GPI3, GPO3, socket activity LED output, ZV switching outputs, CardBus audio PWM, GPE, RI_OUT, or a parallel IRQ. See Section 4.38, *Multifunction Routing Status Register*, for configuration details. Serial clock (SCL). When LATCH is detected low during 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 3.4.4, *Loading the Subsystem Identification (EEPROM Interface)*, for details on other serial bus applications.
MFUNC5	162	A15	I/O	Multifunction terminal 5. MFUNC5 can be configured as GPI4, GPO4, socket activity LED output, ZV switching outputs, CardBus audio PWM, GPE, or a parallel IRQ. See Section 4.38, *Multifunction Routing Status Register*, for configuration details.
MFUNC6/ CLKRUN	163	C14	I/O	Multifunction terminal 6. MFUNC6 can be configured as a PCI CLKRUN or a parallel IRQ. See Section 4.38, *Multifunction Routing Status Register*, for configuration details.
RI_OUT/PME	165	E13	O	Ring indicate out and power-management event output. Terminal provides an output for ring-indicate or PME signals.
SPKROUT	152	F14	O	Speaker output. SPKROUT is the output to the host system that can carry SPKR or CAUDIO through the PCI1620 from the PC Card interface. SPKROUT is driven as the exclusive-OR combination of card SPKR//CAUDIO inputs.
SUSPEND	158	C15	I	Suspend. SUSPEND protects the internal registers from clearing when the GRST or PRST signal is asserted. See Section 3.7.5, *Suspend Mode*, for details.

Table 2−11. CardBus PC Card Interface System Terminals (Slots A and B)

TERMINAL					I/O	DESCRIPTION
NAME	NUMBER					
	SLOT A†		SLOT B‡			
	PDV	GHK	PDV	GHK		
CCLK	115	M14	48	P06	O	CardBus clock. CCLK provides synchronous timing for all transactions on the CardBus interface. All signals except CRST, CCLKRUN, CINT, CSTSCHG, CAUDIO, CCD2, 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.
CCLKRUN	142	H15	74	R09	I/O	CardBus clock run. CCLKRUN is used by a CardBus PC Card to request an increase in the CCLK frequency, and by the PCI1620 to indicate that the CCLK frequency is going to be decreased.
CRST	126	L15	58	W05	O	CardBus reset. CRST brings CardBus PC Card-specific registers, sequencers, and signals to a known state. When CRST is asserted, all CardBus PC Card signals are placed in a high-impedance state, and the PCI1620 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 are A_CCLK.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 48 and P06 are B_CCLK.

Table 2−12. CardBus PC Card Address and Data Terminals (Slots A and B)

TERMINAL					I/O	DESCRIPTION
NAME	NUMBER					
	SLOT A†		SLOT B‡			
	PDV	GHK	PDV	GHK		
CAD31	151	E19	82	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	149	G15	79	R10		
CAD29	148	F18	78	U10		
CAD28	146	G14	77	V10		
CAD27	145	G17	76	W10		
CAD26	136	J14	67	R08		
CAD25	135	J17	66	W07		
CAD24	134	J18	65	V07		
CAD23	131	K15	63	P08		
CAD22	129	K18	60	V06		
CAD21	127	L14	59	U06		
CAD20	125	L17	57	V05		
CAD19	123	L19	55	R06		
CAD18	122	M19	54	U05		
CAD17	121	M18	53	W04		
CAD16	104	W16	36	M03		
CAD15	102	R14	34	M01		
CAD14	103	U15	35	M02		
CAD13	101	V15	33	L05		
CAD12	100	P14	32	L06		
CAD11	99	W15	31	L03		
CAD10	98	U14	30	L02		
CAD9	97	R13	28	K06		
CAD8	94	P13	26	K03		
CAD7	93	U13	25	K02		
CAD6	89	P12	21	J03		
CAD5	90	R12	22	J05		
CAD4	87	V12	19	J01		
CAD3	88	U12	20	J02		
CAD2	85	R11	17	H02		
CAD1	86	W12	18	H01		
CAD0	84	P11	16	H03		
CC/$\overline{BE3}$	132	K14	64	U07	I/O	CardBus bus commands and byte enables. CC/$\overline{BE3}$−CC/$\overline{BE0}$ are multiplexed on the same CardBus terminals. During the address phase of a CardBus cycle, CC/$\overline{BE3}$−CC/$\overline{BE0}$ 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/$\overline{BE0}$ applies to byte 0 (CAD7−CAD0), CC/$\overline{BE1}$ applies to byte 1 (CAD15−CAD8), CC/$\overline{BE2}$ applies to byte 2 (CAD23−CAD8), and CC/$\overline{BE3}$ applies to byte 3 (CAD31−CAD24).
CC/$\overline{BE2}$	120	M17	52	T01		
CC/$\overline{BE1}$	105	T19	37	M06		
CC/$\overline{BE0}$	96	V14	27	K05		
CPAR	107	P15	40	N02	I/O	CardBus parity. In all CardBus read and write cycles, the PCI1620 calculates even parity across the CADx and CC/$\overline{BEx}$ buses. As an initiator during CardBus cycles, the PCI1620 outputs CPAR with a one-CCLK delay. As a target during CardBus cycles, the PCI1620 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 are A_CPAR.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 40 and N02 are B_CPAR.

Table 2−13. CardBus PC Card Terminals (Slots A and B)

TERMINAL					I/O	DESCRIPTION
NAME	NUMBER					
	SLOT A†		SLOT B‡			
	PDV	GHK	PDV	GHK		
CAUDIO	140	H17	72	V09	I	CardBus audio. CAUDIO is a digital input signal from a PC Card to the system speaker. The PCI1620 supports the binary audio mode and outputs a binary signal from the card to SPKROUT.
CBLOCK	108	N14	41	N03	I/O	CardBus lock. CBLOCK is used to gain exclusive access to a target.
CCD1 CCD2	83 144	U11 G18	15 75	H05 P09	I	The card-detect terminals and the voltage-sense terminals are used together to determine the insertion event and type of PC Card inserted (16-bit, CardBus, or UltraMedia). The PCI1620 implements changes in the interrogation logic that handles this function. See Section 3.5.1, *Card Detedtion in an UltraMedia System*, for more information.
CDEVSEL	113	N15	46	P03	I/O	CardBus device select. The PCI1620 asserts CDEVSEL to claim a CardBus cycle as the target device. As a CardBus initiator on the bus, the PCI1620 monitors CDEVSEL until a target responds. If no target responds before timeout occurs, then the PCI1620 terminates the cycle with an initiator abort.
CFRAME	119	M15	51	R03	I/O	CardBus cycle frame. CFRAME is driven by the initiator of a CardBus bus cycle. CFRAME is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When CFRAME is deasserted, the CardBus bus transaction is in the final data phase.
CGNT	112	P18	45	N05	O	CardBus bus grant. CGNT is driven by the PCI1620 to grant a CardBus PC Card access to the CardBus bus after the current data transaction has been completed.
CINT	138	H19	69	V08	I	CardBus interrupt. CINT is asserted low by a CardBus PC Card to request interrupt servicing from the host.
CIRDY	117	N18	50	P05	I/O	CardBus initiator ready. 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 CIRDY and CTRDY are asserted. Until CIRDY and CTRDY are both sampled asserted, wait states are inserted.
CPERR	109	R18	42	N06	I/O	CardBus parity error. CPERR reports parity errors during CardBus transactions, except during special cycles. It is driven low by a target two clocks following the data cycle when a parity error is detected.
CREQ	130	K17	61	R07	I	CardBus request. CREQ indicates to the arbiter that the CardBus PC Card desires use of the CardBus bus as an initiator.
CSERR	139	H18	71	W09	I	CardBus system error. CSERR reports address parity errors and other system errors that could lead to catastrophic results. CSERR is driven by the card synchronous to CCLK, but deasserted by a weak pullup, and deassertion may take several CCLK periods. The PCI1620 can report CSERR to the system by assertion of SERR on the PCI interface.
CSTOP	111	P17	44	P02	I/O	CardBus stop. CSTOP is driven by a CardBus target to request the initiator to stop the current CardBus transaction. CSTOP is used for target disconnects, and is commonly asserted by target devices that do not support burst data transfers.
CSTSCHG	141	H14	73	U09	I	CardBus status change. CSTSCHG alerts the system to a change in the card status, and is used as a wake-up mechanism.
CTRDY	116	N17	49	R02	I/O	CardBus target ready. 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 CIRDY and CTRDY are asserted; until this time, wait states are inserted.
CVS1 CVS2	137 124	J15 L18	68 56	U08 P07	I/O	The card-detect terminals and the voltage-sense terminals are used together to determine the insertion event and type of PC Card inserted (16-bit, CardBus, or UltraMedia). The PCI1620 implements changes in the interrogation logic that handles this function.

† Terminal name for slot A is preceded with A_. For example, the full name for terminals 140 and H17 are A_CAUDIO.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 72 and V09 are B_CAUDIO.

Table 2−14. 16-Bit PC Card Address and Data Terminals (Slots A and B)

TERMINAL					I/O	DESCRIPTION
NAME	NUMBER					
	SLOT A†		SLOT B‡			
	PDV	GHK	PDV	GHK		
A25	123	L19	55	R06	O	PC Card address. 16-bit PC Card address lines. A25 is the most significant bit.
A24	121	M18	53	W04		
A23	119	M15	51	R03		
A22	116	N17	49	R02		
A21	113	N15	46	P03		
A20	111	P17	44	P02		
A19	108	N14	41	N03		
A18	106	R17	38	M05		
A17	104	W16	36	M03		
A16	115	M14	48	P06		
A15	117	N18	50	P05		
A14	109	R18	42	N06		
A13	107	P15	40	N02		
A12	120	M17	52	T01		
A11	100	P14	32	L06		
A10	97	R13	28	K06		
A9	103	U15	35	M02		
A8	105	T19	37	M06		
A7	122	M19	54	U05		
A6	125	L17	57	V05		
A5	127	L14	59	U06		
A4	129	K18	60	V06		
A3	131	K15	63	P08		
A2	134	J18	65	V07		
A1	135	J17	66	W07		
A0	136	J14	67	R08		
D15	94	P13	26	K03	I/O	PC Card data. 16-bit PC Card data lines. D15 is the most significant bit.
D14	92	V13	23	J06		
D13	89	P12	21	J03		
D12	87	V12	19	J01		
D11	85	R11	17	H02		
D10	151	E19	82	V11		
D9	149	G15	79	R10		
D8	146	G14	77	V10		
D7	93	U13	25	K02		
D6	90	R12	22	J05		
D5	88	U12	20	J02		
D4	86	W12	18	H01		
D3	84	P11	16	H03		
D2	150	F17	80	P10		
D1	148	F18	78	U10		
D0	145	G17	76	W10		

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

Table 2−15. 16-Bit PC Card Interface Control Terminals (Slots A and B)

TERMINAL					I/O	DESCRIPTION
NAME	NUMBER					
	SLOT A†		SLOT B‡			
	PDV	GHK	PDV	GHK		
BVD1 (STSCHG/RI)	141	H14	73	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. STSCHG is used to alert the system to a change in the READY, write-protect, or battery voltage detect condition of a 16-bit I/O PC Card. Ring indicate. RI is used by 16-bit modem cards to indicate a ring detection.
BVD2 (SPKR)	140	H17	72	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. 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 PCI1620 and are output on SPKROUT.
CE1 CE2	96 98	V14 U14	27 30	K05 L02	O	Card enable 1 and card enable 2. CE1 and CE2 enable even- and odd-numbered address bytes. CE1 enables even-numbered address bytes, and CE2 enables odd-numbered address bytes.
INPACK	130	K17	61	R07	I	Input acknowledge. INPACK is asserted by the PC Card when it can respond to an I/O read cycle at the current address.
IORD	101	V15	33	L05	O	I/O read. IORD is asserted by the PCI1620 to enable 16-bit I/O PC Card data output during host I/O read cycles.
IOWR	102	R14	34	M01	O	I/O write. IOWR is driven low by the PCI1620 to strobe write data into 16-bit I/O PC Cards during host I/O write cycles.
OE	99	W15	31	L03	O	Output enable. OE is driven low by the PCI1620 to enable 16-bit memory PC Card data output during host memory read cycles.
READY (IREQ)	138	H19	69	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 the 16-bit memory PC Cards to indicate that the memory card circuits are busy processing a previous write command. READY is driven high when the 16-bit memory PC Card is ready to accept a new data transfer command. Interrupt request. IREQ is asserted by a 16-bit I/O PC Card to indicate to the host that a device on the 16-bit I/O PC Card requires service by the host software. IREQ is high (deasserted) when no interrupt is requested.
REG	132	K14	64	U07	O	Attribute memory select. REG remains high for all common memory accesses. When REG is asserted, access is limited to attribute memory (OE or WE active) and to the I/O space (IORD or 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.

† Terminal name for slot A is preceded with A_. For example, the full name for terminals 141 and H14 are A_BVD1(STSCHG/RI).
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 73 and U09 are B_BVD1(STSCHG/RI).

Table 2−15. 16-Bit PC Card Interface Control Terminals (Slots A and B) (Continued)

TERMINAL					I/O	DESCRIPTION
NAME	NUMBER					
	SLOT A†		SLOT B‡			
	PDV	GHK	PDV	GHK		
RESET	126	L15	58	W05	O	PC Card reset. RESET forces a hard reset to a 16-bit PC Card.
$\overline{\text{WAIT}}$	139	H18	71	W09	I	Bus cycle wait. $\overline{\text{WAIT}}$ is driven by a 16-bit PC Card to extend the completion of the memory or I/O cycle in progress.
$\overline{\text{WE}}$	112	P18	45	N05	O	Write enable. $\overline{\text{WE}}$ is used to strobe memory write data into 16-bit memory PC Cards. $\overline{\text{WE}}$ is also used for memory PC Cards that employ programmable memory technologies.
WP ($\overline{\text{IOIS16}}$)	142	H15	74	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 ($\overline{\text{IOIS16}}$) function. I/O is 16 bits. $\overline{\text{IOIS16}}$ applies to 16-bit I/O PC Cards. $\overline{\text{IOIS16}}$ is asserted by the 16-bit PC Card when the address on the bus corresponds to an address to which the 16-bit PC Card responds, and the I/O port that is addressed is capable of 16-bit accesses.

† Terminal name for slot A is preceded with A_. For example, the full name for terminals 99 and W15 are A_$\overline{\text{OE}}$.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 31 and L03 are B_$\overline{\text{OE}}$.

UltraMedia defines additional functionality for the CardBus/PC Card terminals. Table 2−16 gives the signal names and mapping of this additional functionality to the PCI1620 CardBus/PC Card terminals, with reference to the 68-pin card socket. Table 2−17 provides the signal descriptions.

Table 2−16. UltraMedia Mapping to the PCMCIA 68-Terminal Connector

CARD PIN	16-Bit PC Card	CardBus	UltraMedia			
			SmartMedia	MMC/SD	Memory Stick	Smart Card
1	GND	GND	GND	GND	GND	GND
2	D3	CAD0	RSVD	RSVD	RSVD	RSVD
3	D4	CAD1	RSVD	RSVD	RSVD	RSVD
4	D5	CAD3	SM_D3	RSVD	MS_BS	RSVD
5	D6	CAD5	SM_D2	RSVD	MS_SDIO	RSVD
6	D7	CAD7	SM_D1	RSVD	MS_RFU5	RSVD
7	CE1	CC/BE0	SM_D0	RSVD	MS_RFU7	RSVD
8	A10	CAD9	SM_WP	RSVD	RSVD	RSVD
9	OE	CAD11	SM_R/B	RSVD	RSVD	RSVD
10	A11	CAD12	RSVD	RSVD	RSVD	RSVD
11	A9	CAD14	RSVD	RSVD	RSVD	RSVD
12	A8	CC/BE1	SM_RE	RSVD	RSVD	RSVD
13	A13	CPAR	SM_WE	RSVD	RSVD	RSVD
14	A14	CPERR	RSVD	SD_DATA1/IRQ	RSVD	RSVD
15	WE	CGNT	SM_ALE	RSVD	RSVD	SC_RFU
16	READY/IREQ	CINT	RSVD	SD_CD/DATA3	RSVD	RSVD
17	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
18	V_{PP}/V_{CORE}	V_{PP}/V_{CORE}	V_{PP}/V_{CORE}	V_{PP}/V_{CORE}	V_{PP}/V_{CORE}	V_{PP}/V_{CORE}
19	A16	CCLK	RSVD	SD_CLK	MS_SCLK	SC_CLK
20	A15	CIRDY	MC_WP	MC_WP	RSVD	RSVD
21	A12	CC/BE2	SM_CLE	RSVD	RSVD	SC_RST
22	A7	CAD18	RSVD	RSVD	RSVD	SC_GPIO0
23	A6	CAD20	RSVD	RSVD	RSVD	SC_GPIO1
24	A5	CAD21	RSVD	RSVD	RSVD	SC_GPIO2
25	A4	CAD22	RSVD	RSVD	RSVD	SC_GPIO3
26	A3	CAD23	RSVD	RSVD	RSVD	SC_GPIO4
27	A2	CAD24	RSVD	RSVD	RSVD	SC_GPIO5
28	A1	CAD25	RSVD	RSVD	RSVD	SC_GPIO06
29	A0	CAD26	RSVD	RSVD	RSVD	SC_GPIO7
30	D0	CAD27	RSVD	RSVD	RSVD	RSVD
31	D1	CAD29	RSVD	RSVD	RSVD	RSVD
32	D2	RFU	RSVD	RSVD	RSVD	RSVD
33	WP(IOIS16)	CCLKRUN	RSVD	RSVD	RSVD	RSVD
34	GND	GND	GND	GND	GND	GND
35	GND	GND	GND	GND	GND	GND
36	CD1	CCD1	CCD1	CCD1	CCD1	CCD1
37	D11	CAD2	SM_D4	RSVD	RSVD	RSVD
38	D12	CAD4	SM_D5	RSVD	RSVD	RSVD
39	D13	CAD6	SM_D6	RSVD	RSVD	RSVD
40	D14	RFU	SM_D7	RSVD	RSVD	RSVD

Table 2−16. UltraMedia Mapping to the PCMCIA 68-Terminal Connector (Continued)

TERM.	16-Bit PC Card	CardBus	UltraMedia			
			SmartMedia	MMC/SD	Memory Stick	Smart Card
41	D15	CAD8	SM_LVD	RSVD	RSVD	RSVD
42	CE2	CAD10	RSVD	RSVD	RSVD	RSVD
43	VS1	CVS1	CVS1	CVS1	CVS1	CVS1
44	IORD/RFU	CAD13	RSVD	RSVD	RSVD	RSVD
45	IOWR/RFU	CAD15	RSVD	RSVD	RSVD	RSVD
46	A17	CAD16	RSVD	RSVD	RSVD	RSVD
47	A18	RFU	SQRY1	SQRY1	SQRY1	SQRY1
48	A19	CBLOCK	RSVD	SD_DATA0	RSVD	RSVD
49	A20	CSTOP	RSVD	SD_CMD	RSVD	RSVD
50	A21	CDEVSEL	RSVD	SD_DATA2	RSVD	RSVD
51	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
52	V_{PP}/V_{CORE}	V_{PP}/V_{CORE}	V_{PP}/V_{CORE}	V_{PP}/V_{CORE}	V_{PP}/V_{CORE}	V_{PP}/V_{CORE}
53	A22	CTRDY	MC_CD	MC_CD	MC_CD	MC_CD
54	A23	CFRAME	RSVD	RSVD	RSVD	SC_FCB
55	A24	CAD17	SM_CE	RSVD	RSVD	SC_I/O
56	A25	CAD19	SQRYDR	SQRYDR	SQRYDR	SQRYDR
57	VS2	CVS2	CVS2	CVS2	CVS2	CVS2
58	RESET	CRST	SQRY2	SQRY2	SQRY2	SQRY2
59	WAIT	CSERR	SQRY3	SQRY3	SQRY3	SQRY3
60	INPACK/RFU	CREQ	SQRY4	SQRY4	SQRY4	SQRY4
61	REG	CC/BE3	SQRY5	SQRY5	SQRY5	SQRY5
62	BVD2(SPKR)	CAUDIO	SQRY6	SQRY6	SQRY6	SQRY6
63	BVD1(STSCHG/RI)	CSTSCHG	SQRY7	SQRY7	SQRY7	SQRY7
64	D8	CAD28	SQRY8	SQRY8	SQRY8	SQRY8
65	D9	CAD30	SQRY9	SQRY9	SQRY9	SQRY9
66	D10	CAD31	SQRY10	SQRY10	SQRY10	SQRY10
67	CD2	CCD2	CCD2	CCD2	CCD2	CCD2
68	GND	GND	GND	GND	GND	GND

Table 2−17. UltraMedia Terminals (Slots A and B)

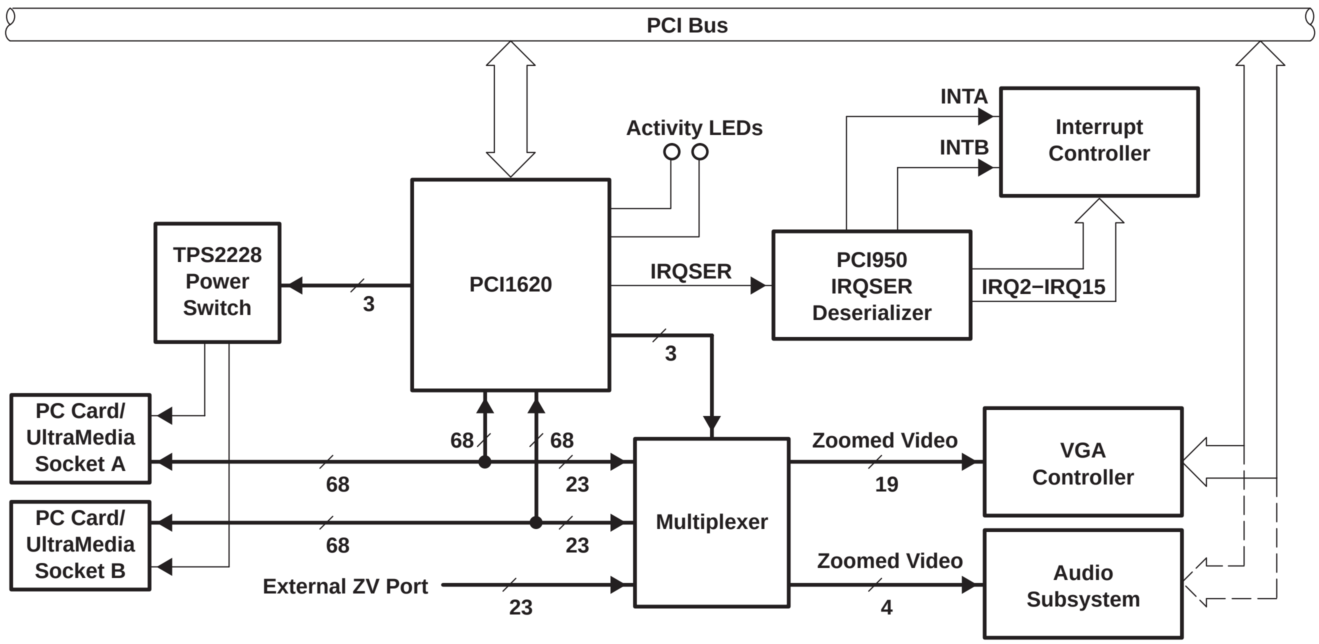
TERMINAL					I/O	DESCRIPTION
NAME	NO.					
	SLOT A		SLOT B			
	PDV	GHK	PDV	GHK		
$\overline{\text{MC\_CD}}$	116	N17	49	R02	I	Media Card detect. This input is asserted when an UltraMedia adapter and its associated media are inserted. For all other UltraMedia cards, the UltraMedia socket is not powered until this signal is low.
$\overline{\text{MC\_WP}}$	117	N18	50	P05	I	UltraMedia write protect data. This signal indicates that the media inserted in the socket is write protected.
MS_BS	88	U12	20	J02	O	Memory Stick bus state. This signal provides Memory Stick bus state information.
MS_RFU5	93	U13	25	K02	I	Memory Stick reserved. This terminal is in a high-impedance state when an UltraMedia Memory Stick adapter has been inserted.
MS_RFU7	96	V14	27	K05	I	Memory Stick reserved. This terminal is in a high-impedance state when an UltraMedia Memory Stick adapter has been inserted.
MS_SCLK	115	M14	48	P06	O	Memory Stick clock. This output provides the MS clock, which operates at 16 MHz.
MS_SDIO	90	R12	22	J05	I/O	Memory Stick serial data I/O. This signal provides Memory Stick data input/output.
SC_CLK	115	M14	48	P06	O	Smart Card clock. The PCI1620 drives a 3-MHz clock to the Smart Card interface when enabled.
SC_FCB	119	M15	51	R03	I	Smart Card function code. The PCI1620 does not support synchronous Smart Cards as specified in ISO/IEC 7816-10, and this terminal is in a high-impedance state when an UltraMedia Smart Card adapter has been inserted.
SC_GPIO0	122	M19	54	U05	I/O	Smart Card general-purpose I/O terminals. These signals can be controlled by firmware and are used as control signals for an external Smart Card interface chip or level shifter.
SC_GPIO1	125	L17	57	V05		
SC_GPIO2	127	L14	59	U06		
SC_GPIO3	129	K18	60	V06		
SC_GPIO4	131	K15	63	P08		
SC_GPIO5	134	J18	65	V07		
SC_GPIO6	135	J17	66	W07		
SC_GPIO7	136	J14	67	R08		
SC_IO	121	M18	53	W04	I/O	Smart Card input/output. This terminal is the input/output terminal for the character exchange between the PCI1620 and the Smart Cards.
SC_RFU	112	P18	45	N05	I	Smart Card reserved. This terminal is in a high-impedance state when an UltraMedia Smart Card adapter has been inserted.
SC_RST	120	M17	52	T01	O	Smart Card reset. This signal starts and stops the Smart Card reset sequence. The PCI1620 asserts this reset when requested by the host.
SD_CD/ DATA3	138	H19	69	V08	I/O	SD flash card detect/data 3. This signal provides the SD data path per the SD specification.
SD_CLK	115	M14	48	P06	O	SD flash clock. This output provides the MMC/SD clock, which operates at 16 MHz.
SD_CMD	111	P17	44	P02	I/O	SD flash command. This signal provides the SD command per the SD specification.
SD_DATA0	108	N14	41	N03	I/O	SD flash data 0. This signal provides the MMC_SD data path per the SD specification.
SD_DATA1	109	R18	42	N06	I/O	SD flash data 1. This signal provides the SD data path per the SD specification.
SD_DATA2	113	N15	46	P03	I/O	SD flash data 2. This signal provides the SD data path and CD per the SD specification.
SM_ALE	112	P18	45	N05	O	SmartMedia address latch enable. This signal functions as specified in the SmartMedia specification, and is used to latch addresses passed over SM_D7−SM_D0.
$\overline{\text{SM\_CE}}$	121	M18	53	W04	O	SmartMedia card enable. This signal functions as specified in the SmartMedia specification, and is used to enable the media for a pending transaction.
SM_CLE	120	M17	52	T01	O	SmartMedia command latch enable. This signal functions as specified in the SmartMedia specification, and is used to latch commands passed over SM_D7−SM_D0.

Table 2−17. UltraMedia Terminals (Slots A & B) (Continued)

TERMINAL					I/O	DESCRIPTION
NAME	NO.					
	SLOT A		SLOT B			
	PDV	GHK	PDV	GHK		
SM_D0	96	V14	27	K05	I/O	SmartMedia data terminals. These signals pass data to and from the SmartMedia, and function as specified in the SmartMedia specification.
SM_D1	93	U13	25	K02		
SM_D2	90	R12	22	J05		
SM_D3	88	U12	20	J02		
SM_D4	85	R11	17	H02		
SM_D5	87	V12	19	J01		
SM_D6	89	P12	21	J03		
SM_D7	92	V13	23	J06		
SM_LVD	94	P13	26	K03	I	SmartMedia low-voltage detect. This signal, when asserted, indicates that a 3.3V SmartMedia card is inserted in the socket. When deasserted (low), the SmartMedia card in the socket is a 5-V card.
SM_R/$\overline{\text{B}}$	99	W15	31	L03	I	SmartMedia ready/busy. This signal functions as specified in the SmartMedia specification, and is used to pace data transfers to the card.
$\overline{\text{SM\_RE}}$	105	T19	37	M06	O	SmartMedia read enable. This signal functions as specified in the SmartMedia specification, and is used to latch a read transfer from the card.
$\overline{\text{SM\_WE}}$	107	P15	40	N02	O	SmartMedia write enable. This signal functions as specified in the SmartMedia specification, and is used to latch a write transfer to the card.
$\overline{\text{SM\_WP}}$	97	R13	28	K06	O	SmartMedia write protect. This signal functions as specified in the SmartMedia specification, and is used to write-protect the card.
SQRY1	106	R17	38	M05	I	Query return terminals. These terminals are connected to the query driver terminal or to ground, to indicate the functionality of the UltraMedia card.
SQRY2	126	L15	58	W05		
SQRY3	139	H18	71	W09		
SQRY4	130	K17	61	R07		
SQRY5	132	K14	64	U07		
SQRY6	140	H17	72	V09		
SQRY7	141	H14	73	U09		
SQRY8	146	G14	77	V10		
SQRY9	149	G15	79	R10		
SQRY10	151	E19	82	V11		
SQRYDRV	123	L19	55	R06	O	Query driver terminal. This terminal is driven high by the UltraMedia controller, to determine the functionality of the UltraMedia card. See *Query Terminals*, Section 3.5.2, for details.

3 Feature/Protocol Descriptions

Figure 3−1 shows a simplified system implementation example using the PCI1620. The PCI interface includes all address/data and control signals for PCI protocol. Highlighted in this diagram is the functionality supported by the PCI1620. The PCI1620 supports $\overline{\text{PME}}$ wake-up from $D3_{cold}$ through D0, three interrupt modes, and multifunction terminals that can be programmed for a wide variety of functions.



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. PCI1620 System Block Diagram

3.1 Summary of UltraMedia Cards

3.1.1 SmartMedia

Formerly called solid-state floppy-disk card (SSFDC), SmartMedia cards are about 1/3 the area of a standard PC Card and only 0,76 mm in thickness. The specifications for SmartMedia cards are governed by the SSFDC Forum. There are two basic types of SmartMedia cards, flash memory cards and mask ROM cards. The majority of SmartMedia cards use an embedded NAND-type flash memory and are based on the *package equals card* concept. This allows the cards to be very thin, and does not require a controller to be included on the SmartMedia card.

Almost all SmartMedia cards are 3.3-V cards, but there are also 5-V versions of the 1-, 2-, and 4-Mbyte flash-memory-based cards. Additionally, all SmartMedia cards have a 22-terminal, 8-bit interface. The recommended logical format of SmartMedia cards is based on the DOS/FAT format.

SmartMedia cards are currently used in many types of consumer electronic devices and can even be incorporated in postcards that can then be accessed by a special reader. The most popular applications are in digital cameras and portable music players. The two primary methods of interfacing SmartMedia cards to current systems are through a floppy disk adapter or PCMCIA adapter.

3.1.2 MultiMediaCard (MMC)

The MultiMediaCard is a flash-memory card about the size of a postage stamp and 1,4 mm in thickness. The specification for MMC is governed by the MultiMediaCard Association (MMCA). The interface for MMC cards is based on a 7-terminal serial bus. The MultiMediaCard system specification defines a communication protocol for MMC cards, referred to as MultiMediaCard mode. In addition, all MMC cards work in the alternate SPI mode. The SPI mode allows a microcontroller to interface directly to the MMC card, but at the cost of slower performance.

The voltage range for communication with MMC cards is 2.0 to 3.6 V, and the memory-access voltage range is a card-specific subrange of the communication voltage range. Like SmartMedia cards, MMC cards can be read-only or read/write; however, MMC cards can also have I/O functionality.

MMC cards are designed to be used in either a stand-alone implementation or in a system with other MMC cards. When in the MultiMediaCard mode, the bus protocol can address cards with up to 64K of memory, and up to 30 cards on a single physical bus. However, the maximum data rate is only available with up to 10 MMC cards on the bus. In order to accommodate such a wide variety of system implementations, the MMC clock rate can be varied from 0 to 20 MHz. UltraMedia will support one MMC card per UltraMedia socket.

MMC cards, like SmartMedia cards, are also used in many types of consumer electronic devices. Because of their small size, they are primarily used in portable music players and phones.

3.1.3 Secure Digital (SD)

SD cards are the same size as MMC cards, except for the thickness, which at 2,1 mm is slightly thicker than an MMC card. SD cards are based upon MMC cards, with the addition of two terminals. The use of these two terminals and a reserved terminal on MMC cards allows the data bus on SD cards to be up to 4 bits wide instead of the 1-bit width of the MMC data bus. SD cards can communicate in either SD mode or SPI mode.

The voltage range for basic communication with SD cards is 2.0 to 3.6 V, and the voltage range for other commands and memory access is 2.7 to 3.6 V. SD cards can be read-only or read/write.

SD is essentially a superset of MMC, in that MMC cards will work in SD systems, but SD cards will not work in current MMC systems. Unlike MMC, each SD card in a system must have a dedicated bus. One of the primary benefits of SD cards is the added security that they provide. SD cards comply with the highest security of SDMI, have built-in write-protect features, and include a mechanical write-protect switch.

SD cards are used in many of the same devices as MMC cards. The additional security features of the SD cards also allow their use in more-secure applications or in devices where content protection is essential.

3.1.4 Memory Stick

Memory Stick cards are about the size of a stick of gum and are 2,8 mm thick. Developed by Sony, Memory Stick cards have a 10-terminal interface of which three terminals are used for serial communication, two terminals apply power, two terminals are ground, one terminal is for insertion detection, and two terminals are reserved for future use. Each card also includes an erasure-prevention switch to protect data stored on the card.

The voltage range for Memory Stick cards is 2.7 to 3.6 V, and the clock speed can be up to 20 MHz. Memory Stick cards use the FAT file system to allow for easy communication with PCs.

There are two types of Memory Stick cards, the standard Memory Stick and the MagicGate Memory Stick. MagicGate technology provides security to Memory Stick cards so that they can be used to store and protect copyrighted data.

Memory Stick cards are primarily used to store still images, moving images, voice and music. As such, they are used in a variety of devices, including portable music players, digital cameras, and digital picture frames.

3.1.5 Smart Card

Smart Cards, also called integrated circuit cards or ICCs, are the same size as a credit card, and they contain an embedded microprocessor chip. Smart Cards can either have contacts or be contactless. In addition, there are both asynchronous and synchronous versions of Smart Cards with contacts. UltraMedia supports asynchronous cards with contacts. Within this data manual, all use of the term Smart Card refers only to asynchronous Smart Cards with contacts.

Smart Cards contain eight contacts, however two of the contacts are reserved for future use and are not included in the UltraMedia interface. Smarts Cards can be either 5-V or 3-V cards; however, all 3-V cards are designed to work also at 5 V.

The primary use of Smart Cards is in security-related applications. They are also used in credit cards, debit systems, and identification systems.

3.2 I/O Characteristics

Figure 3−2 shows a 3-state bidirectional buffer illustration for reference. Section 8.2, *Recommended Operating Conditions*, provides the electrical characteristics of the inputs and outputs. The PCI1620 meets the ac specifications of the *PC Card Standard* and the *PCI Local Bus Specification*.

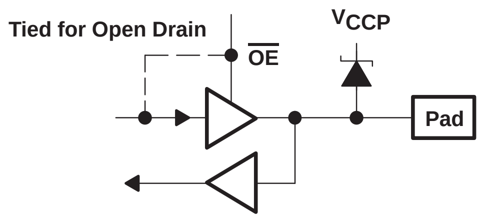


Figure 3−2. 3-State Bidirectional Buffer

3.3 Clamping Voltages

The PCI bus supports either 3.3-V or 5-V signaling. The PC Card/CardBus sockets are also capable of supporting 3.3-V or 5-V cards. The PCI1620 meets these various signaling requirements through the use of 3.3-V I/O buffers that are 5-V tolerant. These buffers output a 3.3-V signal level and can receive either 3.3-V or 5-V signals on their inputs. In addition, there are clamping diodes as shown in Figure 3−2 that limit the overshoot of the signal. The PCI1620 has three clamping-voltage terminals that should be connected to match whatever external environment the PCI1620 is interfaced with, 3.3 V or 5 V.

The PCI bus I/O terminals use the V_{CCP} terminal. If a system designer desires a 5-V PCI bus, then V_{CCP} can be connected to a 5-V power supply. Each PC Card/CardBus socket has its own clamping rail input, V_{CCA} for socket A and V_{CCB} for socket B. By connecting V_{CCA} to the voltage supply output from the external TPS222x power switch to socket A, and V_{CCB} to the voltage supply for socket B, the PCI1620 has the correct clamping-rail voltage for the card signaling levels of both PC Card/CardBus cards.

3.4 Peripheral Component Interconnect (PCI) Interface

This section describes the PCI interface of the PCI1620, and how the device responds to and participates in PCI bus cycles. The PCI1620 provides all required signals for PCI master/slave devices and may operate in either 5-V or 3.3-V PCI signaling environments by connecting the V_{CCP} terminals to the desired signaling level.

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

The bus locking protocol defined in the PCI specification is not highly recommended, but is provided on the PCI1620 as an additional compatibility feature. The use of $\overline{\text{LOCK}}$ is only supported by PCI-to-CardBus bridges in the downstream direction (away from the processor).

The PCI1620 supports all $\overline{\text{LOCK}}$ protocol associated with PCI-to-PCI bridges, as also defined for PCI-to-CardBus bridges. This includes disabling write posting while a locked operation is in progress, which can solve a potential deadlock when using devices such as PCI-to-PCI bridges. The potential deadlock can occur if a CardBus target supports delayed transactions and blocks access as the target until it completes a delayed read. This target characteristic is prohibited by the *PCI Local Bus Specification* revision 2.2, and the issue is resolved by the PCI master using $\overline{\text{LOCK}}$.

3.4.2 Serial EEPROM I²C Bus

The PCI1620 offers many choices for modes of operation, and these choices are selected by programming several configuration registers. For system board applications, these registers are normally programmed through the BIOS routine. For add-in card and docking-station/port-replicator applications, the PCI1620 provides a two-wire inter-integrated circuit (IIC or I^2C) serial bus for use with an external serial EEPROM.

The PCI1620 is always the bus master, and the EEPROM is always the slave. Either device can drive the bus low, but neither device drives the bus high. The high level is achieved through the use of pullup resistors on the SCL and SDA signal lines. The PCI1620 is always the source of the clock signal, SCL.

System designers who wish to load register values with a serial EEPROM must use a pulldown resistor on the LATCH terminal. If the PCI1620 detects a logic-low level on the LATCH terminal at the end of $\overline{\text{GRST}}$, it initiates incremental reads from the external EEPROM. Any size serial EEPROM up to the I^2C limit of 16 Kbits can be used, but only the first 42 bytes are required to configure the PCI1620. Figure 3−3 shows a 2-Kbit serial EEPROM application.

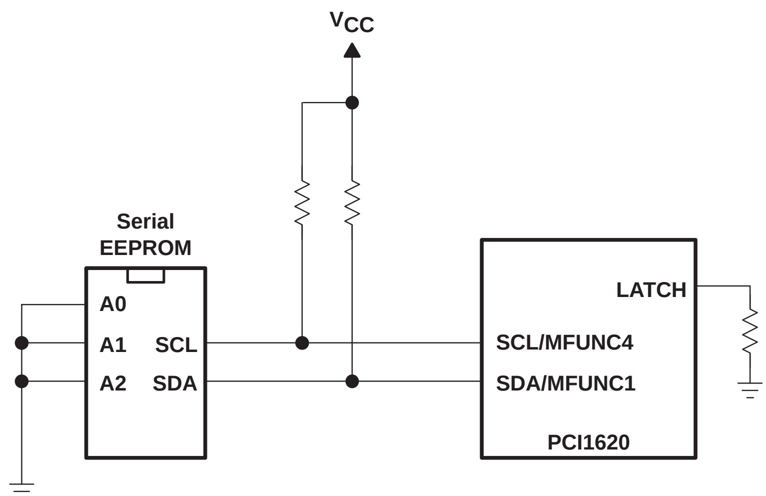


Figure 3−3. Serial EEPROM Application

In addition to loading configuration data from an EEPROM, the PCI1620 I^2C bus can be used to read and write from other I^2C serial devices. A system designer can control the I^2C bus, using the PCI1620 as bus master, by reading and writing PCI configuration registers. Setting the SBDETECT bit (bit 3) in the serial bus control/status register (PCI offset B3h, see Section 4.52) causes the PCI1620 to multiplex the SDA and SCL signals to the MFUNC1 and MFUNC4 terminals, respectively. The read/write data, slave address, and byte addresses are manipulated by accessing the serial bus data, serial bus index, and serial bus slave address registers (PCI offsets B0h, B1, and B2h; see Sections 4.49, 4.50, and 4.51, respectively).

EEPROM interface status information is communicated through the serial bus control and status register (PCI offset B3h, see Section 4.52). Bit 2 (EEDETECT) in this register indicates whether or not the PCI1620 serial EEPROM circuitry detects the pulldown resistor on LATCH. Any undefined condition, such as a missing acknowledge, results in bit 1 (DATAERR) being set. Bit 0 (EEBUSY) is set while the subsystem ID register is loading (serial EEPROM interface is busy).

3.4.3 PCI1620 EEPROM Map

The mapping of the PCI configuration, CardBus, and ExCA register bits that can be loaded from a serial EEPROM is shown in Table 3−1. The PCI 1620 starts at EEPROM address zero and continues to read incrementally the 42 bytes of data. The first byte at EEPROM address 00h is a flag byte with the value 01h. Whenever a serial EEPROM is used to load registers, all 42 bytes of data must be programmed in order, as shown in Table 3−1.

Table 3−1. Serial EEPROM Map

EEPROM OFFSET	PCI/ExCA OFFSET	REGISTER BITS LOADED FROM EEPROM
00h	Flag 00h	Flag with value 01h
01h	PCI 04h	Command register bits 8, 6−5, 2−0 Note: bits loaded as follows: b7 −−> b8 b6 −−> b6 b5 −−> b5 b4 −−> N/A b3 −−> N/A b2 −−> b2 b1 −−> b1 b0 −−> b0
02h	PCI 40h	Subsystem vendor ID byte 0
03h	PCI 41h	Subsystem vendor ID byte 1
04h	PCI 42h	Subsystem ID byte 0
05h	PCI 43h	Subsystem ID byte 1
06h	PCI 44h	PC Card 16-bit I/F legacy-mode base-address byte 0, bits 7−1
07h	PCI 45h	PC Card 16-bit I/F legacy-mode base-address byte 1
08h	PCI 46h	PC Card 16-bit I/F legacy-mode base-address byte 2
09h	PCI 47h	PC Card 16-bit I/F legacy-mode base-address byte 3
0Ah	PCI 80h	System control byte 0
0Bh	PCI 81h	System control byte 1
0Ch	PCI 82h	System control byte 2
0Dh	PCI 83h	System control byte 3
0Eh	PCI 8Ch	Multifunction routing byte 0
0Fh	PCI 8Dh	Multifunction routing byte 1
10h	PCI 8Eh	Multifunction routing byte 2
11h	PCI 8Fh	Multifunction routing byte 3
12h	PCI 90h	Retry status bits 7, 6
13h	PCI 91h	Card control bits 7, 5, 1
14h	PCI 92h	Device control bits 6, 3−0
15h	PCI 93h	Diagnostic bits 7, 4−0
16h	PCI A2h	Power management capabilities bit 15 (bit 7 of EEPROM offset 16h corresponds to bit 15)
17h	CB Socket + 0Ch (function 0)	Reserved – load all 0s
18h	CB Socket + 0Ch (function 1)	Reserved – load all 0s
19h	ExCA 00h	ExCA identification and revision bits 7−0
1Ah	PCI 86h	General control byte 0, bits 5, 4, 1, 0
1Bh	PCI 87h	General control byte 1, bits 1, 0
1Ch	PCI 89h	$\overline{\text{GPE}}$ enable, bits 7, 6, 4−0
1Dh	PCI 8Bh	General-purpose output, bits 4−0
1Eh	PCI 74h	Reserved – load all 0s
1Fh	PCI 75h	Reserved – load all 0s
20h	PCI 76h	Reserved – load all 0s

Table 3− 1. Serial EEPROM Map (Continued)

EEPROM OFFSET	PCI/ExCA OFFSET	REGISTER BITS LOADED FROM EEPROM
21h	PCI 64h	Reserved – load all 0s
22h	PCI 65h	Reserved – load all 0s
23h	PCI 66h	Reserved – load all 0s
24h	PCI 67h	Reserved – load all 0s
25h	PCI 68h	Reserved – load all 0s
26h	PCI 6Ch	Subsystem vendor ID (firmware loader function) byte 0
27h	PCI 6Dh	Subsystem vendor ID (firmware loader function) byte 1
28h	PCI 6Eh	Subsystem ID (firmware loader function) byte 0
29h	PCI 6Fh	Subsystem ID (firmware loader function) byte 1

3.4.4 Loading the Subsystem Identification (EEPROM Interface)

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

The PCI1620 offers two mechanisms to load a read-only value into the subsystem registers. The first mechanism relies upon the system BIOS providing the subsystem ID value. The default access mode to the subsystem registers is read-only, but the access mode can be made read/write by clearing the SUBSYSRW bit (bit 5) of the system control register (PCI offset 80h, see Section 4.28). Once this bit is cleared (0), the BIOS can write a subsystem identification value into the registers at offset 40h. The BIOS must set the SUBSYSRW bit such that the subsystem vendor ID register and subsystem ID register are limited to read-only access.

In some conditions, such as in a docking environment, the subsystem vendor ID register and subsystem ID register can be loaded with a unique identifier through a serial EEPROM interface. The PCI1620 loads the doubleword of data from the serial EEPROM after a reset of the primary bus. (Note that the $\overline{\text{SUSPEND}}$ input gates $\overline{\text{PRST}}$ and $\overline{\text{GRST}}$ from the entire PCI1620 core, including the serial EEPROM state machine. See Section 3.6.6, *Suspend Mode*, for details on using $\overline{\text{SUSPEND}}$.) The PCI1620 provides a two-line serial bus interface to the serial EEPROM.

The system designer must implement a pulldown resistor on the PCI1620 LATCH terminal to indicate the serial EEPROM mode. Only when this pulldown resistor is present does the PCI1620 attempt to load data through the serial EEPROM interface. Figure 3−3 illustrates a typical PCI1620 application using the serial EEPROM interface.

3.5 PC Card Applications Overview

This section describes the PC Card interfaces of the PCI1620. A discussion on PC Card recognition details the card interrogation procedure. This section discusses the card powering procedure, including the protocol of the P^2C power switch interface, and ZV routing. It also describes standard PC Card register models and briefly discusses the PC Card software protocol layers.

3.5.1 Card Detection in an UltraMedia System

The PCI1620 is capable of detecting all the UltraMedia devices defined by the PCMCIA *Proposal 0262* – Smart Media cards, MultiMedia Cards, Multimedia Card−Secure Digital, Memory Stick devices, and Smart Card devices. The detection of these devices is made possible through circuitry included in the PCI1620 and the UltraMedia Adapters used to interface these devices with the PC Card/CardBus sockets. No additional hardware requirements are placed on the system designer in order to support these devices.

The *PC Card Standard* addresses the card detection and recognition process through an interrogation procedure that the socket must initiate upon card insertion into a cold, unpowered socket. Through this interrogation, card voltage requirements and interface type (16-bit vs. CardBus) are determined. The scheme uses the CD1, CD2, VS1, and VS2 signals (CCD1, CCD2, CVS1, CVS2 for CardBus). A PC Card designer connects these four terminals in a certain configuration to indicate the type of card and its supply voltage requirements. The encoding scheme for this, defined in the *PC Card Standard*, is shown in Table 3−2.

Table 3−2. PC Card—Card Detect and Voltage Sense Connections

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

PCMCIA *Proposal 0262* has defined the first (previously) reserved response to be the indication that an UltraMedia card has been detected. Specifically, if the PCI1620 determines that the $\overline{CD1}$ signal is connected to the $\overline{VS1}$ signal, and that the $\overline{CD2}$ and $\overline{VS2}$ signals are both connected to ground, it interprets this as the insertion of an UltraMedia card adapter.

Once an insertion has been detected, the PCI1620 monitors the Media Card Detect ($\overline{MC\_CD}$) signal from the socket to determine if an UltraMedia card is present in the adapter. This ensures that UltraMedia adapter cards function the same as current adapter cards and are not detected or powered until an UltraMedia card is present.

Once $\overline{MC\_CD}$ is detected low, indicating a media card is present, the PCI1620 asserts the socket query driver signal (SQRYDRV) high and monitors the SQRY[10:1] signals to determine the UltraMedia interface type and its corresponding voltage requirements. The query signal assignments are given in Table 3−3 through Table 3−5. An example of a particular UltraMedia device, and the SQRY connections provided by the UltraMedia adapter and card, is shown in Figure 3−4.

Table 3−3. Query Terminal Definition

SQRYx TERMINAL	FUNCTION
10−7	Reserved (connect to ground)
6−3	Interface implementation
2−1	Card voltage

Table 3−4. Query Terminals − Voltage

SQRY2	SQRY1	CARD VOLTAGE
0	0	V_{CC} = 3.3 V, V_{PP}/V_{CORE} = 1.8 V
0	1	V_{CC} = 5 V, V_{PP} = 3.3 V
1	0	Reserved
1	1	Reserved

Table 3−5. Query Terminals − Media Interface Implementation

SQRY6	SQRY5	SQRY4	SQRY3	INTERFACE IMPLEMENTATION
0	0	0	0	Reserved
0	0	0	1	SmartMedia interface
0	0	1	0	MMC/SD interface
0	0	1	1	Memory stick interface
0	1	0	0	Smart card interface
0	1	0	1	Reserved
0	1	1	X	Reserved
1	X	X	X	Reserved

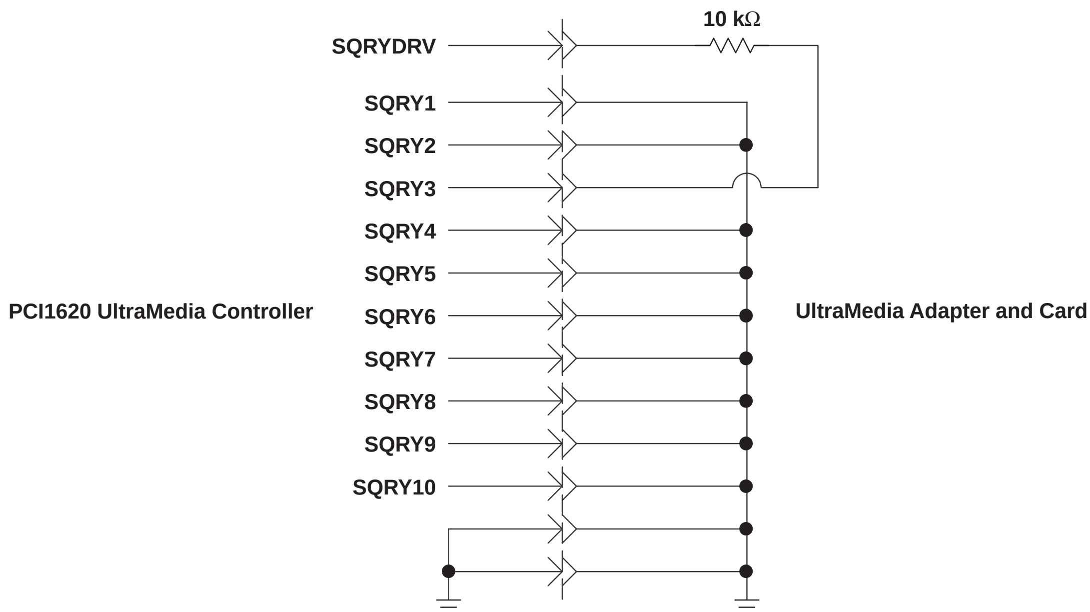


Figure 3−4. Example SmartMedia Query Terminal Configuration

When the query process has completed, the PCI1620 updates its internal registers and signals the card insertion to the host. The SQRY[10:1] terminals are switched to ground. UltraMedia devices are reported as 5-V, 16-bit cards through the socket present state register (CardBus offset 08h, see Section 6.3). The host requests that 5-V power be applied the socket, and the PCI1620 automatically overrides this request and signals the TI TPS222x power switch for the appropriate voltage levels (V_{CC} and V_{PP}) determined from the query process.

3.5.2 Query Terminals

The UltraMedia query terminal assignments and definitions are listed in Table 3−3 through Table 3−5. If a 1 value is needed for a query terminal, that terminal is connected to the query driver terminal. If a 0 value is needed for a query terminal, that terminal is connected to ground.

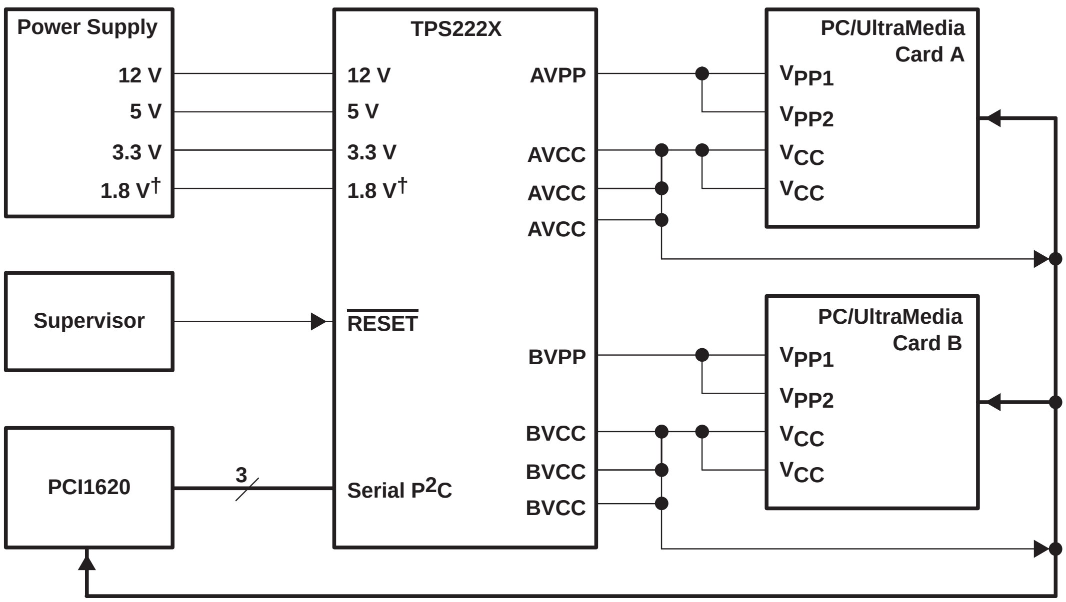
As an example, Figure 3−4 shows the query terminal configuration for a 3.3-V V_{CC} and 1.8-V V_{CORE} UltraMedia card with a SmartMedia interface.

3.5.3 P^{2}C Power Switch Interface

The PCI1620 provides a 3-wire serial PCMCIA-to-peripheral control (P^{2}C) interface for use with TI TPS222x dual-slot PC Card power-interface switches. The clock signal, CLOCK, can be derived from the PCI clock and driven by the PCI1620, or supplied from an external 32-kHz oscillator. Selection of the clock source is controlled by bit 27, P2CCLK, in the system control register (PCI offset 80h, see Section 4.31). No additional support is required to utilize the P^{2}C interface.

System designs with a requirement to provide the newer UltraMedia-compatible 1.8-V Vpp core voltage should select the TI TPS2228 switch and set the VPP1_8_SEL bit (bit 8) of the the general control register (offset 86h, see Section 4.33) to 1. For system designs with a requirement to supply the 12-V Vpp programming voltage, the TI TPS2226 power-interface switch is the best choice, and requires both VPP12_EN and VPP1_8_SEL (bits 9 and 8) of the general control register (offset 86h, see Section 4.33) to be set to 1. Furthermore, it is possible to provide 1.8 V Vpp using the TPS2226 by supplying the 12-V switch input terminal with 1.8 volts, clearing both VPP12_EN and VPP1_8_SEL (bits 9 and 8) of the general control register (offset 86h, see Section 4.33). Lastly, both the TPS2226 and TPS2228 switches are available in pin compatible (30-pin) packages that allow system designers the ability to provide for either voltage level in a single design.

Figure 3−5 illustrates a typical application using the TPS222X with the PCI1620 UltraMedia controller. (The data sheets for the individual TI TPS222x power-interface switches should be consulted for a complete overview of backward and forward compatibility.)



† UltraMedia option. 1.8-V input available on TPS2228

Figure 3−5. TPS222X Typical Application

3.5.4 Zoomed-Video Support

The PCI1620 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 4.40) 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 3−6 illustrates a PCI1620 ZV implementation.

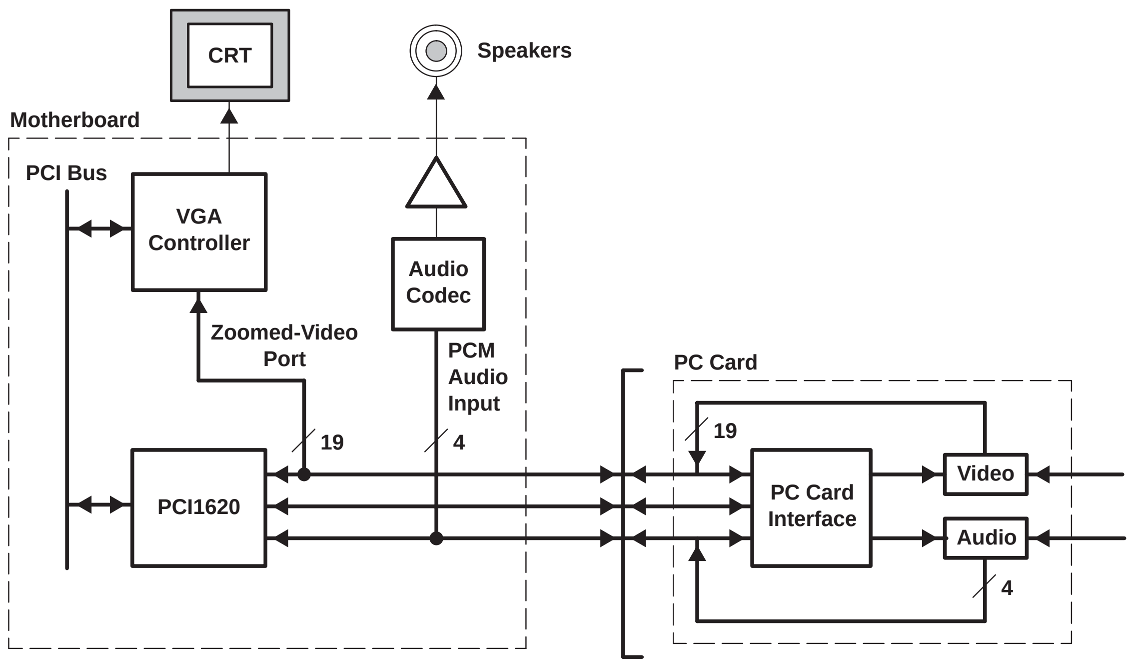


Figure 3−6. Zoomed-Video Implementation Using PCI1620

Not shown in Figure 3−6 is the multiplexing scheme used to route either socket A or socket B ZV source to the graphics controller. The PCI1620 provides ZVSTAT, $\overline{\text{ZVSEL0}}$, and $\overline{\text{ZVSEL1}}$ signals on the multifunction terminals to switch external bus drivers. Figure 3−7 shows an implementation for switching between three ZV streams using external logic.

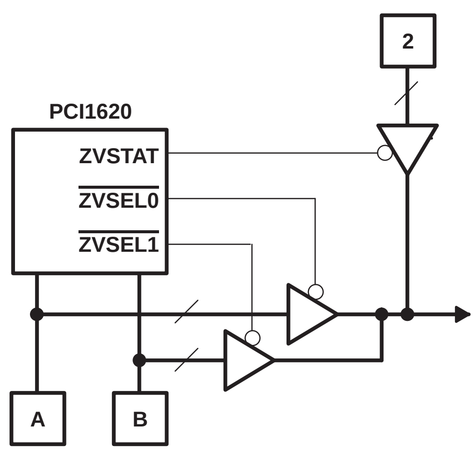


Figure 3−7. Zoomed-Video Switching Application

Figure 3−7 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 A ZV mode is enabled, and $\overline{\text{ZVSEL1}}$ is an active-low output indicating that socket B ZV is enabled. When both sockets have ZV mode enabled, the PCI1620 by defaults indicates socket A enabled through $\overline{\text{ZVSEL0}}$; however, bit 5 (PORT_SEL) in the card control register (see Section 4.40) allows software to select the socket ZV source priority. Table 3−6 illustrates the functionality of the ZV output signals.

Table 3−6. Functionality of the ZV Output Signals

INPUTS			OUTPUTS		
PORTSEL	SOCKET A ENABLE	SOCKET B 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 3−7 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 PCI1620 sockets is enabled for ZV mode. The implementation shown in Figure 3−7 can be used if PC Card ZV is prioritized over other sources.

3.5.5 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 4.42) is 1, then the standardized zoomed-video register model is disabled. For backward compatibility, even if the STDZVEN bit is 0 (enabled), the PCI1620 allows software to access zoomed video through the legacy address in the card control register (PCI offset 91h, see Section 4.40), or through the new register model in the socket control register (CardBus socket address + 10h, see Section 6.5).

3.5.6 Integrated Pullup Resistors

The *PC Card Standard* requires pullup resistors on various terminals to support both CardBus and 16-bit PC Card configurations. Table 3−7 lists these terminals. The PCI1620 has integrated all of these pullup resistors and requires no additional external components. The I/O buffer on the BVD1($\overline{\text{STSCHG}}$)/CSTSCHG terminal has the capability to switch to an internal pullup resistor when a 16-bit PC Card is inserted, or switch to an internal pulldown resistor when a CardBus card is inserted. This prevents inadvertent CSTSCHG events. The pullup resistor requirements for the various UltraMedia interfaces are either included in the UltraMedia cards (or the UltraMedia adapter) or are part of the existing PCMCIA architecture. The PCI1620 does not require any additional components for UltraMedia support.

Table 3−7. Terminals With Integrated Pullup Resistors

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

† These terminals have both pullup and pulldown resistors.

3.5.7 SPKROUT Usage

The SPKROUT terminal 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 the SPKR input terminal from the card. This terminal, in CardBus applications, is referred to as CAUDIO. SPKR passes a TTL-level binary audio signal to the PCI1620. The CardBus CAUDIO signal also can pass a single-amplitude binary waveform as well as a PWM signal. The binary audio signal from each PC Card sockets is enabled by the SPKROUTEN bit (bit 1) of the card control register (PCI offset 91h, see Section 4.40).

Older controllers support CAUDIO in binary or PWM mode but use the same output terminal (SPKROUT). Some audio chips may not support both modes on one terminal and may have a separate terminal for binary and PWM. The PCI1620 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 A audio takes precedence. See Section 4.38, *Multifunction Routing Register*, for details on configuring the MFUNC terminals.

Figure 3−8 illustrates the SPKROUT connection.

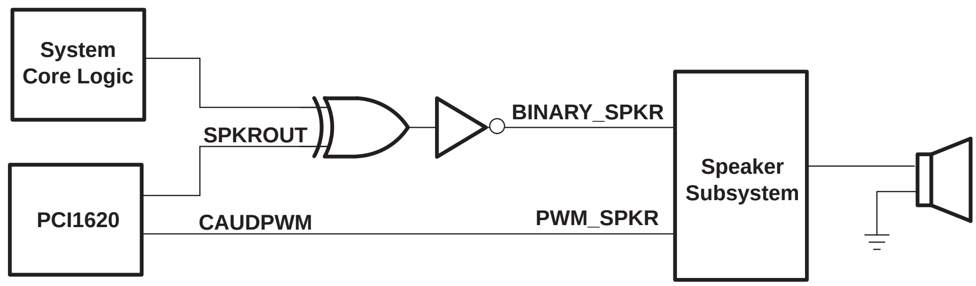


Figure 3−8. SPKROUT Connection to Speaker Driver

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 A (card A) activity, and LEDA2 indicates socket B (card B) activity. The LED_SKT output indicates socket activity to either socket A or socket B. See Section 4.38, *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 3−9 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.

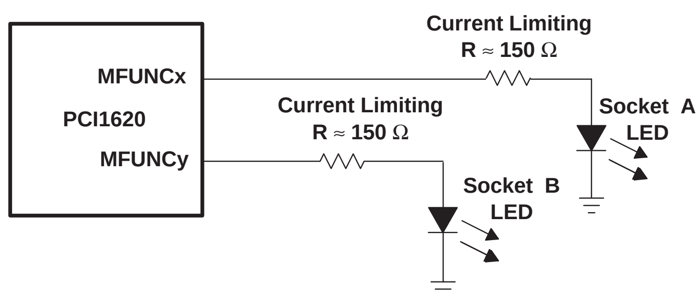


Figure 3−9. Two Sample LED Circuits

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 PCI1620 contains all registers for compatibility with *PCI Local Bus Specification* 2.2 and the *PC Card Standard*. These registers, which exist as the CardBus socket registers, are listed in Table 3−8.

Table 3−8. 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

3.5.10 PCI Firmware Loading Function Programming Model

Function 3 of UltraMedia is a firmware loader function. The purpose of this function is to provide an I/O window that a software driver uses to load the PCI1620 firmware into the internal 192K words of RAM. A simplified method of operation follows:

1. $\overline{\text{GRST}}$ assertions reset the internal RAM and the function 3 firmware loader.
2. While loading the firmware, controller holds the UltraMedia core in reset.
3. The firmware loading software driver interfaces to function 3 and loads the firmware.
4. The software driver indicates load completion to UltraMedia via the done bit (bit 2) of the firmware loader control register (offset 04h, see Section 3.5.10.2) in the function 3 I/O window.

The software driver that interfaces with PCI function 3 of the PCI1620 loads the firmware into the program RAM via the allocated I/O window for that function. Two I/O addresses are allocated, and these are used to load firmware to the PCI1620 program RAM. The functionality of these I/O registers is listed in Table 3−9.

Table 3−9. Firmware Loader I/O Register Map

REGISTER NAME	OFFSET
Data/address	00h
Firmware loader control	04h

3.5.10.1 Data/Address Register

When the ADDR_RST bit is set in the firmware loader control register (offset 04h, see Section 3.5.10.2) the next data written to this register is a doubleword that specifies the start address of the next block of internal RAM to be loaded. When the doubleword of address information is written to this field, the ADDR_RST bit is automatically cleared and the following writes to this register represent the internal RAM data. Because the internal RAM in PCI1620 is 16 bits wide, the internal RAM data written to this register is written one word at a time. The internal RAM address is autoincremented after each word of internal RAM data is written to this location. It is appropriate to buffer requests to the internal RAM and retry PCI writes to this register when the buffer is full. If the firmware loader is unable to update the RAM, a PCI slave retry time-out occurs, data is lost, and the ERR bit in the control register is set. Reads from this register return all 1s.

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

Register:	**Data address**
Offset:	00h
Type:	Read/Write
Default:	FFFF FFFFh

3.5.10.2 Firmware Loader Control Register

This register contains various control and status bits for the firmware loader. Bit descriptions are given in Table 3−10.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Firmware loader 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	Firmware loader control															
Type	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	RU
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Firmware loader control**
Offset: 04h
Type: Read-only, Write-only, Read/Update
Default: 0000 0000h

Table 3−10. Firmware Loader Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
31−4	RSVD	R	Reserved. These bits are read-only and return 0s when read.
3	ADDR_RST	W	Address reset. When set, this bit indicates that the next data written to the data/address register will be a doubleword that specifies the start address of the next block of internal RAM to be loaded. This bit is self-cleared when the address is written to the data/address register.
2	DONE	W	RAM load done. Setting this bit to 1 indicates to the firmware loader function that the firmware loading is complete for the RAM selected by the address written when ADDR_RST was set, and embedded controllers can begin accessing the RAM. This bit is self-clearing.
1	PROGRAM	W	RAM programming in progress. Setting this bit to 1 indicates to the PCI1620 that firmware loading is in progress.
0	ERR	RU	When set, this bit indicates that there was an error during the loading of the internal RAM. This field indicates all loading errors. Software should check this bit after loading each RAM to insure that the data was loaded successfully. This bit is cleared by a read of this register.

3.6 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 PCI1620. The PCI1620 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 PCI1620 is, therefore, backward compatible with existing interrupt control register definitions, and new registers have been defined where required.

The PCI1620 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 PCI1620, PC Card interrupts are classified either as card status change (CSC) or as functional interrupts.

The method by which any type of PCI1620 interrupt is communicated to the host interrupt controller varies from system to system. The PCI1620 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.

3.6.1 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 PCI1620 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 3−11 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 3−11. 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
16-bit I/O/ UltraMedia	Interrupt request ($\overline{\text{IREQ}}$)	Always enabled	PCI configuration offset 91h bit 0
All 16-bit PC Cards/ Smart Card adapters/ UltraMedia/ Flash Media	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 3−12. PC Card Interrupt Events and Description

CARD TYPE	EVENT	TYPE	SIGNAL	DESCRIPTION
16-bit memory	Battery conditions (BVD1, BVD2)	CSC	BVD1(STSCHG)//CSTSCHG	A transition on BVD1 indicates a change in the PC Card battery conditions.
			BVD2(SPKR)//CAUDIO	A transition on BVD2 indicates a change in the PC Card battery conditions.
	Wait states (READY)	CSC	READY(IREQ)//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 (STSCHG)	CSC	BVD1(STSCHG)//CSTSCHG	The assertion of STSCHG indicates a status change on the PC Card.
16-bit I/O/ UltraMedia	Interrupt request (IREQ)	Functional	READY(IREQ)//CINT	The assertion of IREQ indicates an interrupt request from the PC Card.
CardBus	Change in card status (CSTSCHG)	CSC	BVD1(STSCHG)//CSTSCHG	The assertion of CSTSCHG indicates a status change on the PC Card.
	Interrupt request (CINT)	Functional	READY(IREQ)//CINT	The assertion of CINT indicates an interrupt request from the PC Card.
All PC Cards/ Smart Card adapters/ UltraMedia/ Flash Media	Card insertion or removal	CSC	CD1//CCD1, CD2//CCD2	A transition on either CD1//CCD1 or CD2//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(IREQ)//CINT includes READY for 16-bit memory cards, IREQ for 16-bit I/O cards, and 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 PCI1620 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 PCI1620 interrupt scheme can be used to notify the host system (see Table 3−12), denoted by the power cycle complete event. This interrupt source is considered a PCI1620 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.

3.6.2 Interrupt Masks and Flags

Host software may individually mask (or disable) most of the potential interrupt sources listed in Table 3−12 by setting the appropriate bits in the PCI1620. By individually masking the interrupt sources listed, software can control those events that cause a PCI1620 interrupt. Host software has some control over the system interrupt the PCI1620 asserts by programming the appropriate routing registers. The PCI1620 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 PCI1620, the interrupt service routine must determine which of the events listed in Table 3−11 caused the interrupt. Internal registers in the PCI1620 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 3−11 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 PCI1620 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 3−11 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.

3.6.3 Using Parallel IRQ Interrupts

The seven multifunction terminals, MFUNC6−MFUNC0, implemented in the PCI1620 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 4.41), to select the parallel IRQ signaling scheme. See Section 4.38, *Multifunction Routing Register*, for details on configuring the multifunction terminals.

A system using parallel IRQs requires (at a minimum) one PCI terminal, INTA, to signal CSC events. This requirement is dictated by certain card and socket-services software. The INTA requirement calls for routing the MFUNC0 terminal for INTA signaling. The INTRTIE bit is used, in this case, to route socket B interrupt events to 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 INTA signaling and routes the remaining terminals as illustrated in Figure 3−10. Not shown is that INTA must also be routed to the programmable interrupt controller (PIC), or to some circuitry that provides parallel PCI interrupts to the host.

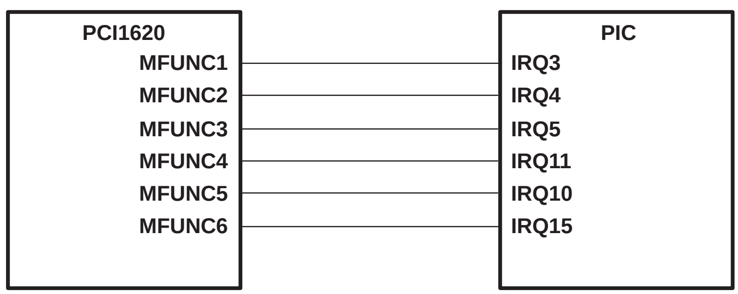


Figure 3−10. IRQ Implementation

Power-on software is responsible for programming the multifunction routing register to reflect the IRQ configuration of a system implementing the PCI1620. The multifunction routing register is shared between the two PCI1620 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 4.38, *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 PCI1620 makes available.

3.6.4 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 INTA and INTB can be routed to MFUNC terminals (MFUNC0 and MFUNC1). However, interrupts of both socket functions can be routed to INTA (MFUNC0) if bit 29 (INTRTIE) is set in the system control register (PCI offset 80h, see Section 4.31).

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 3−13 summarizes the interrupt signaling modes.

Table 3−13. Interrupt Pin Register Cross Reference

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

3.6.5 Using Serialized IRQSER Interrupts

The serialized interrupt protocol implemented in the PCI1620 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*.

3.6.6 SMI Support in the PCI1620

The PCI1620 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 PCI1620, 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 4.31). These bits are SMIROUTE (bit 26), SMISTATUS (bit 25), and SMIENB (bit 24). Table 3−14 describes the SMI control bits function.

Table 3−14. 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 4.38).

3.7 Power Management Overview

In addition to the low-power CMOS technology process used for the PCI1620, various features are designed into the device to allow implementation of popular power-saving techniques. These features and techniques are discussed in this section.

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

The PCI1620 requires 1.8-V core voltage. The core power can be supplied by the PCI1620 itself using the internal LDO-VR. The core power can alternatively be supplied by an external power supply through the VR_OUT terminal. Table 3−15 lists the requirements for both the internal core power supply and the external core power supply.

Table 3−15. Requirements for Internal/External 2.5-V Core Power Supply

SUPPLY	V_{CC}	$\overline{\text{VR\_EN}}$	VR_OUT	NOTE
Internal	3.3 V	GND	1.8-V output	Internal 1.8-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}	1.8-V input	Internal 1.8-V LDO-VR is disabled. An external 1.8-V power supply, of minimum 50-mA capacity, is required. A 0.1 μF bypass capacitor on the VR_OUT terminal is required.

3.7.2 Clock Run Protocol

The PCI CLKRUN feature is the primary method of power management on the PCI interface of the PCI1620. CLKRUN signaling is provided through the MFUNC6 terminal. Since some chip sets do not implement CLKRUN, this is not always available to the system designer, and alternate power-saving features are provided. For details on the CLKRUN protocol see the *PCI Mobile Design Guide*.

The PCI1620 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 4.31) is set.
- The 16-bit PC Card- resource manager is busy.
- The PCI1620 CardBus master state machine is busy. A cycle may be in progress on CardBus.
- The PCI1620 master is busy. There may be posted data from CardBus to PCI in the PCI1620.
- Interrupts are pending.
- The CardBus CCLK for either socket has not been stopped by the PCI1620 CCLKRUN manager.

The PCI1620 restarts the PCI clock using the CLKRUN protocol under any of the following conditions:

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

3.7.3 CardBus PC Card Power Management

The PCI1620 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 CCLKRUN interface to control this clock management.

3.7.4 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.

3.7.5 Suspend Mode

The SUSPEND signal, provided for backward compatibility, gates the PRST (PCI reset) signal and the GRST (global reset) signal from the PCI1620. Besides gating PRST and GRST, SUSPEND also gates PCLK inside the PCI1620 in order to minimize power consumption.

Gating PCLK does not create any issues with respect to the power switch interface in the PCI1620. This is because the PCI1620 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 PCI1620:

- Use an external clock to the PCI1620 CLOCK terminal
- Use the internal oscillator

It should also be noted that asynchronous signals, such as card status change interrupts and RI_OUT, can be passed to the host system without a PCI clock. However, if card status change interrupts are routed over the serial interrupt stream, then the PCI clock 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 3−11 is a signal diagram of the suspend function.

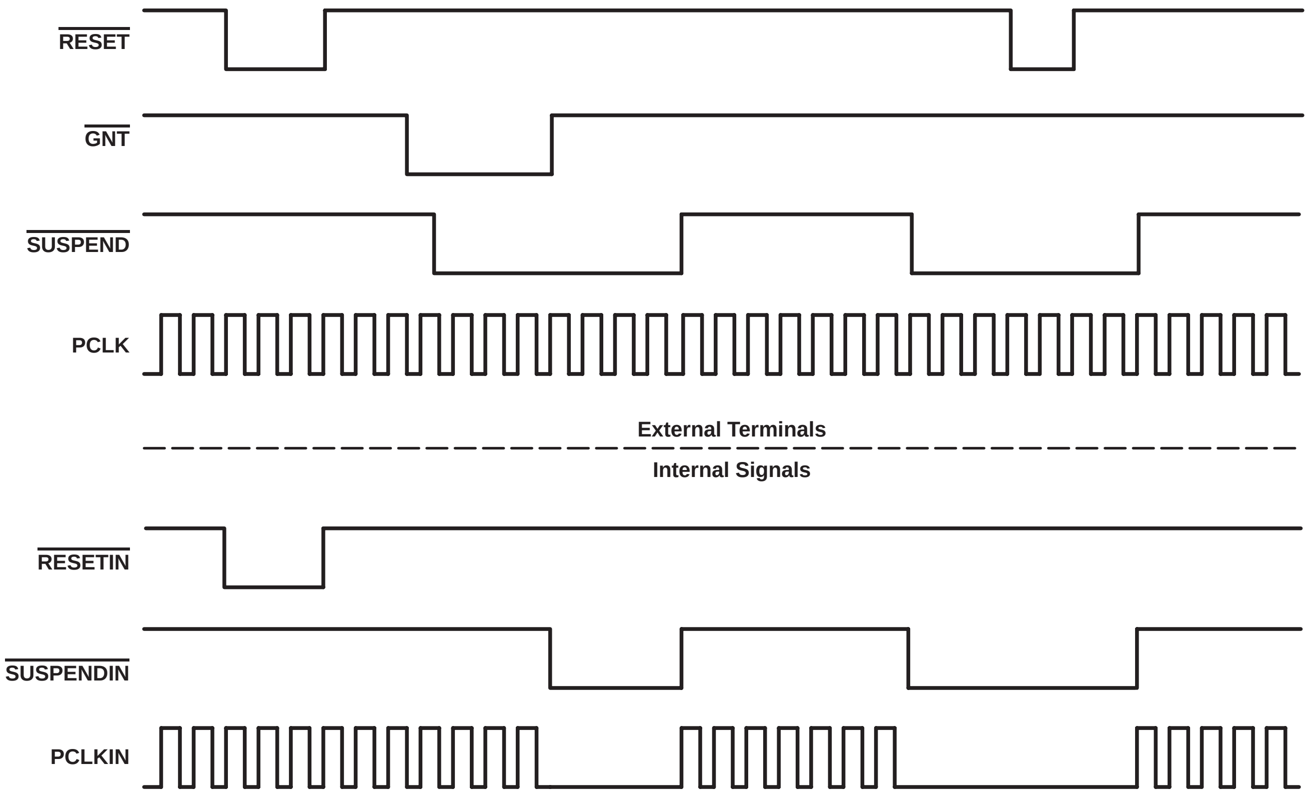


Figure 3−11. Signal Diagram of Suspend Function

3.7.6 Requirements for Suspend Mode

The suspend mode prevents the clearing of all register contents on the assertion of reset (PRST or GRST) which would require the reconfiguration of the PCI1620 by software. Asserting the 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 (GNT is asserted). It is important that the PCI bus not be parked on the PCI1620 when SUSPEND is asserted because the outputs are in a high-impedance state.

The GPIOs, MFUNC signals, and RI_OUT signal are all active during SUSPEND, unless they are disabled in the appropriate PCI1620 registers.

3.7.7 Ring Indicate

The 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. RI_OUT on the PCI1620 can be asserted under any of the following conditions:

- A 16-bit PC Card modem in a powered socket asserts 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 removal of cards or change in battery voltage levels.

Figure 3−12 shows various enable bits for the PCI1620 RI_OUT function; however, it does not show the masking of CSC events. See Table 3−11 for a detailed description of CSC interrupt masks and flags.

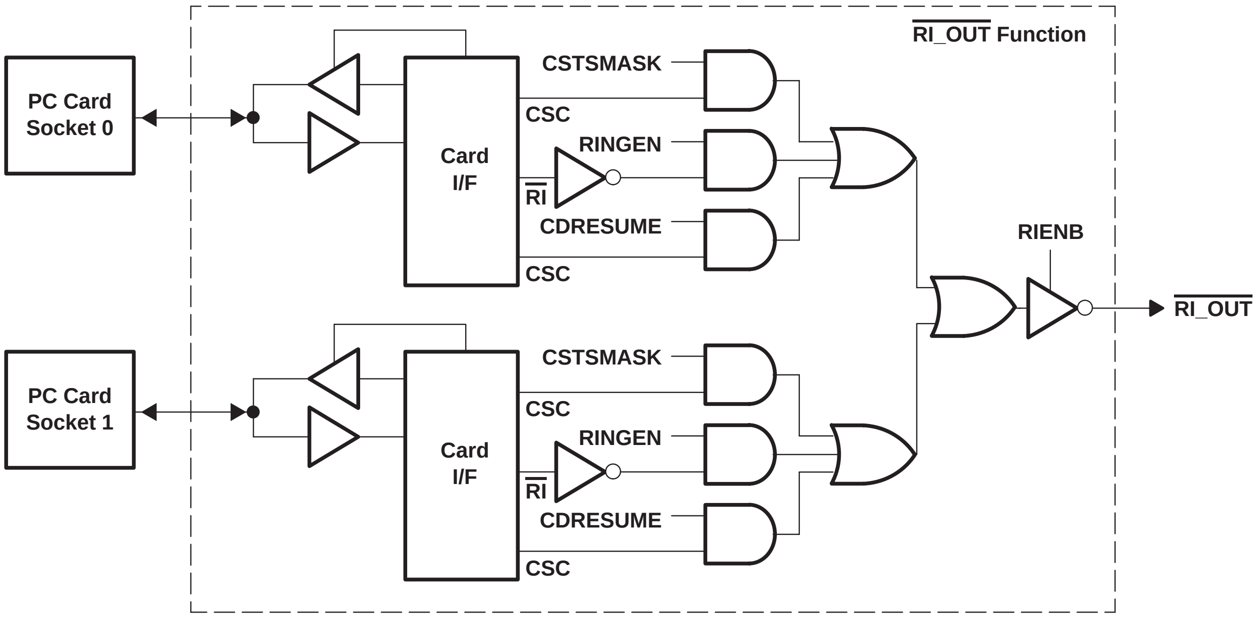


Figure 3−12. RI_OUT Functional Diagram

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 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.

RI_OUT can be routed through any of three different pins, RI_OUT/PME, MFUNC2, or MFUNC4. The RI_OUT function is enabled by setting RIENB in the card control register (PCI offset 91h, see Section 4.40). The PME function is enabled by setting PME_EN in the power management control/status register (PCI offset A4h, see Section 4.46). When RIMUX in the system control register (PCI offset 80h, see Section 4.31) is set to 0, both the RI_OUT function and the PME function are routed to the RI_OUT/PME terminal. If both functions are enabled and RIMUX is set to 0, the RI_OUT/PME terminal becomes RI_OUT only and PME assertions will never be seen. Therefore, in a system using both the RI_OUT function and the PME function, RIMUX must be set to 1 and RI_OUT must be routed to either MFUNC2 or MFUNC4.

3.7.8 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 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 non-functional

NOTE 1: In the D0-uninitialized state, the PCI1620 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 PCI1620 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 4.46) 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.

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. Each item in the list consists of 2 bytes. The first byte of each capability register block is required to be a unique ID of that capability, and the second byte is a pointer to the next capability item in the list. The next-item pointer of the last item in the list must be set to 0. For the PCI1620, a CardBus bridge with PCI configuration space header type 2, the capabilities pointer is located at PCI offset 14h, and points to the capabilities ID register (PCI offset A0h, see Section 4.43). The capabilities ID register contains a value of 01h, which is the unique ID assigned to PCI power management. Because PCI power management is the only capability in the PCI1620, the next byte, in the next item pointer register (PCI offset A1h, see Section 4.44) is 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 3−16.

Table 3−16. 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 4.45) is a static read-only register that provides information on the capabilities of the function related to power management. The power-management control/status register (PCI offset A4h, see Section 4.46) 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*.

3.7.9 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 PCI1620 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 PCI1620 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 3.7.11.
- 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 PCI1620 V_{CC} terminals.

3.7.10 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 PCI1620 offers a generic interface that is compliant with ACPI design rules.

Two doublewords of general-purpose ACPI programming bits reside in PCI1620 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 88h, see Section 4.34) and general-purpose event enable register (PCI offset 89h, see Section 4.35). The status and enable bits are implemented as defined by ACPI and illustrated in Figure 3−13.

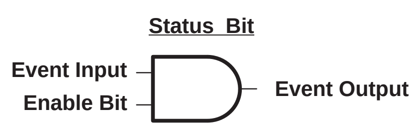


Figure 3−13. Block Diagram of a Status/Enable Cell

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.*

3.7.11 Master List of PME Context Bits and Global Reset-Only Bits

If the PME enable bit (bit 8) of the power-management control/status register (PCI offset A4h, see section 4.46) is asserted, then the assertion of PRST will not clear the following PME context bits. If the PME enable bit is not asserted, then the PME context bits are cleared with PRST. The PME context bits are:

- Bridge control register (PCI offset 3Eh, see Section 4.25): bit 6
- System control register (PCI offset 80h, see Section 4.31): bits 10, 9, 8
- Power-management control/status register (PCI offset A4h, see Section 4.46): bits 15, 8
- ExCA power control register (ExCA offset 802h, see Section 5.3): bits 7, 5†, 4–3, 1–0 († 82365SL mode only)
- ExCA interrupt and general control register (ExCA offset 803h, see Section 5.4): bits 6−5
- ExCA card status change register (ExCA offset 804h, see Section 5.5): bits 11−8, 3−0
- ExCA card status-change-interrupt configuration register (ExCA offset 805h, see Section 5.6): bits 3−0
- Socket event register (CardBus offset 00h, see Section 6.1): bits 3−0
- Socket mask register (CardBus offset 04h, see Section 6.2): bits 3−0
- Socket present state register (CardBus offset 08h, see Section 6.3): bits 13−7, 5−1
- CardBus socket control register (CardBus offset 10h, see Section 6.5): bits 6−4, 2−0

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

- Status register (PCI offset 06h, see Section 4.5): bits 15−11, 8
- Secondary status register (PCI offset 16h, see Section 4.14): bits 15−11, 8
- Interrupt pin register (PCI offset 3Dh, see Section 4.24): bits 1,0 (function 1 only)
- Subsystem vendor ID register (PCI offset 40h, see Section 4.26): bits 15–0
- Subsystem ID register (PCI offset 42h, see Section 4.27): bits 15–0
- PC Card 16-bit legacy mode base address register (PCI offset 44h, see Section 4.28): bits 31–1
- System control register (PCI offset 80h, see Section 4.31): bits 31–29, 27–13, 11, 6−0
- General-purpose event status register (PCI offset 88h, see Section 4.34): bits 15−14
- General-purpose event enable register (PCI offset 89h, see Section 4.35): bits 15−14, 11, 8, 4−0
- General-purpose output (PCI offset 8Bh, see Section 4.37): bits 4−0
- Multifunction routing status register (PCI offset 8Ch, see Section 4.38): bits 27−0
- Retry status register (PCI offset 90h, see Section 4.39): bits 7−5, 3, 1
- Card control register (PCI offset 91h, see Section 4.40): bits 7−5, 2−0
- Device control register (PCI offset 92h, see Section 4.41): bits 7−5, 3−0
- Diagnostic register (PCI offset 93h, see Section 4.42): bits 7−0
- Power management capabilities register (PCI offset A2h, see Section 4.45): bit 15
- Serial bus data (PCI offset B0h, see Section 4.49): bits 7−0
- Serial bus index (PCI offset B1h, see Section 4.50): bits 7−0
- Serial bus slave address register (PCI offset B2h, see Section 4.51): bits 7−0
- Serial bus control/status register (PCI offset B3h, see Section 4.52): bits 7, 5−0
- ExCA identification and revision register (ExCA offset 00h, see Section 5.1): bits 7−0
- ExCA global control register (ExCA offset 1Eh, see Section 5.20): bits 2−0
- Socket present state register (CardBus offset 08h, see Section 6.3): bit 29
- Socket power management register (CardBus offset 20h, see Section 6.6): bits 25−24

4 PC Card Controller Programming Model

This chapter describes the PCI1620 PCI configuration registers that make up the 256-byte PCI configuration header for each PCI1620 function. There are some bits which affect both CardBus functions, but which, in order to work properly, must be accessed only through function 0. These are called global bits. Registers containing one or more global bits are denoted by § in Table 4−1.

Any bit followed by a † is not cleared by the assertion of PRST (see *PC Card Controller Device Class Power Management Reference Specification*, http://www.microsoft.com/HWDev/specs/PMref/PMcard.htm, for more details) if PME is enabled (PCI offset A4h, bit 8). In this case, these bits are cleared only by GRST. If PME is not enabled, then these bits are cleared by GRST or PRST. These bits are sometimes referred to as PME context bits and are implemented to allow PME context to be preserved during the transition from $D3_{hot}$ or $D3_{cold}$ to D0. If the PME context PRST functionality is not desired, then the PRST and GRST signals should be tied together.

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

4.1 PCI Configuration Registers (Functions 0 and 1)

The PCI1620 is a multifunction PCI device, and the PC Card controller is integrated as PCI functions 0 and 1. The configuration header, compliant with the *PCI Local Bus Specification* as a CardBus bridge header, is *PC99/PC2001* compliant as well. Table 4−1 illustrates the PCI configuration register map, which includes both the predefined portion of the configuration space and the user-definable registers.

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

REGISTER NAME				OFFSET
Device ID		Vendor ID		00h
Status		Command		04h
Class code			Revision ID	08h
BIST	Header type	Latency timer	Cache line size	0Ch
CardBus socket registers/ExCA base address register				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−68h

† One or more bits in the register are PME context bits and can be cleared only by the assertion of GRST when PME is enabled. If PME is not enabled, then these bits are cleared by the assertion of PRST or GRST.
‡ One or more bits in this register are cleared only by the assertion of GRST.

Table 4−1. Functions 0 and 1 PCI Configuration Register Map (Continued)

REGISTER NAME				OFFSET
Subsystem ID (firmware loader function)		Subsystem vendor ID (firmware loader function)		6Ch
Reserved				70h−7Ch
System control‡§				80h
General control		Reserved	MC_CD debounce	84h
General-purpose output‡	General-purpose input‡	General-purpose event enable‡	General-purpose event status‡	88h
Multifunction routing status†				8Ch
Diagnostic‡§	Device control‡§	Card control‡§	Retry status‡§	90h
Reserved				94h−9Ch
Power management capabilities†		Next item pointer	Capability ID	A0h
Power management data (Reserved)	Power management control/status bridge support extensions	Power management control/status†		A4h
Reserved				A8h−ACh
Serial bus control/status	Serial bus slave address	Serial bus index	Serial bus data	B0h
Reserved				B4h−FCh

† One or more bits in the register are PME context bits and can be cleared only by the assertion of $\overline{\text{GRST}}$ when $\overline{\text{PME}}$ is enabled. If $\overline{\text{PME}}$ is not enabled, then these bits are cleared by the assertion of $\overline{\text{PRST}}$ or $\overline{\text{GRST}}$.
‡ One or more bits in this register are cleared only by the assertion $\overline{\text{GRST}}$.
§ One or more bits in this register are global in nature and must be accessed only through function 0.

4.2 Vendor ID Register

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

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

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

4.3 Device ID Register

The device ID register contains a value assigned to the PCI1620 by Texas Instruments. The device identification for the PCI1620 is AC54.

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	0

Register: **Device ID**
Offset: 02h (Functions 0, 1)
Type: Read-only
Default: AC54h

4.4 Command Register

The PCI command register provides control over the PCI1620 interface to the PCI bus. All bit functions adhere to the definitions in the *PCI Local Bus Specification* (see Table 4−2). None of the bit functions in this register are shared among the PCI1620 PCI functions. Three command registers exist in the PCI1620, one for each function. Software manipulates the PCI1620 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 three functions, and these control bits appear to software to be separate for each function.

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 4−2. 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 PCI1620 does not generate fast back-to-back transactions; therefore, this bit is read-only. This bit returns a 0 when read.
8	SERR_EN	RW	System error ($\overline{SERR}$) enable. This bit controls the enable for the $\overline{SERR}$ driver on the PCI interface. $\overline{SERR}$ can be asserted after detecting an address parity error on the PCI bus. Both this bit and bit 6 must be set for the PCI1620 to report address parity errors. 0 = Disables the $\overline{SERR}$ output driver (default) 1 = Enables the $\overline{SERR}$ output driver
7	STEP_EN	R	Address/data stepping control. The PCI1620 does not support address/data stepping, and this bit is hardwired to 0. Writes to this bit have no effect.
6	PERR_EN	RW	Parity error response enable. This bit controls the PCI1620 response to parity errors through the $\overline{PERR}$ signal. Data parity errors are indicated by asserting $\overline{PERR}$, while address parity errors are indicated by asserting $\overline{SERR}$. 0 = PCI1620 ignores detected parity errors (default). 1 = PCI1620 responds to detected parity errors.
5	VGA_EN	RW	VGA palette snoop. When set to 1, palette snooping is enabled (i.e., the PCI1620 does not respond to palette register writes and snoops the data). When the bit is 0, the PCI1620 treats all palette accesses like all other accesses.
4	MWI_EN	R	Memory write-and-invalidate enable. This bit controls whether a PCI initiator device can generate memory write-and-invalidate commands. The PCI1620 controller does not support memory write-and-invalidate commands, it uses memory write commands instead; therefore, this bit is hardwired to 0. This bit returns 0 when read. Writes to this bit have no effect.
3	SPECIAL	R	Special cycles. This bit controls whether or not a PCI device ignores PCI special cycles. The PCI1620 does not respond to special cycle operations; therefore, this bit is hardwired to 0. This bit returns 0 when read. Writes to this bit have no effect.
2	MAST_EN	RW	Bus master control. This bit controls whether or not the PCI1620 can act as a PCI bus initiator (master). The PCI1620 can take control of the PCI bus only when this bit is set. 0 = Disables the PCI1620 ability to generate PCI bus accesses (default) 1 = Enables the PCI1620 ability to generate PCI bus accesses
1	MEM_EN	RW	Memory space enable. This bit controls whether or not the PCI1620 can claim cycles in PCI memory space. 0 = Disables the PCI1620 response to memory space accesses (default) 1 = Enables the PCI1620 response to memory space accesses
0	IO_EN	RW	I/O space control. This bit controls whether or not the PCI1620 can claim cycles in PCI I/O space. 0 = Disables the PCI1620 from responding to I/O space accesses (default) 1 = Enables the PCI1620 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 Bus Specification*, as seen in the bit descriptions. PCI bus status is shown through each 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	Status															
Type	RW	RW	RW	RW	RW	R	R	RW	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/Write
Default: 0210h

Table 4−3. Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	PAR_ERR	RW	Detected parity error. This bit is set when a parity error is detected, either an address or data parity error. Write a 1 to clear this bit.
14	SYS_ERR	RW	Signaled system error. This bit is set when $\overline{\text{SERR}}$ is enabled and the PCI1620 signaled a system error to the host. Write a 1 to clear this bit.
13	MABORT	RW	Received master abort. This bit is set when a cycle initiated by the PCI1620 on the PCI bus has been terminated by a master abort. Write a 1 to clear this bit.
12	TABT_REC	RW	Received target abort. This bit is set when a cycle initiated by the PCI1620 on the PCI bus was terminated by a target abort. Write a 1 to clear this bit.
11	TABT_SIG	RW	Signaled target abort. This bit is set by the PCI1620 when it terminates a transaction on the PCI bus with a target abort. Write a 1 to clear this bit.
10−9	PCI_SPEED	R	DEVSEL timing. These bits encode the timing of $\overline{\text{DEVSEL}}$ and are hardwired to 01b indicating that the PCI1620 asserts this signal at a medium speed on nonconfiguration cycle accesses.
8	DATAPAR	RW	Data parity error detected. Write a 1 to clear this bit. 0 = The conditions for setting this bit have not been met. 1 = A data parity error occurred and the following conditions were met: a. $\overline{\text{PERR}}$ was asserted by any PCI device including the PCI1620. b. The PCI1620 was the bus master during the data parity error. c. The parity error response bit is set in the command register.
7	FBB_CAP	R	Fast back-to-back capable. The PCI1620 cannot accept fast back-to-back transactions; thus, this bit is hardwired to 0.
6	UDF	R	UDF supported. The PCI1620 does not support user-definable features; therefore, this bit is hardwired to 0.
5	66MHZ	R	66-MHz capable. The PCI1620 operates at a maximum PCLK frequency of 33 MHz; therefore, this bit is hardwired to 0.
4	CAPLIST	R	Capabilities list. This bit returns 1 when read. This bit indicates that capabilities in addition to standard PCI capabilities are implemented. The linked list of PCI power-management capabilities is implemented in this function.
3−0	RSVD	R	These bits return 0s when read.

4.6 Revision ID Register

The revision ID register indicates the silicon revision of the PCI1620.

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 Class Code Register

The class code register recognizes PCI1620 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 timer for the PCI1620, in units of PCI clock cycles. When the PCI1620 is a PCI bus initiator and asserts $\overline{\text{FRAME}}$, the latency timer begins counting from zero. If the latency timer expires before the PCI1620 transaction has terminated, then the PCI1620 terminates the transaction when its $\overline{\text{GNT}}$ is deasserted.

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

The header type register returns 82h when read, indicating that the PCI1620 functions 0 and 1 configuration spaces adhere to the CardBus bridge PCI header. The CardBus bridge PCI header ranges from PCI registers 00h−7Fh, and 80h−FFh is user-definable extension registers.

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

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

4.11 BIST Register

Because the PCI1620 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 Registers/ExCA Base Address Register

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

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CardBus socket registers/ExCA base address															
Type	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 registers/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 registers/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 is read-only and returns A0h when read.

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

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

4.14 Secondary Status Register

The secondary status register is compatible with the PCI-PCI bridge secondary status register. It indicates CardBus-related device information to the host system. This register is very similar to the PCI status register (PCI offset 06h, see Section 4.5), and status bits are cleared by a writing a 1. This register is not shared by the two socket functions, but is accessed on a per-socket basis. 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	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 4−4. Secondary Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	CBPARITY	RC	Detected parity error. This bit is set when a CardBus parity error is detected, either an address or data parity error. Write a 1 to clear this bit.
14	CBSERR	RC	Signaled system error. This bit is set when CSERR is signaled by a CardBus card. The PCI1620 does not assert the CSERR signal. Write a 1 to clear this bit.
13	CBMABORT	RC	Received master abort. This bit is set when a cycle initiated by the PCI1620 on the CardBus bus is terminated by a master abort. Write a 1 to clear this bit.
12	REC_CBTA	RC	Received target abort. This bit is set when a cycle initiated by the PCI1620 on the CardBus bus is terminated by a target abort. Write a 1 to clear this bit.
11	SIG_CBTA	RC	Signaled target abort. This bit is set by the PCI1620 when it terminates a transaction on the CardBus bus with a target abort. Write a 1 to clear this bit.
10−9	CB_SPEED	R	CDEVSEL timing. These bits encode the timing of CDEVSEL and are hardwired to 01b indicating that the PCI1620 asserts this signal at a medium speed.
8	CB_DPAR	RC	CardBus data parity error detected. Write a 1 to clear this bit. 0 = The conditions for setting this bit have not been met. 1 = A data parity error occurred and the following conditions were met: a. CPERR was asserted on the CardBus interface. b. The PCI1620 was the bus master during the data parity error. c. The parity error response enable bit (bit 0) is set in the bridge control register (offset 3Eh, see Section 4.25).
7	CBFBB_CAP	R	Fast back-to-back capable. The PCI1620 cannot accept fast back-to-back transactions; therefore, this bit is hardwired to 0.
6	CB_UDF	R	User-definable feature support. The PCI1620 does not support user-definable features; therefore, this bit is hardwired to 0.
5	CB66MHZ	R	66-MHz capable. The PCI1620 CardBus interface operates at a maximum CCLK frequency of 33 MHz; therefore, this bit is hardwired to 0.
4−0	RSVD	R	These bits return 0s when read.

4.15 PCI Bus Number Register

The PCI bus number register is programmed by the host system to indicate the bus number of the PCI bus to which the PCI1620 is connected. The PCI1620 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

The CardBus bus number register is programmed by the host system to indicate the bus number of the CardBus bus to which the PCI1620 is connected. The PCI1620 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 PCI1620 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

The subordinate bus number register is programmed by the host system to indicate the highest numbered bus below the CardBus bus. The PCI1620 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

The CardBus latency timer register is programmed by the host system to specify the latency timer for the PCI1620 CardBus interface, in units of CCLK cycles. When the PCI1620 is a CardBus initiator and asserts $\overline{\text{CFRAME}}$, the CardBus latency timer begins counting. If the latency timer expires before the PCI1620 transaction has terminated, then the PCI1620 terminates the transaction at the end of the next data phase. A recommended minimum value for this register of 20h allows most transactions to be completed.

Bit	7	6	5	4	3	2	1	0
Name	CardBus latency timer							
Type	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 CardBus Memory Base Registers 0, 1

These registers indicate the lower address of a PCI memory address range. They are used by the PCI1620 to determine when to forward a memory transaction to the CardBus bus, and likewise, when to forward a CardBus cycle to PCI. Bits 31−12 of these registers are read/write and allow the memory base to be located anywhere in the 32-bit PCI memory space on 4-Kbyte boundaries. Bits 11−0 are read-only and always return 0s. Writes to these bits have no effect. Bits 8 and 9 of the bridge control register (offset 3Eh, see Section 4.25) specify whether memory windows 0 and 1 are prefetchable or nonprefetchable. The memory base register or the memory limit register must be nonzero in order for the PCI1620 to claim any memory transactions through CardBus memory windows (i.e., these windows by default are not enabled 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 CardBus Memory Limit Registers 0, 1

These registers indicate the upper address of a PCI memory address range. They are used by the PCI1620 to determine when to forward a memory transaction to the CardBus bus, and likewise, when to forward a CardBus cycle to PCI. Bits 31−12 of these registers are read/write and allow the memory base to be located anywhere in the 32-bit PCI memory space on 4-Kbyte boundaries. Bits 11−0 are read-only and always return 0s. Writes to these bits have no effect. Bits 8 and 9 of the bridge control register (offset 3Eh, see Section 4.25) specify whether memory windows 0 and 1 are prefetchable or nonprefetchable. The memory base register or the memory limit register must be nonzero in order for the PCI1620 to claim any memory transactions through CardBus memory windows (i.e., these windows by default are not enabled 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 CardBus I/O Base Registers 0, 1

These registers indicate the lower address of a PCI I/O address range. They are used by the PCI1620 to determine when to forward an I/O transaction to the CardBus bus, and likewise, when to forward a CardBus cycle to the PCI bus. The lower 16 bits of this register locate the bottom of the I/O window within a 64-Kbyte page. The upper 16 bits (31−16) are all 0s, which locates this 64-Kbyte page in the first page of the 32-bit PCI I/O address space. Bits 31−16 and bits 1−0 are read-only and always return 0s, forcing I/O windows to be aligned on a natural doubleword boundary in the first 64-Kbyte page of PCI I/O address space. These I/O windows are enabled when either the I/O base register or the I/O limit register is nonzero. The I/O windows by default are not enabled to pass the first doubleword of I/O to CardBus.

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	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	I/O base registers 0, 1															
Type	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 CardBus I/O Limit Registers 0, 1

These registers indicate the upper address of a PCI I/O address range. They are used by the PCI1620 to determine when to forward an I/O transaction to the CardBus bus, and likewise, when to forward a CardBus cycle to PCI. The lower 16 bits of this register locate the top of the I/O window within a 64-Kbyte page, and the upper 16 bits are a page register which locates this 64-Kbyte page in 32-bit PCI I/O address space. Bits 15−2 are read/write and allow the I/O limit address to be located anywhere in the 64-Kbyte page (indicated by bits 31−16 of the appropriate I/O base register) on doubleword boundaries.

Bits 31−16 are read-only and always return 0s when read. The page is set in the I/O base register. Bits 1−0 are read-only and always return 0s, forcing I/O windows to be aligned on a natural doubleword boundary. Writes to read-only bits have no effect. The PCI1620 assumes that the lower 2 bits of the limit address are 1s.

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

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 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 is a read/write register used by the host software. As part of the interrupt routing procedure, the host software writes this register with the value of the system IRQ assigned to the function.

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 this register is function dependent. The default value for function 0 is 01h ($\overline{\text{INTA}}$) and the default value for function 1 is 02h ($\overline{\text{INTB}}$). The value also depends on the value of bit 29, the interrupt tie bit (INTRTIE) in the system control register (PCI offset 80h, see Section 4.31). The INTRTIE bit is compatible with previous TI CardBus controllers, and when set to 1, ties $\overline{\text{INTB}}$ to $\overline{\text{INTA}}$ internally. This results in both functions reporting interrupts through the $\overline{\text{INTA}}$ pin (01h). See Table 4−5.

PCI function 0

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

PCI function 1

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

Register: **Interrupt pin**
Offset: 3Dh
Type: Read-only
Default: 01h (function 0), 02h (function 1)

Table 4−5. Interrupt Pin Register Cross Reference

INTRTIE BIT (BIT 29, OFFSET 80H)	INTERRUPT PIN FUNCTION 0	INTERRUPT PIN FUNCTION 1
0	01h ($\overline{\text{INTA}}$)	02h ($\overline{\text{INTB}}$)
1	01h ($\overline{\text{INTA}}$)	01h ($\overline{\text{INTA}}$)

4.25 Bridge Control Register

The bridge control register provides control over various PCI1620 bridging functions. Some bits in this register are global in nature and should be accessed only through function 0. See Table 4−6 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 (Function 0, 1)
Type: Read-only, Read/Write
Default: 0340h

Table 4−6. Bridge Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
15−11	RSVD	R	These bits return 0s when read.
10	POSTEN	RW	Write posting enable. Enables write posting to and from the CardBus sockets. Write posting enables the posting of write data on burst cycles. Operating with write posting disabled impairs performance on burst cycles. Note that burst write data can be posted, but various write transactions may not. This bit is socket dependent and is not shared between functions 0 and 1.
9	PREFETCH1	RW	Memory window 1 type. This bit specifies whether or not memory window 1 is prefetchable. This bit is socket dependent. This bit is encoded as: 0 = Memory window 1 is nonprefetchable. 1 = Memory window 1 is prefetchable (default).
8	PREFETCH0	RW	Memory window 0 type. This bit specifies whether or not memory window 0 is prefetchable. This bit is socket dependent. 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. This bit is used to select whether PC Card functional interrupts are routed to PCI interrupts or to the IRQ specified in the ExCA registers. 0 = Functional interrupts are routed to PCI interrupts (default). 1 = Functional interrupts are routed by ExCA registers.
6‡	CRST	RW	CardBus reset. When this bit is set, the $\overline{\text{CRST}}$ signal is asserted on the CardBus interface. The $\overline{\text{CRST}}$ signal can also be asserted by passing a $\overline{\text{PRST}}$ assertion to CardBus. 0 = $\overline{\text{CRST}}$ is deasserted. 1 = $\overline{\text{CRST}}$ is asserted (default). This bit is not cleared by the assertion of $\overline{\text{PRST}}$. It is only cleared by the assertion of $\overline{\text{GRST}}$.
5†	MABTMODE	RW	Master abort mode. This bit controls how the PCI1620 responds to a master abort when the PCI1620 is an initiator on the CardBus interface. This bit is common between each socket. 0 = Master aborts not reported (default). 1 = Signal target abort on PCI and signal $\overline{\text{SERR}}$, if enabled.
4	RSVD	R	This bit returns 0 when read.
3	VGAEN	RW	VGA enable. This bit affects how the PCI1620 responds to VGA addresses. When this bit is set, accesses to VGA addresses will be forwarded.
2	ISAEN	RW	ISA mode enable. This bit affects how the PCI1620 passes I/O cycles within the 64-Kbyte ISA range. This bit is not common between sockets. When this bit is set, the PCI1620 does not forward the last 768 bytes of each 1K I/O range to CardBus.
1	CSERREN	RW	$\overline{\text{CSERR}}$ enable. This bit controls the response of the PCI1620 to $\overline{\text{CSERR}}$ signals on the CardBus bus. This bit is separate for each socket. 0 = $\overline{\text{CSERR}}$ is not forwarded to PCI $\overline{\text{SERR}}$ (default) 1 = $\overline{\text{CSERR}}$ is forwarded to PCI $\overline{\text{SERR}}$.

† This bit is global in nature and should be accessed only through function 0.
‡ This is a PME context bit and can be cleared only by the assertion of $\overline{\text{GRST}}$ when $\overline{\text{PME}}$ is enabled. If $\overline{\text{PME}}$ is not enabled, then these bits are cleared by the assertion of $\overline{\text{PRST}}$ or $\overline{\text{GRST}}$.

Table 4−6. Bridge Control Register Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION
0	CPERREN	RW	CardBus parity error response enable. This bit controls the response of the PCI1620 to CardBus parity errors. This bit is separate for each socket. 0 = CardBus parity errors are ignored (default). 1 = CardBus parity errors are reported using CPERR.

4.26 Subsystem Vendor ID Register

The subsystem vendor ID register, used for system and option card identification purposes, may be required for certain operating systems. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register (offset 80h, See Section 4.31). When bit 5 is 0, this register is read/write; when bit 5 is 1, this register is read-only. The default mode is read-only.

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

Register: **Subsystem vendor ID**
Offset: 40h (Functions 0, 1)
Type: Read-only, (Read/Write when bit 5 in the system control register is 0)
Default: 0000h

4.27 Subsystem ID Register

The subsystem ID register, used for system and option card identification purposes, may be required for certain operating systems. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register (offset 80h, See Section 4.31). When bit 5 is 0, this register is read/write; when bit 5 is 1, this register is read-only. The default mode is read-only.

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

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

Register: **Subsystem ID**
Offset: 42h (Functions 0, 1)
Type: Read-only, (Read/Write when bit 5 in the system control register is 0)
Default: 0000h

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

The PCI1620 supports the index/data scheme of accessing the ExCA registers, which is mapped by this register. An address written to this register is the address for the index register and the address+1 is the data address. Using this access method, applications requiring index/data ExCA access can be supported. The base address can be mapped anywhere in 32-bit I/O space on a word boundary; hence, bit 0 is read-only, returning 1 when read. As specified in the *PCI to PCMCIA CardBus Bridge Register Description* specification, this register is shared by functions 0 and 1. See the ExCA register set description in Section 5 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 Subsystem Vendor ID Register (Firmware Loader Function)

This register, used for system and option card identification purposes, may be required for certain operating systems. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register. When bit 5 is 0, this register is read/write; when bit 5 is 1, this register is read-only. The default mode is read-only. This register is provided in function 0 to allow it to be easily loaded from the EEPROM or BIOS. In the firmware loader function, read accesses to the subsystem vendor ID register (PCI offset 2Ch, see Section 7.11) are redirected to this register. This register can only be changed through function 0 and is read-only in the firmware loader function. All bits in this register are $\overline{\text{GRST}}$ only bits.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Subsystem vendor ID (firmware loader function)															
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 (firmware loader function)**
Offset: 6Ch
Type: Read-only (Read/Write when bit 5 in the system control register is 0)
Default: 0000h

4.30 Subsystem ID Register (Firmware Loader Function)

This register, used for system and option card identification purposes, may be required for certain operating systems. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register. When bit 5 is 0, this register is read/write; when bit 5 is 1, this register is read-only. The default mode is read-only. This register is provided in function 0 to allow it to be easily loaded from the EEPROM or BIOS. In the firmware loader function, read accesses to the subsystem ID register (PCI offset 2Eh, see Section 7.12) are redirected to this register. This register can only be changed through function 0 and is read-only in the firmware loader function. All bits in this register are $\overline{\text{GRST}}$ only bits.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Subsystem ID (firmware loader function)															
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 (firmware loader function)**
Offset: 6Eh
Type: Read-only (Read/Write when bit 5 in the system control register is 0)
Default: 0000h

4.31 System Control Register

System-level initializations are performed through programming this doubleword register. Some of the bits are global in nature and should be accessed only through function 0. See Table 4−7 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	System control															
Type	RW	RW	RW	RW	RW	RW	RW	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	R	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
Default: 0044 9060h

Table 4−7. System Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
31−30†	SER_STEP	RW	Serial input stepping. In serial PCI interrupt mode, these bits are used to configure the serial stream PCI interrupt frames, and can be used to accomplish an even distribution of interrupts signaled on the four PCI interrupt slots. 00 = INTA/INTB signal in INTA/INTB slots (default) 01 = INTA/INTB signal in INTB/INTC slots 10 = INTA/INTB signal in INTC/INTD slots 11 = INTA/INTB signal in INTD/INTA slots
29†	INTRTIE	RW	This bit ties INTA to INTB internally (to INTA), and reports this through the interrupt pin register (PCI offset 3Dh, see Section 4.24). This bit has no effect on INTC or INTD.
28	RSVD	R	Reserved. Bit 28 returns 0 when read.
27†	P2CCLK	RW	P2C power switch CLOCK. This bit determines whether the CLOCK terminal (PDV 154 or GHK F15) is an input that requires an external clock source or if this terminal is an output that uses the internal oscillator. Bit 27 can be set to enable the PCI1620 to generate and drive CLOCK from the PCI clock. 0 = CLOCK provided externally, input to PCI1620 (default) 1 = CLOCK generated by PCI clock and driven by PCI1620
26†	SMIROUTE	RW	SMI interrupt routing. This bit is shared between functions 0 and 1, and selects whether IRQ2 or CSC is signaled when a write occurs to power a PC Card socket. 0 = PC Card power change interrupts are routed to IRQ2 (default). 1 = A CSC interrupt is generated on PC Card power changes.
25	SMISTATUS	RW	SMI interrupt status. This socket-dependent bit is set when a write occurs to set the socket power, and the SMIENB bit is set. Writing a 1 to this bit clears the status. 0 = SMI interrupt is signaled. 1 = SMI interrupt is not signaled.
24†	SMIENB	RW	SMI interrupt mode enable. When this bit is set, the SMI interrupt signaling generates an interrupt when a write to the socket power control occurs. This bit is shared and defaults to 0 (disabled). 0 = SMI interrupt mode is disabled (default). 1 = SMI interrupt mode is enabled.

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

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

BIT	SIGNAL	TYPE	FUNCTION
23	RSVD	R	Reserved
22	CBRSVD	RW	CardBus reserved terminals signaling. When this bit is set, the RSVD CardBus terminals are driven low when a CardBus card has been inserted. When this bit is low, these signals are placed in a high-impedance state. 0 = Place the CardBus RSVD terminals in a high-impedance state. 1 = Drive the CardBus RSVD terminals low (default).
21	VCCPROT	RW	V_{CC} protection enable. This bit is socket dependent. 0 = V_{CC} protection is enabled for 16-bit cards (default). 1 = V_{CC} protection is disabled for 16-bit cards.
20	REDUCEZV	RW	Reduced zoomed-video enable. When this bit is enabled, AD25−AD22 of the card interface for 16-bit PC Cards are placed in the high impedance state. This bit is encoded as: 0 = Reduced zoomed video is disabled (default). 1 = Reduced zoomed video is enabled.
19−16	RSVD	RW	Reserved. To ensure proper device operation, do not alter the default values in these bits.
15†	MRBURSTDN	RW	Memory read burst enable downstream. When this bit is set, the PCI1620 allows memory read transactions to burst downstream. 0 = MRBURSTDN downstream is disabled. 1 = MRBURSTDN downstream is enabled (default).
14†	MRBURSTUP	RW	Memory read burst enable upstream. When this bit is set, the PCI1620 allows memory read transactions to burst upstream. 0 = MRBURSTUP upstream is disabled (default). 1 = MRBURSTUP upstream is enabled.
13	SOCACTIVE	R	Socket activity status. When set, this bit indicates access has been performed to or from a PC Card. Reading this bit causes it to be cleared. This bit is socket dependent. 0 = No socket activity (default) 1 = Socket activity
12	RSVD	R	Reserved. This bit returns 1 when read.
11	PWRSTREAM	R	Power-stream-in-progress status bit. When set, this bit indicates that a power stream to the power switch is in progress and a powering change has been requested. When this bit is cleared, it indicates that the power stream is complete. 0 = Power stream is complete, delay has expired (default). 1 = Power stream is in progress.
10	DELAYUP	R	Power-up delay-in-progress status bit. When set, this bit indicates that a power-up stream has been sent to the power switch, and proper power may not yet be stable. This bit is cleared when the power-up delay has expired. 0 = Power-up delay has expired (default). 1 = Power-up stream sent to switch. Power might not be stable.
9	DELAYDOWN	R	Power-down delay-in-progress status bit. When set, this bit indicates that a power-down stream has been sent to the power switch, and proper power may not yet be stable. This bit is cleared when the power-down delay has expired. 0 = Power-down delay has expired (default). 1 = Power-down stream sent to switch. Power might not be stable.
8	INTERROGATE	R	Interrogation in progress. When set, this bit indicates an interrogation is in progress, and clears when the interrogation completes. This bit is socket-dependent. 0 = Interrogation not in progress (default) 1 = Interrogation in progress
7	RSVD	R	Reserved. This bit returns 0 when read.
6†	PWRSAVINGS	RW	Power savings mode enable. When this bit is set, the PCI1620 consumes less power with no performance loss. This bit is shared between the two PCI1620 CardBus functions. 0 = Power savings mode disabled 1 = Power savings mode enabled (default)

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

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

BIT	SIGNAL	TYPE	FUNCTION
5†	SUBSYSRW	RW	Subsystem ID and subsystem vendor ID, ExCA ID and revision register read/write enable. This bit also controls read/write for the function 2 subsystem ID register. 0 = Registers are read/write. 1 = Registers are read-only (default).
4†	CB_DPAR	RW	CardBus data parity SERR signaling enable. 0 = CardBus data parity not signaled on PCI SERR signal (default) 1 = CardBus data parity signaled on PCI SERR signal
3	RSVD	RW	Reserved. To ensure proper device operation, do not alter the default value in this bit.
2	EXCAPOWER	R	ExCA power control bit. 0 = Enables 3.3 V (default) 1 = Enables 5 V
1†	KEEPCLK	RW	Keep clock. When this bit is set, the PCI1620 follows the CLKRUN protocol to maintain the system PCLK and the CCLK (CardBus clock). This bit is global to the PCI1620 functions. 0 = Allow system PCLK and CCLK to stop (default) 1 = Never allow system PCLK or CCLK clock to stop Note that the functionality of this bit has changed relative to that of the PCI12XX family of TI CardBus controllers. In these CardBus controllers, setting this bit only maintains the PCI clock, not the CCLK. In the PCI1620, setting this bit maintains both the PCI clock and the CCLK.
0†	RIMUX	RW	PME/RI_OUT select bit. When this bit is 1, the PME signal is routed to the PME/RI_OUT terminal (PDV 165, GHK E13). When this bit is 0 and bit 7 (RIENB) of the card control register is 1, the RI_OUT signal is routed to the PME/RI_OUT terminal (PDV 165, GHK E13). If this bit is 0 and bit 7 (RIENB) of the card control register is 0, then the output is placed in a high-impedance state. This terminal is encoded as: 0 = RI_OUT signal is routed to the PME/RI_OUT terminal (PDV 165, GHK E13) if bit 7 of the card control register is 1. (default) 1 = PME signal is routed to the PME/RI_OUT terminal (PDV 165, GHK E13) of the PCI1620 controller. NOTE: If this bit (bit 0) is 0 and bit 7 of the card control register (offset 91h, see Section 4.40) is 0, then the output on the PME/RI_OUT terminal (PDV 165, GHK E13) is placed in a high-impedance state.

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

4.32 MC_CD Debounce Register

This register provides debounce time in units of 2 ms for the MC_CD signal on UltraMedia cards. This register defaults to19h, which gives a default debounce time of 50 ms. All bits in this register are reset by GRST only.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MC_CD debounce							
Type	RW	RW	RW	RW	RW	RW	RW	RW
Default	0	0	0	1	1	0	0	1

Register: **MC_CD debounce**
Offset: 84h (Functions 0, 1)
Type: Read/Write
Default: 19h

4.33 General Control Register

The general control register provides top level PCI arbitration control. See Table 4−8 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 control															
Type	R	R	R	R	R	R	RW	RW	R	R	RW	RW	R	R	RW	RW
Default	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Register: **General control**
Offset: 86h
Type: Read/Write, Read-only
Default: 1000h

Table 4−8. General Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
15−10	RSVD	RW	These bits are for test purposes and should not be changed from their default values of 00 0100b.
9	VPP12_ENB	RW	Controls 12-V V_{PP} requests to the TPS power switch. 0 = 12-V V_{PP} requests are filtered and passed as GND V_{PP} requests (default). 1 = 12-V V_{PP} requests are passed directly to the TPS power switch.
8	VPP1_8_SEL	RW	Controls 1.8-V V_{PP} requests to the TPS power switch. 1.8-V requests are generated when either a V_{PP} request is made to an UltraMedia or CardBus card that requires 1.8-V V_{PP} per the interrogation process. 0 = 1.8-V V_{PP} requests are passed to the switch as 12-V V_{PP} requests (default). 1 = 1.8-V V_{PP} requests are passed to the switch as 1.8-V V_{PP} requests.
7−6	RSVD	R	Reserved. These bits return 0s when read.
5	DISABLE_FWL	RW	When set, the firmware loader function is completely inaccessible and nonfunctional.
4	RSVD	R	Reserved. This bit returns 0 when read.
3	DISABLE_CB_ SLOT_B	RW	When set, the second CardBus function (function 1) is inaccessible and completely non-functional. 0 = Normal operation of function 1 (default) 1 = Function 1 disabled
2	RSVD	R	Reserved. This bit returns 0 when read.
1−0	RSVD	RW	These bits are for test purposes and should not be changed from their default values of 00b.

4.34 General-Purpose Event Status Register

The general-purpose event status register contains status bits that are set when general events occur, and can be programmed to generate general-purpose event signaling through $\overline{GPE}$. See Table 4−9 for a complete description of the register contents.

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

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

Table 4−9. General-Purpose Event Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	PWR_STS	RCU	Power change status. This bit is set when software changes the V_{CC} or V_{PP} power state of either socket.
6	VPP12_STS	RCU	12-V V_{PP} request status. This bit is set when software has changed the requested V_{PP} level to or from 12 V for either socket.
5	RSVD	R	Reserved. This bit returns 0 when read. A write has no effect.
4	GP4_STS	RCU	GPI4 status. This bit is set on a change in status of the MFUNC5 terminal input level if configured as a general-purpose input, GPI4.
3	GP3_STS	RCU	GPI3 status. This bit is set on a change in status of the MFUNC4 terminal input level if configured as a general-purpose input, GPI3.
2	GP2_STS	RCU	GPI2 status. This bit is set on a change in status of the MFUNC2 terminal input level if configured as a general-purpose input, GPI2.
1	GP1_STS	RCU	GPI1 status. This bit is set on a change in status of the MFUNC1 terminal input level if configured as a general-purpose input, GPI1.
0	GP0_STS	RCU	GPI0 status. This bit is set on a change in status of the MFUNC0 terminal input level if configured as a general-purpose input, GPI0.

4.35 General-Purpose Event Enable Register

The general-purpose event enable register contains bits that are set to enable $\overline{GPE}$ signals. See Table 4−10 for a complete description of the register contents.

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

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

Table 4−10. General-Purpose Event Enable Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	PWR_EN	RW	Power change $\overline{GPE}$ enable. When this bit is set, $\overline{GPE}$ is signaled on PWR_STS events.
6	VPP12_EN	RW	12-V V_{PP} $\overline{GPE}$ enable. When this bit is set, $\overline{GPE}$ is signaled on VPP12_STS events.
5	RSVD	R	Reserved. This bit returns 0 when read. A write has no effect.
4	GP4_EN	RW	GPI4 $\overline{GPE}$ enable. When this bit is set, $\overline{GPE}$ is signaled on GP4_STS events.
3	GP3_EN	RW	GPI3 $\overline{GPE}$ enable. When this bit is set, $\overline{GPE}$ is signaled on GP3_STS events.
2	GP2_EN	RW	GPI2 $\overline{GPE}$ enable. When this bit is set, $\overline{GPE}$ is signaled on GP2_STS events.
1	GP1_EN	RW	GPI1 $\overline{GPE}$ enable. When this bit is set, $\overline{GPE}$ is signaled on GP1_STS events.
0	GP0_EN	RW	GPI0 $\overline{GPE}$ enable. When this bit is set, $\overline{GPE}$ is signaled on GP0_STS events.

4.36 General-Purpose Input Register

The general-purpose input register contains the logical value of the data input to the GPI terminals. See Table 4−11 for a complete description of the register contents.

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

Register: **General-purpose input**
Offset: 8Ah
Type: Read/Update, Read-only
Default: XXh

Table 4−11. General-Purpose Input Register Description

BIT	SIGNAL	TYPE	FUNCTION
7−5	RSVD	R	Reserved. These bits return 0s when read. Writes have no effect.
4	GPI4_DATA	RU	GPI4 data input. This bit represents the logical value of the data input from GPI4.
3	GPI3_DATA	RU	GPI3 data input. This bit represents the logical value of the data input from GPI3.
2	GPI2_DATA	RU	GPI2 data input. This bit represents the logical value of the data input from GPI2.
1	GPI1_DATA	RU	GPI1 data input. This bit represents the logical value of the data input from GPI1.
0	GPI0_DATA	RU	GPI0 data input. This bit represents the logical value of the data input from GPI0.

4.37 General-Purpose Output Register

The general-purpose output register is used to drive the GPO4−GPO0 outputs. See Table 4−12 for a complete description of the register contents.

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

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

Table 4−12. General-Purpose Output Register Description

BIT	SIGNAL	TYPE	FUNCTION
7−5	RSVD	R	Reserved. These bits return 0s when read. Writes have no effect.
4	GPO4_DATA	RW	This bit represents the logical value of the data driven to GPO4.
3	GPO3_DATA	RW	This bit represents the logical value of the data driven to GPO3.
2	GPO2_DATA	RW	This bit represents the logical value of the data driven to GPO2.
1	GPO1_DATA	RW	This bit represents the logical value of the data driven to GPO1.
0	GPO0_DATA	RW	This bit represents the logical value of the data driven to GPO0.

4.38 Multifunction Routing Status Register

The multifunction routing status register is used to configure the MFUNC6−MFUNC0 terminals. These terminals may be configured for various functions. This register is intended to be programmed once at power-on initialization. The default value for this register can also be loaded through a serial EEPROM. See Table 4−13 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 status															
Type	R	RW	RW	RW	R	RW	RW	RW	R	RW	RW	RW	R	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 status															
Type	R	RW	RW	RW	R	RW	RW	RW	R	RW	RW	RW	R	RW	RW	RW
Default	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Multifunction routing status**
Offset: 8Ch
Type: Read/Write, Read-only
Default: 0000 1000h

Table 4−13. Multifunction Routing Status 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 = CLKRUN 0101 = IRQ5 1001 = IRQ9 1101 = IRQ13 0010 = IRQ2 0110 = IRQ6 1010 = IRQ10 1110 = IRQ14 0011 = IRQ3 0111 = IRQ7 1011 = RSVD 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 = CAUDPWM 1100 = LEDA1 0001 = GPO4 0101 = IRQ5 1001 = IRQ9 1101 = LED_SKT 0010 = PCGNT 0110 = ZVSTAT 1010 = IRQ10 1110 = GPE 0011 = IRQ3 0111 = ZVSEL1 1011 = RSVD 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 (EEPROM) serial bus mode is implemented by pulling down the LATCH terminal, the SBDETECT bit in the serial bus control and status register (PCI offset B3h, see Section 4.52) is set and the MFUNC4 terminal is used to provide the SCL signalling; MFUNC4 is not available for the following signals while the SBDETECT bit is set. 0000 = GPI3 0100 = IRQ4 1000 = CAUDPWM 1100 = RI_OUT 0001 = GPO3 0101 = IRQ5 1001 = IRQ9 1101 = LED_SKT 0010 = LOCK PCI 0110 = ZVSTAT 1010 = RSVD 1110 = GPE 0011 = IRQ3 0111 = ZVSEL1 1011 = IRQ11 1111 = IRQ15
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

Table 4−13. Multifunction Routing Status Register Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION
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 = RI_OUT 0001 = GPO2 0101 = IRQ5 1001 = IRQ9 1101 = LEDA2 0010 = PCREQ 0110 = ZVSTAT 1010 = IRQ10 1110 = GPE 0011 = IRQ3 0111 = ZVSEL0 1011 = RSVD 1111 = IRQ7
7−4	MFUNC1	RW	Multifunction terminal 1 configuration. These bits control the internal signal mapped to the MFUNC1 terminal as follows: NOTE: When the (EEPROM) serial bus mode is implemented by pulling down the LATCH terminal, the SBDETECT bit in the serial bus control and status register (PCI offset B3h, see Section 4.52) is set and the MFUNC1 terminal is used to provide the SDA signalling; MFUNC1 is not available for the following signals while the SBDETECT bit is set. 0000 = GPI1 0100 = IRQ4 1000 = CAUDPWM 1100 = LEDA1 0001 = GPO1 0101 = IRQ5 1001 = IRQ9 1101 = LEDA2 0010 = INTB 0110 = ZVSTAT 1010 = IRQ10 1110 = GPE 0011 = IRQ3 0111 = 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 = GPO0 0101 = IRQ5 1001 = IRQ9 1101 = LEDA2 0010 = INTA 0110 = ZVSTAT 1010 = IRQ10 1110 = GPE 0011 = IRQ3 0111 = ZVSEL0 1011 = IRQ11 1111 = IRQ15

4.39 Retry Status Register

The contents of the retry status register enable the retry time-out counters and display the retry expiration status. The flags are set when the PCI1620, as a master, receives a retry and does not retrythe request withinn 2^{15} clock cycles. The flags are cleared by writing a 1 to the bit. Access this register only through function 0. See Table 4−14 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 4−14. Retry Status Register Description

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

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

4.40 Card Control Register

The card control register is provided for PCI1130 compatibility. RI_OUT is enabled through this register, and the enable bit is shared between functions 0 and 1. See Table 4−15 for a complete description of the register contents.

The RI_OUT signal is enabled through this register, and the enable bit is shared between functions 0 and 1.

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

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

Table 4−15. Card Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7†	RIENB	RW	Ring indicate enable. When this bit is 1, the RI_OUT output is enabled. This bit defaults to 0.
6	ZVENABLE	RW	Compatibility ZV mode enable. When this bit is 1, 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 ZV_SEL0 and ZV_SEL1 signaling if bit 6 (ZVENABLE) is set in both functions. 0 = Socket 0 takes priority, as signaled through ZV_SEL0, when both sockets are in ZV mode. 1 = Socket 1 takes priority, as signaled through ZV_SEL1, when both sockets are in ZV mode.
4−3	RSVD	R	Reserved. These bits default to 0.
2	AUD2MUX	RW	CardBus audio-to-MFUNC. When this bit is set, the CAUDIO CardBus signal is routed to the corresponding MFUNC terminal, which may be configured for CAUDWPM. When both socket 0 and socket 1 functions have AUD2MUX set, socket 0 takes precedence. 0 = CAUDIO to SPKROUT (default) 1 = CAUDIO to MFUNC
1	SPKROUTEN	RW	When bit 1 is set, the SPKR termijnal from the PC Card is enabled and is routed to tthe SPKROUT terminal. The SPKR signal from socket 0 is XORed with the 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 = SPKR to SPKROUT not enabled (default) 1 = SPKR to SPKROUT enabled
0	IFG	RW	Interrupt flag. This bit is the interrupt flag for 16-bit I/O PC Cards and for CardBus cards. This bit is set when a functional interrupt is signaled from a PC Card interface, and is socket dependent (i.e., not global). Write back a 1 to clear this bit. 0 = No PC Card functional interrupt detected (default) 1 = PC Card functional interrupt detected

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

4.41 Device Control Register

The device control register is provided for PCI1130 compatibility. It contains bits that are shared between functions 0 and 1. The interrupt mode select is programmed through this register. The socket-capable force bits are also programmed through this register. See Table 4−16 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 4−16. 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. It may be necessary to lock socket power in order 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 bit. 0 = Not 3-V capable 1 = 3-V capable (default)
5	IO16R2	RW	Diagnostic bit. This bit defaults to 1.
4	RSVD	R	Reserved. This bit returns 0 when read. A write has no effect.
3†	TEST	RW	TI test bit. Write only 0 to this bit.
2−1§	INTMODE	RW	Interrupt mode. These bits select the interrupt signaling mode. The interrupt mode bits are encoded: 00 = Parallel PCI interrupts only 01 = Reserved 10 = IRQ serialized interrupts and parallel PCI interrupts $\overline{\text{INTA}}$ and $\overline{\text{INTB}}$ 11 = IRQ and PCI serialized interrupts (default)
0†	RSVD	RW	Reserved. Bit 0 is reserved for test purposes. Only a 0 should be written to this bit.

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

4.42 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 4−17 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 4−17. 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 = Returns all 1s to reads from the PCI vendor ID and PCI device ID registers
6	RSVD	R	Reserved. This bit is read-only and 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	STDZV_EN	RW	Zoomed-video enable 0 = Enable new ZV register model (default) 1 = Disable new ZV register mode

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

4.43 Capability ID Register

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

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

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

4.44 Next Item Pointer Register

The contents of this register indicate the next item in the linked list of the PCI power management capabilities. Because the PCI1620 functions only include one capabilities item, this register returns 0s when read.

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

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

4.45 Power Management Capabilities Register

The power management capabilities register contains information on the capabilities of the PC Card function related to power management. Both PCI1620 CardBus bridge functions support D0, D1, D2, and D3 power states. Default register value is FE12h for operation in accordance with *PCI Bus Power Management Interface Specification* revision 1.1. 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	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 (Functions 0, 1)
Type: Read-only, Read/Write
Default: FE12h

Table 4−18. Power Management Capabilities Register Description

BIT	SIGNAL	TYPE	FUNCTION
	PME support		This 5-bit field indicates the power states from which the PCI1620 device functions can assert $\overline{\text{PME}}$. A 0 for any bit indicates that the function cannot assert the $\overline{\text{PME}}$ signal while in that power state. These 5 bits return 11111b when read. Each of these bits is described below:
15†		RW	Bit 15 − defaults to a 1 indicating the $\overline{\text{PME}}$ signal can be asserted from the $D3_{cold}$ state. This bit is read/write because wake-up support from $D3_{cold}$ is contingent on the system providing an auxiliary power source to the V_{CC} terminals. If the system designer chooses not to provide an auxiliary power source to the V_{CC} terminals for $D3_{cold}$ wake-up support, then BIOS should write a 0 to this bit.
14−11		R	Bit 14 − contains the value 1 to indicate that the $\overline{\text{PME}}$ signal can be asserted from the $D3_{hot}$ state. Bit 13 − contains the value 1 to indicate that the $\overline{\text{PME}}$ signal can be asserted from the D2 state. Bit 12 − contains the value 1 to indicate that the $\overline{\text{PME}}$ signal can be asserted from the D1 state. Bit 11 − contains the value 1 to indicate that the $\overline{\text{PME}}$ signal can be asserted from the D0 state.
10	D2_Support	R	This bit returns a 1 when read, indicating that the function supports the D2 device power state.
9	D1_Support	R	This bit returns a 1 when read, indicating that the function supports the D1 device power state.
8−6	RSVD	R	Reserved. These bits return 000b when read.
5	DSI	R	Device-specific initialization. This bit returns 0 when read.
4	AUX_PWR	R	Auxiliary power source. This bit is meaningful only if bit 15 ($D3_{cold}$ supporting $\overline{\text{PME}}$) is set. When this bit is set, it indicates that support for $\overline{\text{PME}}$ in $D3_{cold}$ requires auxiliary power supplied by the system by way of a proprietary delivery vehicle. A 0 (zero) in this bit field indicates that the function supplies its own auxiliary power source. If the function does not support PME while in the $D3_{cold}$ state (bit 15=0), then this field must always return 0.
3	PMECLK	R	When this bit is 1, it indicates that the function relies on the presence of the PCI clock for $\overline{\text{PME}}$ operation. When this bit is 0, it indicates that no PCI clock is required for the function to generate $\overline{\text{PME}}$. Functions that do not support PME generation in any state must return 0 for this field.
2−0	Version	R	These 3 bits return 010b when read, indicating that there are 4 bytes of general-purpose power management (PM) registers as described in draft revision 1.1 of the *PCI Bus Power Management Interface Specification*.

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

4.46 Power Management Control/Status Register

The power management control/status register determines and changes the current power state of the PCI1620 CardBus function. The contents of this register are not affected by the internally generated reset caused by the transition from the $D3_{hot}$ to D0 state. See Table 4−19 for a complete description of the register contents.

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

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Power management control/status															
Type	RWC	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/Write/Clear
Default: 0000h

Table 4−19. Power Management Control/Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15†	PMESTAT	RC	PME status. This bit is set when the CardBus function would normally assert the $\overline{PME}$ signal, independent of the state of the PME_EN bit. This bit is cleared by a 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	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_ENABLE	RW	This bit enables the function to assert $\overline{PME}$. If this bit is cleared, then assertion of $\overline{PME}$ is disabled. This bit is not cleared by the assertion of $\overline{PRST}$. It is only cleared by the assertion of $\overline{GRST}$.
7−2	RSVD	R	Reserved. These bits return 0s when read.
1−0	PWRSTATE	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 = $D3_{hot}$

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

4.47 Power Management Control/Status Bridge Support Extensions Register

This register supports PCI bridge-specific functionality. It is required for all PCI-to-PCI bridges. See Table 4−20 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Power management control/status 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 bridge support extensions**
Offset: A6h (Functions 0, 1)
Type: Read-only
Default: C0h

Table 4−20. Power Management Control/Status Bridge Support Extensions Register Description

BIT	SIGNAL	TYPE	FUNCTION
7	BPCC_EN	R	Bus power/clock control enable. This bit returns 1 when read. This bit is encoded as: 0 = Bus power/clock control is disabled. 1 = Bus power/clock control is enabled (default). A 0 indicates that the bus power/clock control policies defined in the *PCI Bus Power Management Interface Specification* are disabled. When the bus power/clock control enable mechanism is disabled, the power state field (bits 1−0) of the power management control/status register (offset A4h, see Section 4.46) cannot be used by the system software to control the power or the clock of the secondary bus. A 1 indicates that the bus power/clock control mechanism is enabled.
6	$\overline{\text{B2\_B3}}$	R	B2/B3 support for $D3_{hot}$. The state of this bit determines the action that is to occur as a direct result of programming the function to $D3_{hot}$. This bit is only meaningful if bit 7 (BPCC_EN) is a 1. This bit is encoded as: 0 = When the bridge is programmed to $D3_{hot}$, its secondary bus will have its power removed (B3). 1 = When the bridge function is programmed to $D3_{hot}$, its secondary bus PCI clock is stopped (B2) (default).
5−0	RSVD	R	Reserved. These bits return 0s when read.

4.48 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

4.49 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 (see Section 4.52) 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.

4.50 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 must be programmed with both the 7-bit slave address and the read/write indicator.

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 4.52) 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.

4.51 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 (see Section 4.52) 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.

4.52 Serial Bus Control/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/status							
Type	RW	R	R	R	RW	RW	RC	RC
Default	0	0	0	0	0	0	0	0

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

Table 4−24. Serial Bus Control/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 (see Section 4.50) is not output by the PCI1620 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 (see Section 4.51). Bit 5 must be polled on reads from the serial interface. After the byte read access has been completed, this bit is cleared and 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 PCI1620 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	RW	Serial bus detect. When bit 3 is set, it indicates that the serial bus interface is detected and the MFUNC1 and MFUNC4 terminals are reconfigured as SDA and SCL. A pulldown resistor on the LATCH terminal causes this bit to be set. This bit can also be set by writing a 1 to it. Resetting this bit to 0 allows the MFUNC4 and MFUNC1 terminals to be used for alternate functions. 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 may 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 may be set due to a missing acknowledge. Bit 0 is also set on invalid EEPROM data formats. See Section 3.4.2, *Serial EEPROM I^2C Bus*, 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 (exchangeable card architecture) registers implemented in the PCI1620 are register-compatible with the Intel 82365SL-DF PCMCIA controller. ExCA registers are identified by an offset value, which is compatible with the legacy I/O index/data scheme used on the Intel™ 82365 ISA controller. The ExCA registers are accessed through this scheme by writing the register offset value into the index register (I/O base), and reading or writing the data register (I/O base + 1). The I/O base address used in the index/data scheme is programmed in the PC Card 16-bit I/F legacy mode base address register, which is shared by both card sockets. The offsets from this base address run 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. Table 5−1 identifies each ExCA register and its respective ExCA offset.

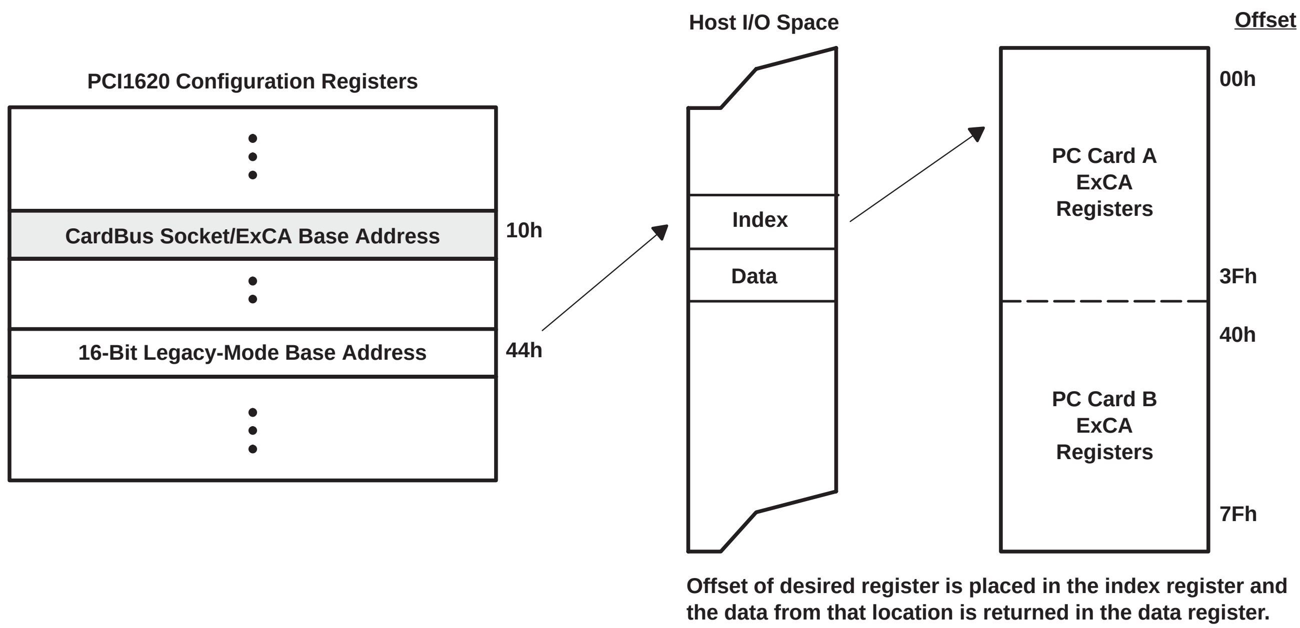
The TI PCI1620 also provides a memory-mapped alias of the ExCA registers by directly mapping them into PCI memory space. They are located through the CardBus socket registers/ExCA registers base address register (PCI register 10h) at memory offset 800h. Each socket has a separate base address programmable by function. 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 4K window at memory offset 0h.

The interrupt registers in the ExCA register set, as defined by the 82365SL 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 PCI1620 to ensure that all possible PCI1620 interrupts can potentially be routed to the programmable interrupt controller. The ExCA registers that are critical to the interrupt signaling are at memory address ExCA offsets 803h and 805h.

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

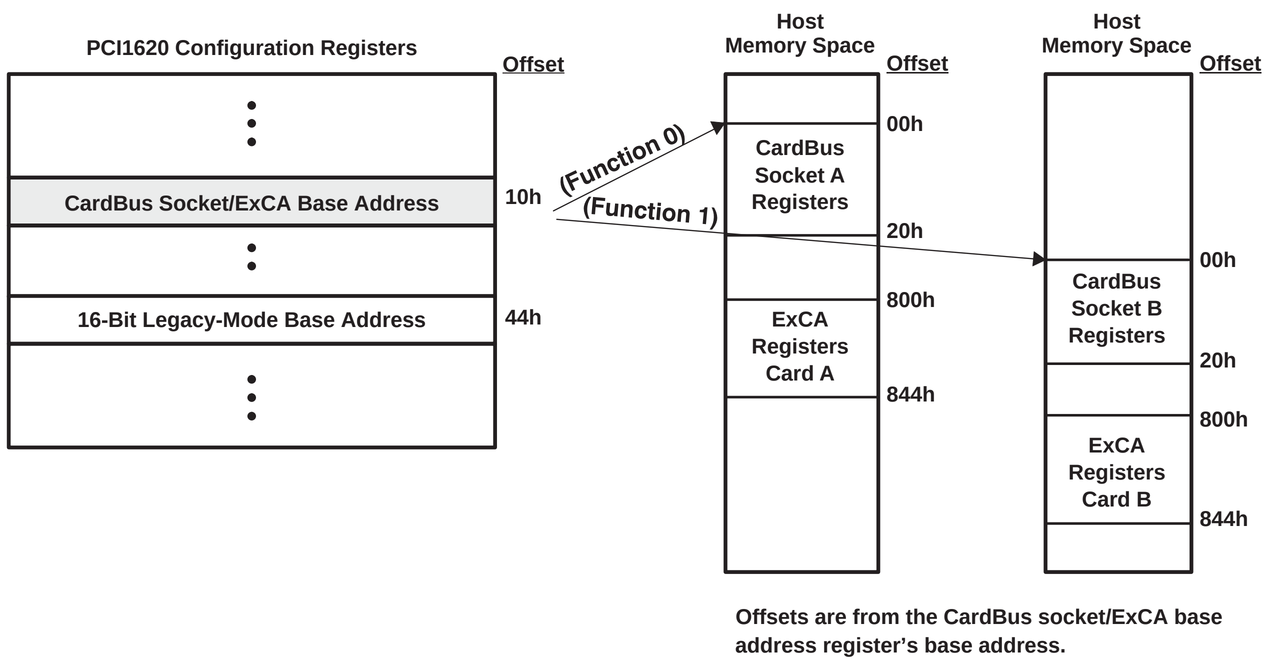
Access to memory mapped 16-bit PC Cards is available to the host system via five ExCA memory windows. These are regions of host memory space into which the card memory space is mapped. These windows are defined by start, end, and offset addresses programmed in the ExCA registers described in this section.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

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



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

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



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

Figure 5−2. ExCA Register Access Through Memory

Table 5−1. ExCA Registers and Offsets

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

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

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

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

5.1 ExCA Identification and Revision Register

This register provides host software with information on 16-bit PC Card support and 82365SL-DF compatibility. See Table 5−2 for a complete description of the register contents.

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

Bit	7	6	5	4	3	2	1	0
Name	ExCA identification and revision							
Type	R	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/Write, Read-only
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 PCI1620. The PCI1620 supports both I/O and memory 16-bit PC Cards.
5−4	RSVD	RW	These bits can be used for 82365SL emulation.
3−0	365REV	RW	82365SL-DF revision. This field stores the Intel 82365SL-DF revision supported by the PCI1620. Host software can read this field to determine compatibility to the 82365SL-DF register set. This field defaults to 0100b upon reset. Writing 0010b to this field puts the controller in the 82356SL mode.

5.2 ExCA Interface Status Register

This register provides information on current status of the PC Card interface. An X in the default bit values indicates that the value of the bit after reset depends on the state of the PC Card interface. 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	This bit returns 0 when read. A write has no effect.
6	CARDPWR	R	CARDPWR. Card power. This bit indicates the current power status of the PC Card socket. This bit reflects how the ExCA power control register has been programmed. The bit is encoded as: 0 = V_{CC} and V_{PP} to the socket are turned off (default). 1 = V_{CC} and V_{PP} to the socket are turned on.
5	READY	R	This bit indicates the current status of the READY signal at the PC Card interface. 0 = PC Card is not ready for a data transfer. 1 = PC Card is ready for a data transfer.
4	CARDWP	R	Card write protect. This bit indicates the current status of the WP signal at the PC Card interface. This signal reports to the PCI1620 whether or not the memory card is write protected. Further, write protection for an entire PCI1620 16-bit memory window is available by setting the appropriate bit in the ExCA memory window offset-address high-byte register. 0 = WP signal is 0. PC Card is R/W. 1 = WP signal is 1. PC Card is read-only.
3	CDETECT2	R	Card detect 2. This bit indicates the status of the CD2 signal at the PC Card interface. Software can use this and CDETECT1 to determine if a PC Card is fully seated in the socket. 0 = $\overline{\text{CD2}}$ signal is 1. No PC Card inserted. 1 = $\overline{\text{CD2}}$ signal is 0. PC Card at least partially inserted.
2	CDETECT1	R	Card detect 1. This bit indicates the status of the CD1 signal at the PC Card interface. Software can use this and CDETECT2 to determine if a PC Card is fully seated in the socket. 0 = $\overline{\text{CD1}}$ signal is 1. No PC Card inserted. 1 = $\overline{\text{CD1}}$ signal is 0. PC Card at least partially inserted.
1−0	BVDSTAT	R	Battery voltage detect. When a 16-bit memory card is inserted, the field indicates the status of the battery voltage detect signals (BVD1, BVD2) at the PC Card interface, where bit 0 reflects the BVD1 status, and bit 1 reflects BVD2. 00 = Battery is dead. 01 = Battery is dead. 10 = Battery is low; warning. 11 = Battery is good. When a 16-bit I/O card is inserted, this field indicates the status of the $\overline{\text{SPKR}}$ (bit 1) signal and the $\overline{\text{STSCHG}}$ (bit 0) at the PC Card interface. In this case, the two bits in this field directly reflect the current state of these card outputs.

5.3 ExCA Power Control Register

This register provides PC Card power control. Bit 7 of this register enables the 16-bit outputs on the socket interface, and can be used for power management in 16-bit PC Card applications. See 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 PCI1620. 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 4.31).
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 PCI1620 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. This bit controls the state of all of the 16-bit outputs on the PCI1620. This bit is encoded as: 0 = 16-bit PC Card outputs are disabled (default). 1 = 16-bit PC Card outputs are enabled.
6−5	RSVD	R	Reserved. These bits return 0s when read. Writes have no effect.
4−3†	EXCAVCC	RW	V_{CC}. These bits are used to request changes to card V_{CC}. This field is encoded as: 00 = 0 V (default) 01 = 0 V reserved 10 = 5 V 11 = 3.3 V
2	RSVD	R	This bit returns 0 when read. A write has no effect.
1−0†	EXCAVPP	RW	V_{PP}. These bits are used to request changes to card V_{PP}. The PCI1620 ignores this field unless V_{CC} to the socket is enabled (i.e., 5 Vdc or 3.3 Vdc). This field is encoded as: 00 = 0 V (default) 01 = V_{CC} 10 = 12 V 11 = 0 V reserved

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

5.4 ExCA Interrupt and General Control Register

This 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. Enables the ring indicate function of the BVD1/$\overline{\text{RI}}$ terminals. This bit is encoded as: 0 = Ring indicate disabled (default) 1 = Ring indicate enabled
6†	RESET	RW	Card reset. This bit controls the 16-bit PC Card RESET signal, and allows host software to force a card reset. This bit affects 16-bit cards only. This bit is encoded as: 0 = RESET signal asserted (default) 1 = RESET signal deasserted.
5†	CARDTYPE	RW	Card type. This bit indicates the PC Card type. This bit is encoded as: 0 = Memory PC Card is installed (default) 1 = I/O PC Card is installed
4	CSCROUTE	RW	PCI interrupt − CSC routing enable bit. This bit has meaning only if the CSC interrupt routing control bit (PCI offset 93h, bit 5) is 0. In this case, when this bit is set (high), the card status change interrupts are routed to PCI interrupts. When low, the card status change interrupts are routed using bits 7−4 in the ExCA card status-change interrupt configuration register (ExCA offset 805h, see Section 5.6). This bit is encoded as: 0 = CSC interrupts routed by ExCA registers (default) 1 = CSC interrupts routed to PCI interrupts If the CSC interrupt routing control bit (bit 5) of the diagnostic register (PCI offset 93h, see Section 4.42) is set to 1, this bit has no meaning, which is the default case.
3−0	INTSELECT	RW	Card interrupt select for I/O PC Card functional interrupts. These bits select the interrupt routing for I/O PC Card functional interrupts. This field is encoded as: 0000 = No IRQ selected (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 0110 = IRQ6 enabled 0111 = IRQ7 enabled 1000 = IRQ8 enabled 1001 = IRQ9 enabled 1010 = IRQ10 enabled 1011 = IRQ11 enabled 1100 = IRQ12 enabled 1101 = IRQ13 enabled 1110 = IRQ14 enabled 1111 = IRQ15 enabled

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

5.5 ExCA Card Status-Change Register

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 writeback 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 (CB offset 81Eh, 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**
Type: Read-only
Offset: CardBus socket address + 804h; Card A ExCA offset 04h
Card B ExCA offset 44h
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 PCI1620 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 PCI1620 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 PCI1620 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 PCI1420 is configured for ring indicate operation, bit 0 indicates the status of $\overline{RI}$.

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

5.6 ExCA Card Status-Change Interrupt Configuration Register

This register controls interrupt routing for CSC interrupts, as well as masks/unmasks 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 card 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. These bits select the interrupt routing for card status-change interrupts. This field is encoded as: 0000 = CSC interrupts routed to PCI interrupts if bit 5 of the diagnostic register (PCI offset 93h) is set to 1b. In this case bit 4 of ExCA 803 is a don't care. This is the default setting. 0000 = No ISA interrupt routing if bit 5 of the diagnostic register (PCI offset 93h) is set to 0b. In this case, CSC interrupts are routed to PCI interrupts by setting bit 4 of ExCA 803h to 1b. 0001 = IRQ1 enabled 0010 = SMI enabled 0011 = IRQ3 enabled 0100 = IRQ4 enabled 0101 = IRQ5 enabled 0110 = IRQ6 enabled 0111 = IRQ7 enabled 1000 = IRQ8 enabled 1001 = IRQ9 enabled 1010 = IRQ10 enabled 1011 = IRQ11 enabled 1100 = IRQ12 enabled 1101 = IRQ13 enabled 1110 = IRQ14 enabled 1111 = IRQ15 enabled
3†	CDEN	RW	Card detect enable. Enables interrupts on CD1 or CD2 changes. This bit is encoded as: 0 = Disables interrupts on CD1 or CD2 line changes (default) 1 = Enables interrupts on CD1 or CD2 line changes
2†	READYEN	RW	Ready enable. This bit enables/disables a low-to-high transition on the PC Card READY signal to generate a host interrupt. This interrupt source is considered a card status change. This bit is encoded as: 0 = Disables host interrupt generation (default) 1 = Enables host interrupt generation
1†	BATWARNEN	RW	Battery warning enable. This bit enables/disables a battery warning condition to generate a CSC interrupt. This bit is encoded as: 0 = Disables host interrupt generation (default) 1 = Enables host interrupt generation
0†	BATDEADEN	RW	Battery dead enable. This bit enables/disables a battery dead condition on a memory PC Card or assertion of the $\overline{\text{STSCHG}}$ I/O PC Card signal to generate a CSC interrupt. 0 = Disables host interrupt generation (default) 1 = Enables host interrupt generation

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

5.7 ExCA Address Window Enable Register

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 PCI1620 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**
Type: Read-only, Read/Write
Offset: CardBus socket address + 806h; Card A ExCA offset 06h
Card B ExCA offset 46h
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**
Type: Read/Write
Offset: CardBus socket address + 807h: Card A ExCA offset 07h
Card B ExCA offset 47h
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	IOSIS16W1	RW	I/O window 1 IOIS16 source. Bit 5 controls the I/O window 1 automatic data-sizing feature that uses 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 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	IOSIS16W0	RW	I/O window 0 IOIS16 source. Bit 1 controls the I/O window 0 automatic data sizing feature that uses 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 start 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	This bit controls the memory window data width. This bit is encoded as: 0 = Window data width is 8 bits (default) 1 = Window data width is 16 bits
6	ZEROWAIT	RW	Zero wait-state. This bit controls the memory window wait state for 8- and 16-bit accesses. This wait-state timing emulates the ISA wait state used by the 82365SL-DF. This bit is encoded as: 0 = 8- and 16-bit cycles have standard length (default). 1 = 8-bit cycles reduced to equivalent of three ISA cycles 16-bit cycles reduced to the equivalent of two ISA cycles
5−4	SCRATCH	RW	Scratch pad bits. These bits have no effect on memory window operation.
3−0	STAHN	RW	Start address high-nibble. These bits represent the upper address bits A23−A20 of the memory window start address.

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

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/Write, Read-only
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. These bits specify the number of equivalent ISA wait states to be added to 16-bit memory accesses. The number of wait states added is equal to the binary value of these 2 bits.
5−4	RSVD	R	Reserved. These bits return 0s when read. Writes have no effect.
3−0	ENDHN	RW	End-address high nibble. These bits represent the upper address bits A23−A20 of the memory window end address.

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

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 window 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. This bit specifies whether write operations to this memory window are enabled. This bit is encoded as: 0 = Write operations are allowed (default). 1 = Write operations are not allowed.
6	REG	RW	This bit specifies whether this memory window is mapped to card attribute or common memory. This bit is encoded as: 0 = Memory window is mapped to common memory (default). 1 = Memory window is mapped to attribute memory.
5−0	OFFHB	RW	Offset-address high byte. These bits represent the upper address bits A25−A20 of the memory window offset address.

5.19 ExCA Card Detect and General Control Register

This register controls how the ExCA registers for the socket respond to card removal. It also reports the status of the VS1 and VS2 signals at the PC Card interface. Table 5−14 describes each bit in the ExCA card detect and general control register.

Bit	7	6	5	4	3	2	1	0
Name	ExCA card detect and general control							
Type	R	R	W	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, Write-only, Read/Write	
Default:	XX00 0000b	

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

BIT	SIGNAL	TYPE	FUNCTION
7	VS2STAT	R	VS2. This bit reports the current state of the VS2 signal at the PC Card interface, and, therefore, does not have a default value. 0 = VS2 is low. 1 = VS2 is high.
6	VS1STAT	R	VS1. This bit reports the current state of the VS1 signal at the PC Card interface, and, therefore, does not have a default value. 0 = VS1 is low. 1 = VS1 is high.
5	SWCSC	W	Software card detect interrupt. If card detect enable, bit 3 in the ExCA card status change interrupt configuration register (ExCA offset 805h, see Section 5.6) set, then writing a 1 to this bit causes a card-detect card-status-change interrupt for the associated card socket. If the card-detect enable bit is cleared to 0 in the ExCA card status-change interrupt configuration register (ExCA offset 805h, see Section 5.6), then writing a 1 to the software card-detect interrupt bit has no effect. This bit is write-only. A read operation of this bit always returns 0. Writing a 1 to this bit also clears it. If bit 2 of the ExCA global control register (ExCA offset 81Eh, see Section 5.20) is set and a 1 is written to clear bit 3 of the ExCA card status change interrupt register, then this bit also is cleared.
4	CDRESUME	RW	Card detect resume enable. If this bit is set to 1 and a card detect change has been detected on the CD1 and CD2 inputs, then the RI_OUT output goes from high to low. The RI_OUT remains low until the card status change bit in the ExCA card status-change register (ExCA offset 804h, see Section 5.5) is cleared. If this bit is a 0, then the card detect resume functionality is disabled. 0 = Card detect resume disabled (default) 1 = Card detect resume enabled
3−2	RSVD	R	These bits return 0s when read. Writes have no effect.
1	REGCONFIG	RW	Register configuration upon card removal. This bit controls how the ExCA registers for the socket react to a card removal event. This bit is encoded as: 0 = No change to ExCA registers upon card removal (default) 1 = Reset ExCA registers upon card removal
0	RSVD	R	This bit returns 0 when read. A write has no effect.

5.20 ExCA Global Control Register

This register controls both PC Card sockets, and is not duplicated for each socket. The host interrupt mode bits in this register are retained for 82365SL-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	These bits return 0s when read. Writes have no effect.
4	INTMODEB	RW	Level/edge interrupt mode select, card B. This bit selects the signaling mode for the PCI1620 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. This bit selects the signaling mode for the PCI1620 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. This bit selects the interrupt flag clear mechanism for the flags in the ExCA card status change register. This bit is encoded as: 0 = Interrupt flags cleared by read of CSC register (default) 1 = Interrupt flags cleared by explicit writeback of 1
1	CSCMODE	RW	Card status change level/edge mode select. This bit selects the signaling mode for the PCI1620 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 the bit is set to 1, the PCI1620 is in power-down mode. In power-down mode the PCI1620 card outputs are placed in a high-impedance state until an active cycle is executed on the card interface. Following an active cycle the outputs are again placed in a high-impedance state. The PCI1620 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 disabled (default) 1 = Power-down mode 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/Write, Read-only
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 one of 256 16-Mbyte regions in the 4-gigabyte PCI address space. These registers are only accessible when the ExCA registers are memory-mapped, that is, these registers may not be accessed using the index/data I/O scheme.

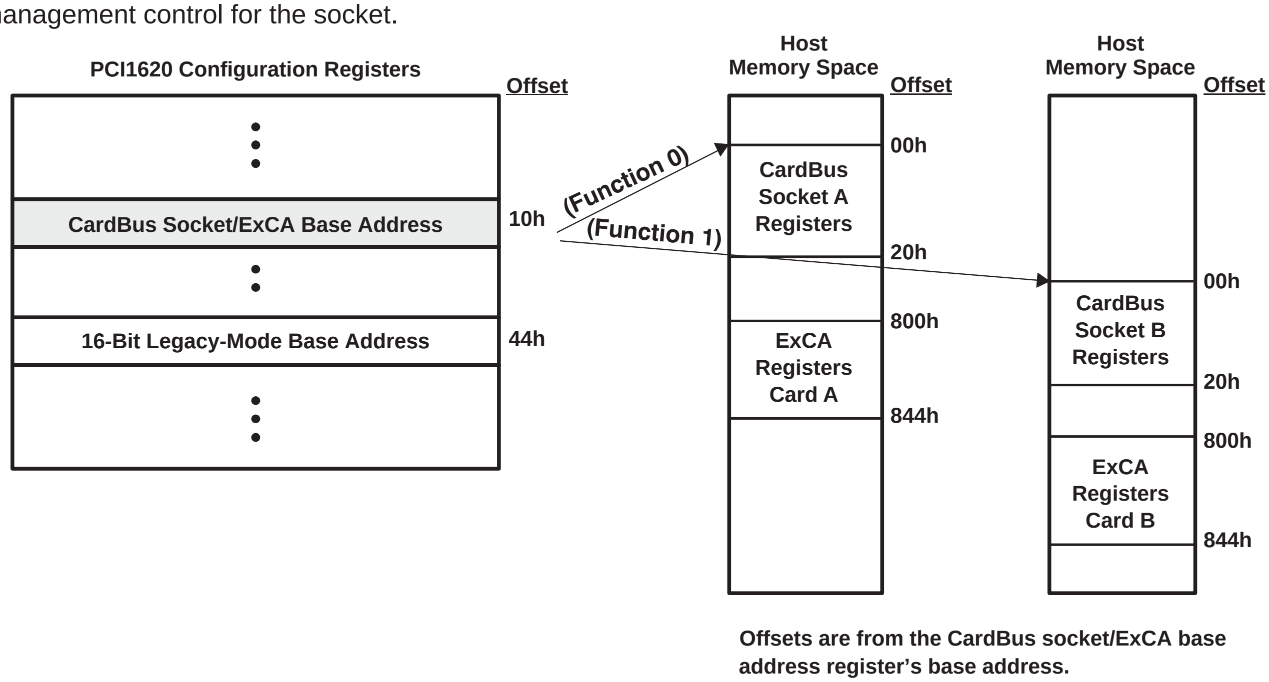
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ExCA memory windows 0−4 page							
Type	RW	RW	RW	RW	RW	RW	RW	R
Default	0	0	0	0	0	0	0	0

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 PCI1620 provides the CardBus socket/ExCA base address register (offset 10h, see Section 4.12) to locate these CardBus socket registers in PCI memory address space. Each function 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.

In addition to the five required registers, the PCI1620 implements a 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

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

6.1 Socket Event Register

This register indicates a change in socket status has occurred. These bits do not indicate what the change is, only that one has occurred. Software must read the socket present state register for current status. Each bit in this register can be cleared by writing a 1 to that bit. The bits in this register can be set to a 1 by software through writing a 1 to the corresponding bit in the socket force event register. All bits in this register are cleared by PCI reset. They can be immediately set again, if, when coming out of PC Card reset, the bridge finds the status unchanged (i.e., CSTSCHG reasserted or card detect is still true). Software needs to clear this register before enabling interrupts. If it is not cleared and interrupts are enabled, then an unmasked interrupt is generated based on any bit that is 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	RWC	RWC	RWC	RWC
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 to Clear
Default: 0000 0000h

Table 6−2. Socket Event Register Description

BIT	SIGNAL	TYPE	FUNCTION
31−4	RSVD	R	These bits return 0s when read.
3†	PWREVENT	RWC	Power cycle. This bit is set when the PCI1620 detects that the PWRCYCLE bit in the socket present state register (offset 08h, see Section 6.3) has changed. This bit is cleared by writing a 1.
2†	CD2EVENT	RWC	$\overline{\text{CCD2}}$. This bit is set when the PCI1620 detects that the CDETECT2 field in the socket present state register (offset 08h, see Section 6.3) has changed. This bit is cleared by writing a 1.
1†	CD1EVENT	RWC	$\overline{\text{CCD1}}$. This bit is set when the PCI1620 detects that the CDETECT1 field in the socket present state register (offset 08h, see Section 6.3) has changed. This bit is cleared by writing a 1.
0†	CSTSEVENT	RWC	CSTSCHG. This bit is set when the CARDSTS field in the socket present state register (offset 08h, see Section 6.3) has changed state. For CardBus cards, this bit is set on the rising edge of the CSTSCHG signal. For 16-bit PC Cards, this bit is set on both transitions of the CSTSCHG signal. This bit is reset by writing a 1.

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

6.2 Socket Mask Register

This register allows software to control the CardBus card events which generate a status change interrupt. The state of these mask bits does not prevent the corresponding bits from reacting in the socket event register (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	These bits return 0s when read.
3†	PWRMASK	RW	Power cycle. This bit masks the PWRCYCLE bit in the socket present state register (offset 08h, see Section 6.3) from causing a status change interrupt. 0 = PWRCYCLE event does not cause a CSC interrupt (default). 1 = PWRCYCLE event causes a CSC interrupt.
2−1†	CDMASK	RW	Card detect mask. These bits mask the CDETECT1 and CDETECT2 bits in the socket present state register (offset 08h, see Section 6.3) from causing a CSC interrupt. 00 = Insertion/removal does not cause a CSC interrupt (default). 01 = Reserved (undefined) 10 = Reserved (undefined) 11 = Insertion/removal causes a CSC interrupt.
0†	CSTSMASK	RW	CSTSCHG mask. This bit masks the CARDSTS field in the socket present state register (offset 08h, see Section 6.3) from causing a CSC interrupt. 0 = CARDSTS event does not cause a CSC interrupt (default). 1 = CARDSTS event causes a CSC interrupt.

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

6.3 Socket Present State Register

This register reports information about the socket interface. Writes to the socket force event register (offset 0Ch, see Section 6.4), as well as general socket interface status, are reflected here. Information about PC Card V_{CC} support and card type is only updated at each insertion. Also note that the PCI1620 uses the $\overline{\text{CCD1}}$ and $\overline{\text{CCD2}}$ signals during card identification, and changes on these signals during this operation are not reflected in this register.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Name	Socket present state															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Socket present state															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	X	0	0	0	X	X	X

Register: **Socket present state**
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. This bit indicates whether or not the socket can supply V_{CC} = Y.Y V to PC Cards. The PCI1620 does not support Y.Y-V V_{CC}; therefore, this bit is always reset unless overridden by the socket force event register (offset 0Ch, see Section 6.4). This bit defaults to 0.
30	XVSOCKET	R	XV socket. This bit indicates whether or not the socket can supply V_{CC} = X.X V to PC Cards. The PCI1620 does not support X.X-V V_{CC}; therefore, this bit is always reset unless overridden by the socket force event register (offset 0Ch, see Section 6.4). This bit defaults to 0.
29	3VSOCKET	R	3-V socket. This bit indicates whether or not the socket can supply V_{CC} = 3.3 Vdc to PC Cards. The PCI1620 does support 3.3-V V_{CC}; therefore, this bit is always set unless overridden by the socket force event register (offset 0Ch, see Section 6.4).
28	5VSOCKET	R	5-V socket. This bit indicates whether or not the socket can supply V_{CC} = 5 Vdc to PC Cards. The PCI1620 does support 5-V V_{CC}; therefore, this bit is always set unless overridden by bit 6 0f the device control register (PCI offset 92h, see Section 4.41).
27	ZVSUPPORT	R	Zoomed-video support. This bit indicates whether or not the socket has support for zoomed video. 0 = ZV support disabled 1 = ZV support enabled
26−14	RSVD	R	These bits return 0s when read.
13	YVCARD	R	YV card. This bit indicates whether or not the PC Card inserted in the socket supports V_{CC} = Y.Y Vdc. This bit can be set by writing a 1 to the corresponding bit in the socket force event register (offset 0Ch, see Section 6.4).
12	XVCARD	R	XV card. This bit indicates whether or not the PC Card inserted in the socket supports V_{CC} = X.X Vdc. This bit can be set by writing a 1 to the corresponding bit in the socket force event register (offset 0Ch, see Section 6.4).
11	3VCARD	R	3-V card. This bit indicates whether or not the PC Card inserted in the socket supports V_{CC} = 3.3 Vdc. This bit can be set by writing a 1 to the corresponding bit in the socket force event register (offset 0Ch, see Section 6.4).
10	5VCARD	R	5-V card. This bit indicates whether or not the PC Card inserted in the socket supports V_{CC} = 5 Vdc. This bit can be set by writing a 1 to the corresponding bit in the socket force event register (offset 0Ch, see Section 6.4).
9	BADVCCREQ	R	Bad V_{CC} request. This bit indicates that the host software has requested that the socket be powered at an invalid voltage. 0 = Normal operation (default) 1 = Invalid V_{CC} request by host software

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

BIT	SIGNAL	TYPE	FUNCTION
8	DATALOST	R	Data lost. This bit indicates that a PC Card removal event may have caused lost data because the cycle did not terminate properly or because write data still resides in the PCI1620. 0 = Normal operation (default) 1 = Potential data loss due to card removal
7	NOTACARD	R	Not a card. This bit indicates that an unrecognizable PC Card has been inserted in the socket. This bit is not updated until a valid PC Card is inserted into the socket. 0 = Normal operation (default) 1 = Unrecognizable PC Card detected

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

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

6.4 Socket Force Event Register

This register is used to force changes to the socket event register (offset 00h, see Section 6.1) and the socket present state register (offset 08h, see Section 6.3). The CVSTEST bit (bit 14) 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	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Socket force event															
Type	R	W	W	W	W	W	W	W	W	R	W	W	W	W	W	W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: **Socket force event**
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	These bits return 0s when read.
27	FZVSUPPORT	W	Force zoomed-video support. Writes to this bit cause the ZVSUPPORT bit in the socket present state register to be written.
26−15	RSVD	R	These bits return 0s when read.
14	CVSTEST	W	Card VS test. When this bit is set, the PCI1620 reinterrogates the PC Card, updates the socket present state register (offset 08h, see Section 6.3), and re-enables the socket power control.
13	FYVCARD	W	Force YV card. Writes to this bit cause the YVCARD bit in the socket present state register (offset 08h, see Section 6.3) to be written. When set, this bit disables the socket power control.
12	FXVCARD	W	Force XV card. Writes to this bit cause the XVCARD bit in the socket present state register (offset 08h, see Section 6.3) to be written. When set, this bit disables the socket power control.
11	F3VCARD	W	Force 3-V card. Writes to this bit cause the 3VCARD bit in the socket present state register (offset 08h, see Section 6.3) to be written. When set, this bit disables the socket power control.
10	F5VCARD	W	Force 5-V card. Writes to this bit cause the 5VCARD bit in the socket present state register (offset 08h, see Section 6.3) to be written. When set, this bit disables the socket power control.
9	FBADVCCREQ	W	Force BadVccReq. Changes to the BADVCCREQ bit in the socket present state register (offset 08h, see Section 6.3) can be made by writing this bit.
8	FDATALOST	W	Force data lost. Writes to this bit cause the DATALOST bit in the socket present state register (offset 08h, see Section 6.3) to be written.
7	FNOTACARD	W	Force not a card. Writes to this bit cause the NOTACARD bit in the socket present state register (offset 08h, see Section 6.3) to be written.
6	RSVD	R	This bit returns 0 when read.
5	FCBCARD	W	Force CardBus card. Writes to this bit cause the CBCARD bit in the socket present state register (offset 08h, see Section 6.3) to be written.
4	F16BITCARD	W	Force 16-bit card. Writes to this bit cause the 16BITCARD bit in the socket present state register (offset 08h, see Section 6.3) to be written.
3	FPWRCYCLE	W	Force power cycle. Writes to this bit cause the PWREVENT bit in the socket event register (offset 00h, see Section 6.1) to be written, and the PWRCYCLE bit in the socket present state register (offset 08h, see Section 6.3) is unaffected.
2	FCDETECT2	W	Force $\overline{\text{CCD2}}$. Writes to this bit cause the CD2EVENT bit in the socket event register (offset 00h, see Section 6.1) to be written, and the CDETECT2 bit in the socket present state register (offset 08h, see Section 6.3) is unaffected.
1	FCDETECT1	W	Force $\overline{\text{CCD1}}$. Writes to this bit cause the CD1EVENT bit in the socket event register (offset 00h, see Section 6.1) to be written, and the CDETECT1 bit in the socket present state register (offset 08h, see Section 6.3) is unaffected.
0	FCARDSTS	W	Force CSTSCHG. Writes to this bit cause the CSTSEVENT bit in the socket event register (offset 00h, see Section 6.1) to be written. The CARDSTS bit in the socket present state register (offset 08h, see Section 6.3) is unaffected.

6.5 Socket Control Register

This register provides control of the voltages applied to the socket V_{PP} and V_{CC}. The PCI1620 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	0	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 0000h

Table 6−6. Socket Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
31−12	RSVD	R	These bits return 0s when read.
11	ZV_ACTIVITY	R	This bit returns 0 when the ZVEN bits (bit 0) for both sockets are 0 (disabled). If either ZVEN bit is set to 1, the ZV_ACTIVITY bit returns 1.
10	STANDARDZVREG	R	Standardized zoomed-video register model supported. Because PCI1620 supports this register model, this bit is hardwired to 1.
9	ZVEN	RW	Zoomed-video enable. This bit enables zoomed video for the socket.
8	RSVD	R	These bits return 0s when read.
7	STOPCLK	RW	This bit controls how the CardBus clock run state machine decides when to stop the CardBus clock to the CardBus card: 0 = The CardBus $\overline{\text{CLKRUN}}$ protocol can only attempt to stop/slow the CaredBus clock if the sockethas been idle for 8 clocks and the PCI $\overline{\text{CLKRUN}}$ protocol is preparing to stop/slow the PCI bus clock. 1 = The CardBus $\overline{\text{CLKRUN}}$ protocol can only attempt to stop/slow the CaredBus clock if the sockethas been idle for 8 clocks, regardless of the state of the PCI $\overline{\text{CLKRUN}}$ signal.
6−4†	VCCCTRL	RW	V_{CC} control. These bits are used to request card V_{CC} changes. 000 = Request power off (default) 001 = Reserved 010 = Request V_{CC} = 5 V 011 = Request V_{CC} = 3.3 V 100 = Request V_{CC} = X.X V 101 = Request V_{CC} = Y.Y V 110 = Reserved 111 = Reserved
3	RSVD	R	This bit returns 0 when read.

Table 6−6. Socket Control Register Description (Continued)

BIT	SIGNAL	TYPE	FUNCTION
2−0†	VPPCTRL	RW	V_{PP} control. These bits are used to request card V_{PP} changes. 000 = Request power off (default) 001 = Request V_{PP} = 12 V 010 = Request V_{PP} = 5 V 011 = Request V_{PP} = 3.3 V 100 = Request V_{PP} = X.X V 101 = Request V_{PP} = Y.Y V 110 = Reserved 111 = Reserved

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

6.6 Socket Power Management Register

This register provides power management control over the socket through a mechanism for slowing or stopping the clock on the card interface when the card is idle. 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**
Offset: CardBus Socket Address + 20h
Type: Read-only, Read/Write
Default: 0000 0000h

Table 6−7. Socket Power Management Register Description

BIT	SIGNAL	TYPE	FUNCTION
31−26	RSVD	R	Reserved. These bits return 0s when read.
25	SKTACCES	R	Socket access status. This bit provides information on whether a socket access has occurred. This bit is cleared by a read access. 0 = No PC Card access has occurred (default). 1 = PC Card has been accessed.
24	SKTMODE	R	Socket mode status. This bit provides clock mode information. 0 = Normal clock operation 1 = Clock frequency has changed.
23−17	RSVD	R	These bits return 0s when read.
16	CLKCTRLEN	RW	CardBus clock control enable. This bit, when set, enables clock control according to bit 0 (CLKCTRL). 0 = Clock control disabled (default) 1 = Clock control enabled
15−1	RSVD	R	These bits return 0s when read.
0	CLKCTRL	RW	CardBus clock control. This bit determines whether the CardBus $\overline{\text{CLKRUN}}$ protocol attempts to stop or slow the CardBus clock during idle states. The CLKCTRLEN bit enables this bit. 0 = Allows the CardBus $\overline{\text{CLKRUN}}$ protocol to attempt to stop the CardBus clock (default) 1 = Allows the CardBus $\overline{\text{CLKRUN}}$ protocol to attempt to slow the CardBus clock by a factor of 16

7 PCI Firmware Loading Function Programming Model (Function 2)

PCI1620 is a multifunction PCI device. Function 2 is provided so that the firmware can be loaded into internal program memory. The configuration header is compliant with the PCI Specification as a standard header. Table 7−1 illustrates the PCI configuration header for function 2.

Table 7−1. Function 2 Configuration Register Map

REGISTER NAME				OFFSET
Device ID		Vendor ID		00h
Status		Command		04h
Class code			Revision ID	08h
BIST	Header type	Latency timer	Cache line size	0Ch
Base address register				10h
Reserved				14h−28h
Subsystem ID		Subsystem vendor ID		2Ch
Reserved				30h
Reserved			Capabilities pointer	34h
Reserved				38h
Max latency	Min grant	Interrupt pin	Interrupt line	3Ch
Reserved				40h
Power management capabilities		Next item pointer	Capability ID	44h
PM data (Reserved)	PMCSR_BSE	Power management CSR		48h
Reserved				4Ch−FCh

7.1 Vendor ID Register

This 16-bit read-only register contains the value 104Ch, which is the vendor ID assigned to Texas Instruments.

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
Type: Read-only
Default: 104Ch

7.2 Device ID Register

This 16-bit read-only register contains the value 8201h assigned by TI to the PCI1620 firmware loading function.

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

Register: **Device ID**
Offset: 02h
Type: Read-only
Default: 8201h

7.3 Command Register

This register provides control over the PCI1620 interface to the PCI bus. All bit functions adhere to the definitions in the *PCI Local Bus Specification*. See Table 7−2 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	R	RW	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 7−2. Command Register Description

BIT	SIGNAL	TYPE	FUNCTION
15−10	RSVD	R	Reserved. Returns zeros when read.
9	FBB_EN	R	Fast back-to-back enable. The PCI1620 does not generate fast back-to-back transactions; thus, this bit returns 0 when read.
8	SERR_EN	RW	$\overline{\text{SERR}}$ enable. When set, the PCI1620 $\overline{\text{SERR}}$ driver is enabled. $\overline{\text{SERR}}$ can be asserted after detecting an address parity error on the PCI bus.
7	STEP_EN	R	Address/data stepping control. The PCI1620 does not support address/data stepping, so this bit is hard-wired to 0.
6	PERR_EN	RW	Parity error enable. When set, the PCI1620 is enabled to drive $\overline{\text{PERR}}$ response to parity errors through the $\overline{\text{PERR}}$ signal.
5	VGA_EN	R	VGA palette snoop enable. The PCI1620 does not feature VGA palette snooping; thus, this bit returns 0 when read.
4	MWI_EN	RW	Memory write-and-invalidate (MWI) enable. When set, the PCI1620 is enabled to generate MWI PCI bus commands. If reset, the PCI1620 generates memory write commands instead.
3	SPECIAL	R	Special cycle enable. The PCI1620 does not respond to special cycle transactions. This bit returns 0 when read.
2	MAST_EN	RW	Bus master enable. When set, the PCI1620 is enabled to initiate cycles on the PCI bus.
1	MEM_EN	RW	Memory response enable. Setting this bit enables the PCI1620 to respond to memory cycles on the PCI bus.
0	IO_EN	RW	I/O space enable. Setting this bit enables the PCI1620 to respond to I/O space accesses.

7.4 Status Register

This register provides device information to the host system. Bits in this register may be read normally. A bit in the status register is reset when a 1 is written to that bit location; a 0 written to a bit location has no effect. All bit functions adhere to the definitions in the *PCI Local Bus Specification*. See Table 7−3 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	Status															
Type	RCU	RCU	RCU	RCU	RCU	R	R	RCU	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0

Register: **Status**
Offset: 06h
Type: Read-only, Read/Clear/Update
Default: 0210h

Table 7−3. Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15	PAR_ERR	RCU	Detected parity error. This bit is set when a parity error is detected, either an address- or data-parity error.
14	SYS_ERR	RCU	Signaled system error. This bit is set when $\overline{\text{SERR}}$ is enabled and the PCI1620 has signaled a system error to the host.
13	MABORT	RCU	Received master abort. This bit is set when a cycle initiated by the PCI1620 on the PCI bus has been terminated by a master abort.
12	TABORT_REC	RCU	Received target abort. This bit is set when a cycle initiated by the PCI1620 on the PCI bus has been terminated by a target abort.
11	TABORT_SIG	RCU	Signaled target abort. This bit is set by the PCI1620 when it terminates a transaction on the PCI bus with a target abort.
10−9	PCI_SPEED	R	DEVSEL timing. These bits encode the timing of $\overline{\text{DEVSEL}}$ and are hardwired 01b, indicating that the PCI1620 asserts this signal at a medium speed on nonconfiguration cycle accesses.
8	DATAPAR	RCU	Data parity error detected. This bit is set when the following conditions have been met: a. $\overline{\text{PERR}}$ was asserted by any PCI device, including the PCI1620. b. The PCI1620 was the bus master during the data parity error. c. The parity error response bit is set in the command register.
7	FBB_CAP	R	Fast back-to-back capable. The PCI1620 cannot accept fast back-to-back transactions; thus, this bit is hardwired to 0.
6	UDF	R	UDF supported. The PCI1620 does not support the user definable features; thus, this bit is hardwired to 0.
5	66MHZ	R	66-MHz capable. The PCI1620 operates at a maximum PCLK frequency of 33 MHZ; therefore, this bit is hardwired to 0.
4	CAPLIST	R	Capabilities list. This bit returns 1 when read, indicating that the firmware loading function of the PCI1620 supports additional PCI capabilities.
3−0	RSVD	R	Reserved. These bits return 0s when read.

7.5 Class Code and Revision ID Register

This read-only register categorizes the base class, subclass, and programming interface of the function. The base class is 08h, identifying the function as a generic system peripheral. The subclass is 80h, identifying the function as an other system peripheral. The programming interface is 00h. Furthermore, the TI chip revision (00h) is indicated in the lower byte. See Table 7−4 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	Class code and revision ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Class code and revision ID															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Register: **Class code and revision ID**
Offset: 08h
Type: Read-only
Default: 0880 0001h

Table 7−4. Class Code and Revision ID Register Description

BIT	SIGNAL	ACCESS	DESCRIPTION
31−24	BASECLASS	R	Base class. This field returns 08h when read, which broadly classifies the function as a generic system peripheral.
23−16	SUBCLASS	R	Subclass. This field returns 80h when read, which specifically classifies the function as other system peripheral.
15−8	PGMIF	R	Programming interface. This field returns 00h when read.
7−0	CHIPREV	R	Silicon revision. This field returns the silicon revision of PCI1620.

7.6 Cache Line Size Register

This read/write 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
Type: Read/Write
Default: 00h

7.7 Latency Timer Register

This read/write register specifies the latency timer for PCI1620, in units of PCI clock cycles.

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

7.8 Header Type Register

This read-only register indicates that this function has a standard PCI header type.

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

Register: **Header type**
Offset: 0Eh
Type: Read-only
Default: 00h

7.9 BIST Register

Because PCI1620 does not support a built-in self test (BIST), this read-only 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
Type: Read-only
Default: 00h

7.10 Base Address Register

This register specifies the base address of a 4-byte I/O space used for loading the PCI1620 firmware. See Table 7−5 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	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	Base address															
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	1

Register: **Base address**
Offset: 10h
Type: Read-only, Read/Write
Default: 0000 0001h

Table 7−5. Base Address Register Description

BIT	SIGNAL	TYPE	FUNCTION
31−2	BAR	RW	Base address. This field specifies the upper 30 bits of the 32-bit starting base address.
1	RSVD	R	Reserved. This bit returns 0 when read.
0	IO_INDICATOR	R	I/O space indicator. This bit is hardwired to 1 to indicate that the base address maps into I/O space.

7.11 Subsystem Vendor ID Register

This register, used for system and option card identification purposes, may be required for certain operating systems. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register (PCI offset 80h, see Section 4.31) of function 0. When bit 5 is 0, this register is read/write; when bit 5 is 1, this register is read-only. The default mode is read-only.

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

Register: **Subsystem vendor ID**
Offset: 2Ch
Type: Read-only (Read/Write when bit 5 of the system control register is 0)
Default: 0000h

7.12 Subsystem ID Register

This register, used for system and option card identification purposes, may be required for certain operating systems. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register (PCI offset 80h, see Section 4.31) of function 0. When bit 5 is 0, this register is read/write; when bit 5 is 1, this register is read-only. The default mode is read-only.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Subsystem 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: 2Eh
Type: Read-only (Read/Write when bit 5 of the system control register is 0)
Default: 0000h

7.13 Capabilities Pointer Register

This read-only register provides a pointer into the PCI configuration header where the PCI power management block resides. Because the PCI power management registers begin at 44h, this register is hardwired to 44h.

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

Register: **Capabilities pointer**
Offset: 34h
Type: Read-only
Default: 44h

7.14 Interrupt Line Register

This read/write register is programmed by the system and indicates to the software which interrupt line the firmware loading function of PCI1620 is connected to. The default value of this register is FFh, indicating that an interrupt line has not yet been assigned to the function.

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

7.15 Interrupt Pin Register

This register defaults to 00h.

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	0	0

Register: **Interrupt pin**
Offset: 3Dh
Type: Read-only
Default: 00h

7.16 Min Grant Register

This read/write register can be used by host BIOS to assign a latency timer register value to PCI1620.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Min grant							
Type	RW	RW	RW	RW	RW	RW	RW	RW
Default	0	0	0	0	0	1	1	1

Register: **Min grant**
Offset: 3Eh
Type: Read/Write
Default: 07h

7.17 Max Latency Register

This read/write register may be used by host BIOS to assign an arbitration priority level to PCI1620.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Max latency							
Type	RW	RW	RW	RW	RW	RW	RW	RW
Default	0	0	0	0	0	1	0	0

Register: **Max latency**
Offset: 3Fh
Type: Read/Write
Default: 04h

7.18 Capability ID Register

This read-only register identifies the linked list item as the register for PCI power management. The register returns 01h when read.

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: 44h
Type: Read-only
Default: 01h

7.19 Next-Item Pointer Register

The contents of this read-only register indicate the next item in the linked list of capabilities for PCI1620. Because PCI power management is the only entry in the capabilities list for the firmware loading function of PCI1620, this register returns 00h 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: 45h
Type: Read-only
Default: 00h

7.20 Power-Management Capabilities Register

This register indicates the capabilities of the firmware loading function of PCI1620 related to PCI power management. See Table 7−6 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	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	1	0

Register: **Power-management capabilities**
Offset: 46h
Type: Read-only
Default: 0002h

Table 7−6. Power-Management Capabilities Register Description

BIT	SIGNAL	TYPE	FUNCTION
15−11	PME_SUPPORT	R	$\overline{\text{PME}}$ support. This 5-bit field indicates the power states from which the PCI1620 can assert $\overline{\text{PME}}$. These five bits return a value of 00000b by default, indicating that the firmware loading function does not assert $\overline{\text{PME}}$.
10	D2_SUPPORT	R	This bit returns a 0 when read, indicating that the function does not support the D2 device power state.
9	D1_SUPPORT	R	This bit returns a 0 when read, indicating that the function does not support the D1 device power state.
8−6	AUX_CURRENT	R	3.3-Vaux auxiliary power requirements. Because this function does not support $\overline{\text{PME}}$ generation from $D3_{COLD}$, this field returns 000b when read.
5	DSI	R	Device specific initialization. This bit returns 0 when read, indicating that the PCI1620 does not require special initialization beyond the standard PCI configuration header before a generic class driver is able to use it.
4	RSVD	R	Reserved. Returns zero when read.
3	PME_CLK	R	$\overline{\text{PME}}$ clock. This bit returns 0 when read, because the firmware loading function of the PCI1620 does not support $\overline{\text{PME}}$ generation.
2−0	PM_VERSION	R	Power management version. This field returns 010b, indicating revision 1.1 compatibility.

7.21 Power-Management Control/Status Register

This register determines and changes the current power state of the firmware loading function of PCI1620. The contents of this register are not affected by the internally generated reset caused by the transition from the $D3_{hot}$ to D0 state. See Table 7−7 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	Power-management control/status															
Type	RC	R	R	R	R	R	R	R	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: 48h
Type: Read-only, Read/Clear, Read/Write
Default: 0000h

Table 7−7. Power-Management Control/Status Register Description

BIT	SIGNAL	ACCESS	DESCRIPTION
15	PME_STAT	RC	$\overline{PME}$ status. This bit defaults to 0 because the firmware loading function does not support $\overline{PME}$ generation from any state.
14−13	DATA_SCALE	R	Data scale. This 2-bit field returns 0s when read because the firmware loading function does not use the data register.
12−9	DATA_SEL	R	Data select. This 4-bit field returns 0s when read because the firmware loading function does not use the data register.
8	PME_EN	R	$\overline{PME}$ enable. This bit defaults to 0 because the firmware loading function does not support $\overline{PME}$ generation from any state.
7−2	RSVD	R	Reserved. Returns zeros when read.
1−0	PWR_STATE	RW	Power state. This 2-bit field is used both to determine the current power state of the function and to set the function into a new power state. This field is encoded as follows: 00 = D0 01 = D1 10 = D2 11 = $D3_{hot}$

7.22 Power-Management Bridge Support Extension Register

This read-only register is not applicable to the firmware loading function of PCI1620 and returns 00h when read.

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

Register: **Power-management bridge support extension**
Offset: 4Ah
Type: Read-only
Default: 00h

7.23 Power-Management Data Register

The read-only register is not applicable to the firmware loading function of PCI1620 and returns 00h when read.

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: 4Bh
Type: Read-only
Default: 00h

8 Electrical Characteristics

8.1 Absolute Maximum Ratings Over Operating Temperature Ranges †

Supply voltage range:	V_{CC}	−0.5 V to 4 V
	VR_OUT	−0.2 V to 2.2 V
Clamping voltage range, V_{CCP}, V_{CCA}, V_{CCB}		−0.5 V to 5.5 V
Input voltage range, V_I:	PCI	−0.5 V to V_{CCP} + 0.5 V
	Card A	−0.5 V to V_{CCA} + 0.5 V
	Card B	−0.5 V to V_{CCB} + 0.5 V
	Miscellaneous	−0.5 to V_{CC} + 0.5 V
Output voltage range, V_O:	PCI	−0.5 V to V_{CC} + 0.5 V
	Card A	−0.5 V to V_{CCA} + 0.5 V
	Card B	−0.5 V to V_{CCB} + 0.5 V
	Miscellaneous	−0.5 to V_{CC} + 0.5 V
Input clamp current, I_{IK} ($V_I < 0$ or $V_I > V_{CC}$) (see Note 1)		±20 mA
Output clamp current, I_{OK} ($V_O < 0$ or $V_O > V_{CC}$) (see Note 2)		±20 mA
Storage temperature range, T_{stg}		−65°C to 150°C
Virtual junction temperature, T_J		150°C

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

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

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

8.2 Recommended Operating Conditions (see Note 3)

			OPERATION	MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage	Commercial	3.3 V	3	3.3	3.6	V
V_{CCP}	PCI I/O voltage	Commercial	3.3 V	3	3.3	3.6	V
			5 V	4.5	5	5.25	
$V_{CC(A/B)}$	PC Card I/O voltage	Commercial	3.3 V	3	3.3	3.6	V
			5 V	4.75	5	5.25	
VR_OUT	Core voltage	Commercial	1.8 V	1.6	1.8	2	V
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	3.3 V	0		V_{CCP}	V
		PC Card	5 V	0		$V_{CC(A/B)}$	
		Miscellaneous‡		0		V_{CC}	
V_O§	Output voltage	PCI	3.3 V	0		V_{CC}	V
		PC Card	5 V	0		V_{CC}	
		Miscellaneous‡		0		V_{CC}	
t_t	Input transition times (t_r and t_f)	PCI and PC Card		1		4	ns
		Miscellaneous		0		6	
T_A	Operating ambient temperature range			0	25	70	°C
T_J¶	Virtual junction temperature			0	25	115	°C

† Applies to external inputs and bidirectional buffers without hysteresis
‡ Miscellaneous terminals are A_CVS1//A_VS1, A_CVS2//A_VS2, B_CVS1//B_VS1, B_CVS2//B_VS2, CLOCK, DATA, LATCH, SPKROUT, CLK48, A_CCD1//A_CD1, A_CCD2//A_CD2, GRST, B_CCD1//B_CD1, B_CCD2//B_CD2, RI_OUT (open drain), SUSPEND.
§ Applies to external output buffers
¶ These junction temperatures reflect simulation conditions. The customer is responsible for verifying junction temperature.
NOTE 3: Unused terminals (input or I/O) must be held high or low to prevent them from floating.

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

PARAMETER		TERMINALS	OPERATION	TEST CONDITIONS	MIN	MAX	UNIT
V_{OH}	High-level output voltage	PCI	3.3 V	I_{OH} = −0.5 mA	0.9 V_{CC}		V
			5 V	I_{OH} = −2 mA	2.4		
		PC Card	3.3 V	I_{OH} = −0.15 mA	0.9 V_{CC}		
			5 V	I_{OH} = −0.15 mA	2.4		
		Miscellaneous		I_{OH} = −4 mA	V_{CC}−0.6		
V_{OL}	Low-level output voltage	PCI	3.3 V	I_{OL} = 1.5 mA		0.1 V_{CC}	V
			5 V	I_{OL} = 6 mA		0.55	
		PC Card	3.3 V	I_{OL} = 0.7 mA		0.1 V_{CC}	
			5 V	I_{OL} = 0.7 mA		0.55	
		Miscellaneous		I_{OL} = 4 mA		0.5	
I_{OZL}	3-state output, high-impedance state current	Output	3.6 V	V_I = GND		−20	µA
			5.25 V	V_I = GND		−20	
I_{OZH}	3-state output, high-impedance state current	Output	3.6 V	$V_I = V_{CC}$†		20	µA
			5.25 V	$V_I = V_{CC}$†		20	
I_{IL}	Low-level input current	Input		V_I = GND		−20	µA
		I/O		V_I = GND		−20	
		Pullup		V_I = GND		−300	
I_{IH}	High-level input current	Input	3.6 V	$V_I = V_{CC}$‡		20	µA
			5.25 V	$V_I = V_{CC}$‡		20	
		I/O	3.6 V	$V_I = V_{CC}$‡		20	
			5.25 V	$V_I = V_{CC}$‡		20	
		Miscellaneous	3.6 V	$V_I = V_{CC}$‡		20	
		Pulldown	3.6 V	$V_I = V_{CC}$‡		300	

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

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

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

8.5 PCI Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature (see Note 5, Figure 8−1, and Figure 8−4)

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 5		11	ns
		PCLK-to-shared signal invalid delay time	t_{inv}		2		
t_{en}	Enable time, high impedance-to-active delay time from PCLK		t_{on}		2		ns
t_{dis}	Disable time, active-to-high impedance delay time from PCLK		t_{off}			28	ns
t_{su}	Setup time before PCLK valid		t_{su}		7		ns
t_h	Hold time after PCLK high		t_h		0		ns

NOTES: 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.

5. 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 = delay time, t_{su} = setup time, and t_h = hold time.

8.6 Switching Characteristics for PHY-Link Interface

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

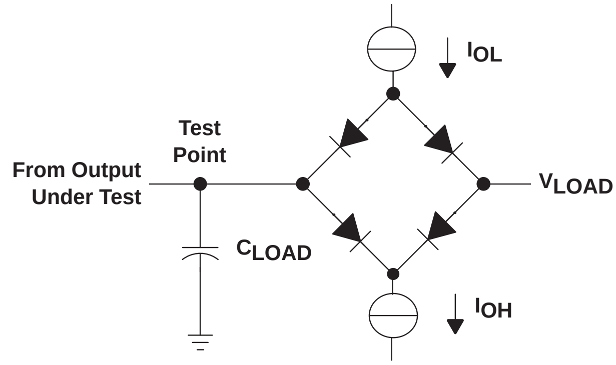
8.7 Parameter Measurement Information

LOAD CIRCUIT PARAMETERS

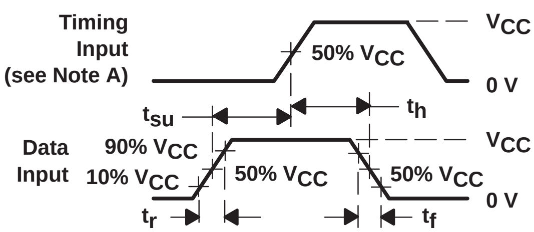
TIMING PARAMETER		C_{LOAD}† (pF)	I_{OL} (mA)	I_{OH} (mA)	V_{LOAD}‡ (V)
t_{en}	t_{PZH}	50	8	−8	0
	t_{PZL}				3
t_{dis}	t_{PHZ}	50	8	−8	1.5
	t_{PLZ}				
t_{pd}		50	8	−8	‡

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

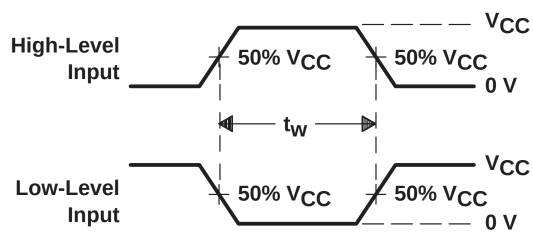
‡ $\frac{V_{LOAD} - V_{OL}}{I_{OL}} = 50\ \Omega$, where $V_{OL} = 0.6$ V, $I_{OL} = 8$ mA



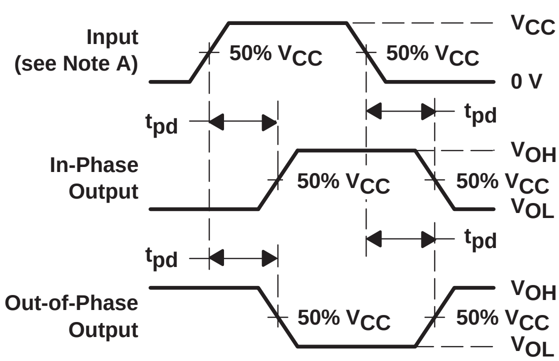
LOAD CIRCUIT



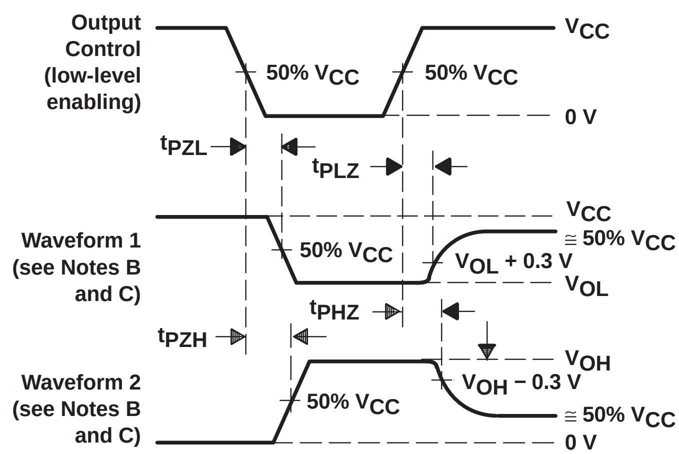
VOLTAGE WAVEFORMS
SETUP AND HOLD TIMES
INPUT RISE AND FALL TIMES



VOLTAGE WAVEFORMS
PULSE DURATION



VOLTAGE WAVEFORMS
PROPAGATION DELAY TIMES



VOLTAGE WAVEFORMS
ENABLE AND DISABLE TIMES, 3-STATE OUTPUTS

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

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

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

Figure 8−1. Load Circuit and Voltage Waveforms

8.8 PCI Bus Parameter Measurement Information

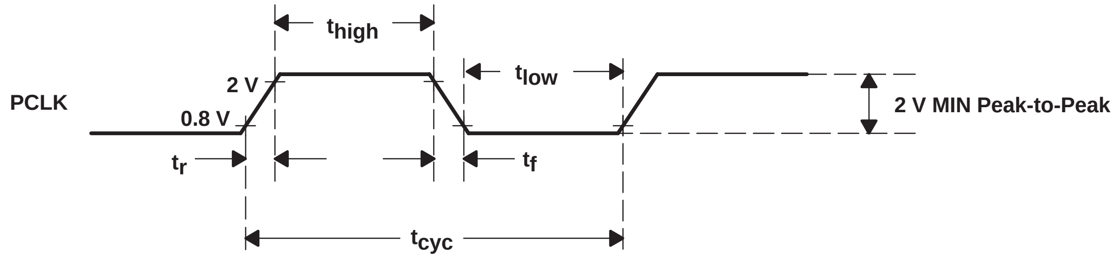


Figure 8−2. PCLK Timing Waveform

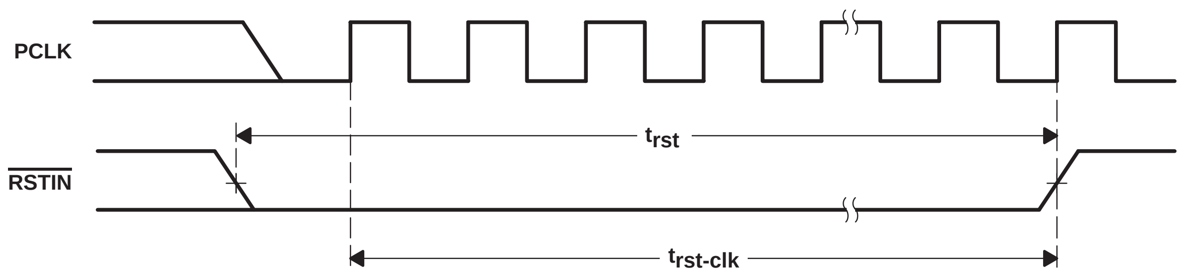


Figure 8−3. $\overline{\text{RSTIN}}$ Timing Waveforms

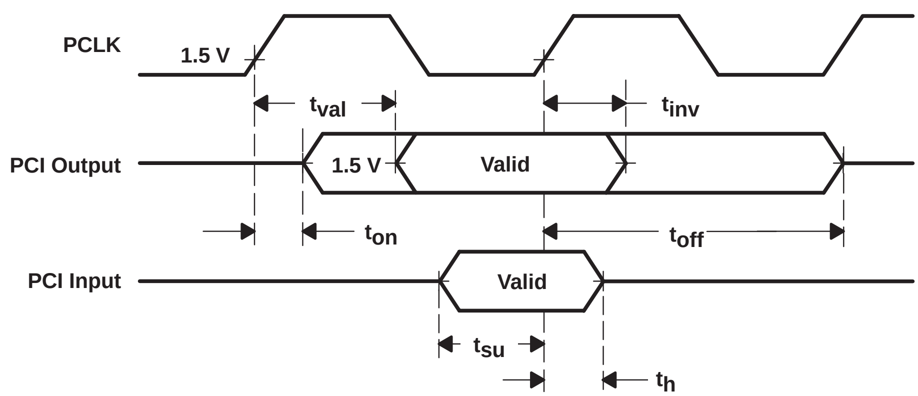


Figure 8−4. Shared Signals Timing Waveforms

8.9 PC Card Cycle Timing

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

The PC Card address setup and hold times are a function of the wait-state bits. Table 8−1 shows address setup time in PCLK cycles and nanoseconds for I/O and memory cycles. Table 8−2 and Table 8−3 show command active time in PCLK cycles and nanoseconds for I/O and memory cycles. Table 8−4 shows address hold time in PCLK cycles and nanoseconds for I/O and memory cycles.

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

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

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

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

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

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

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

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

8.10 Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature, Memory Cycles (for 100-ns Common Memory) (see Note 8 and Figure 8−5)

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

NOTE 6: These times are dependent on the register settings associated with ISA wait states and data size. They are also dependent on cycle type (read/write, memory/I/O) and $\overline{WAIT}$ from PC Card. The times listed here represent absolute minimums (the times that would be observed if programmed for zero wait state, 16-bit cycles) with a 33-MHz PCI clock.

8.11 Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature, I/O Cycles (see Figure 8−6)

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

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

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

8.13 PC Card Parameter Measurement Information

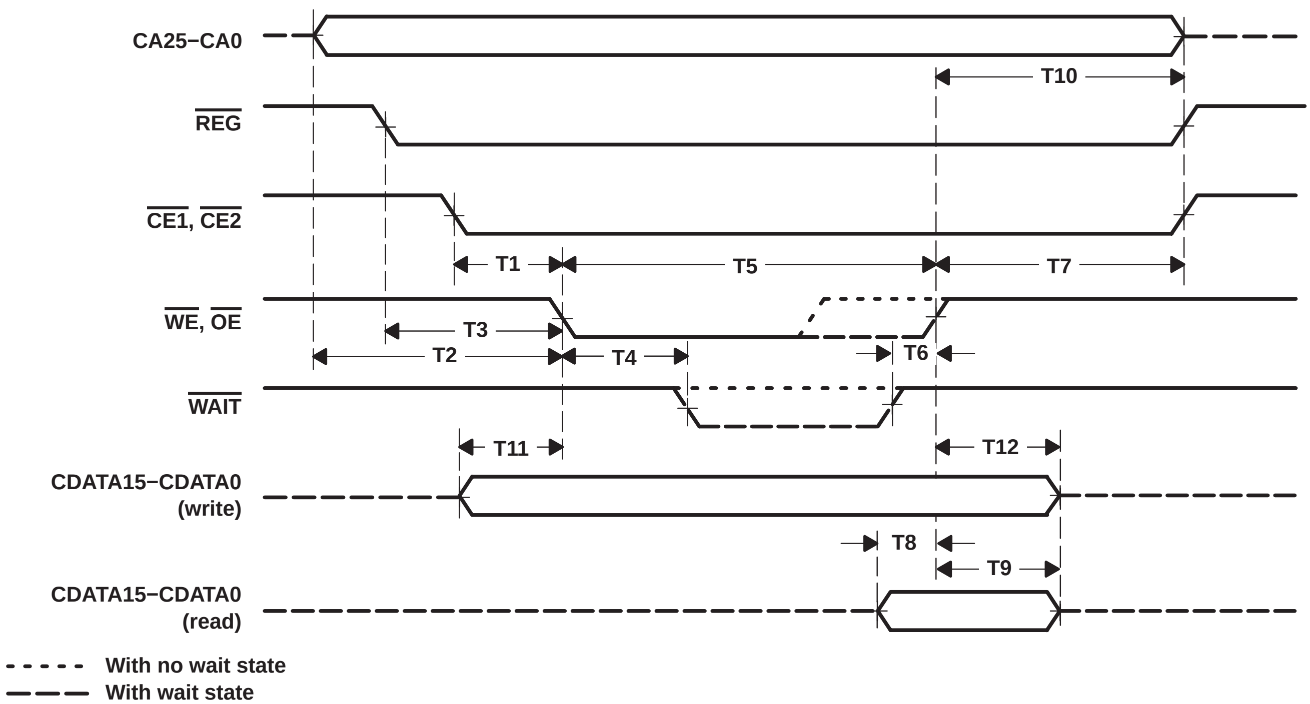


Figure 8−5. PC Card Memory Cycle

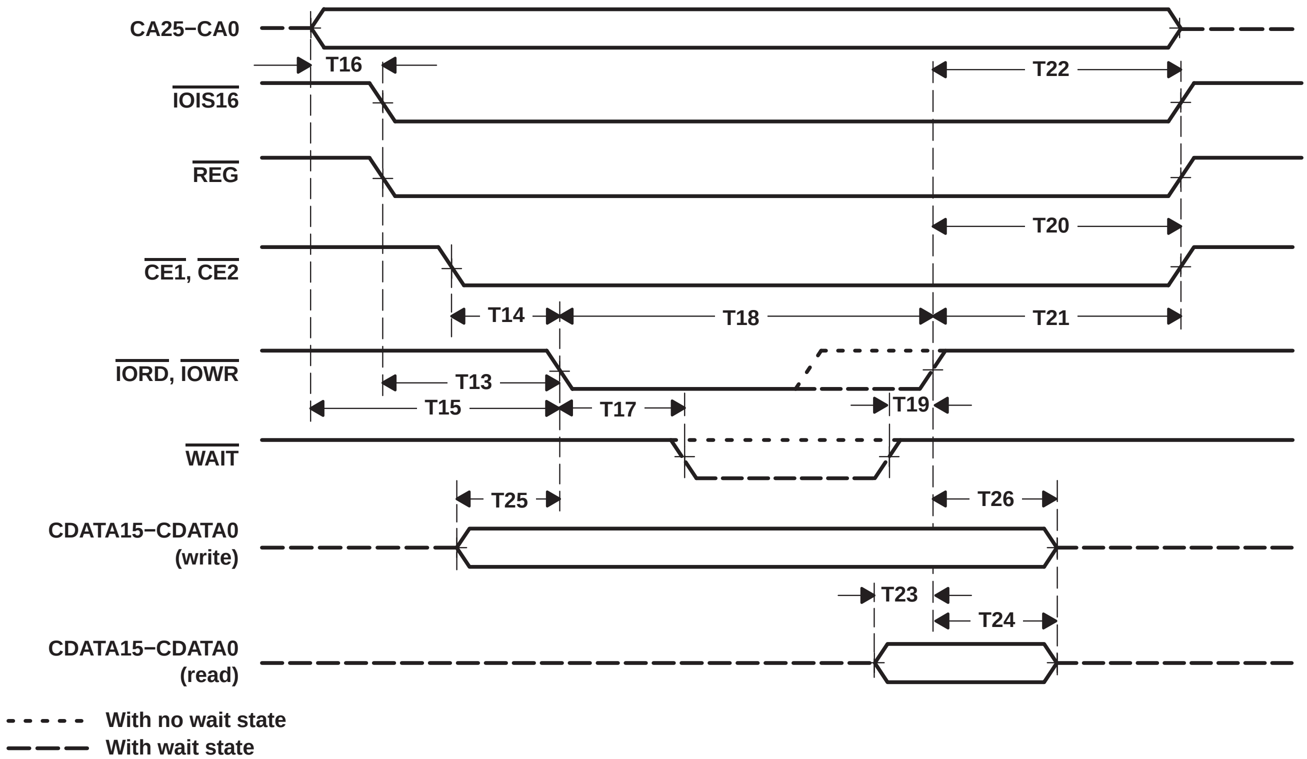


Figure 8−6. PC Card I/O Cycle

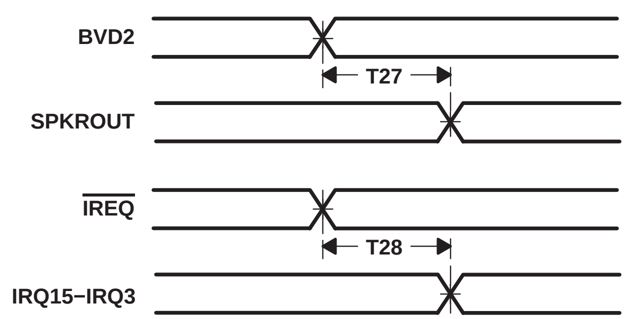
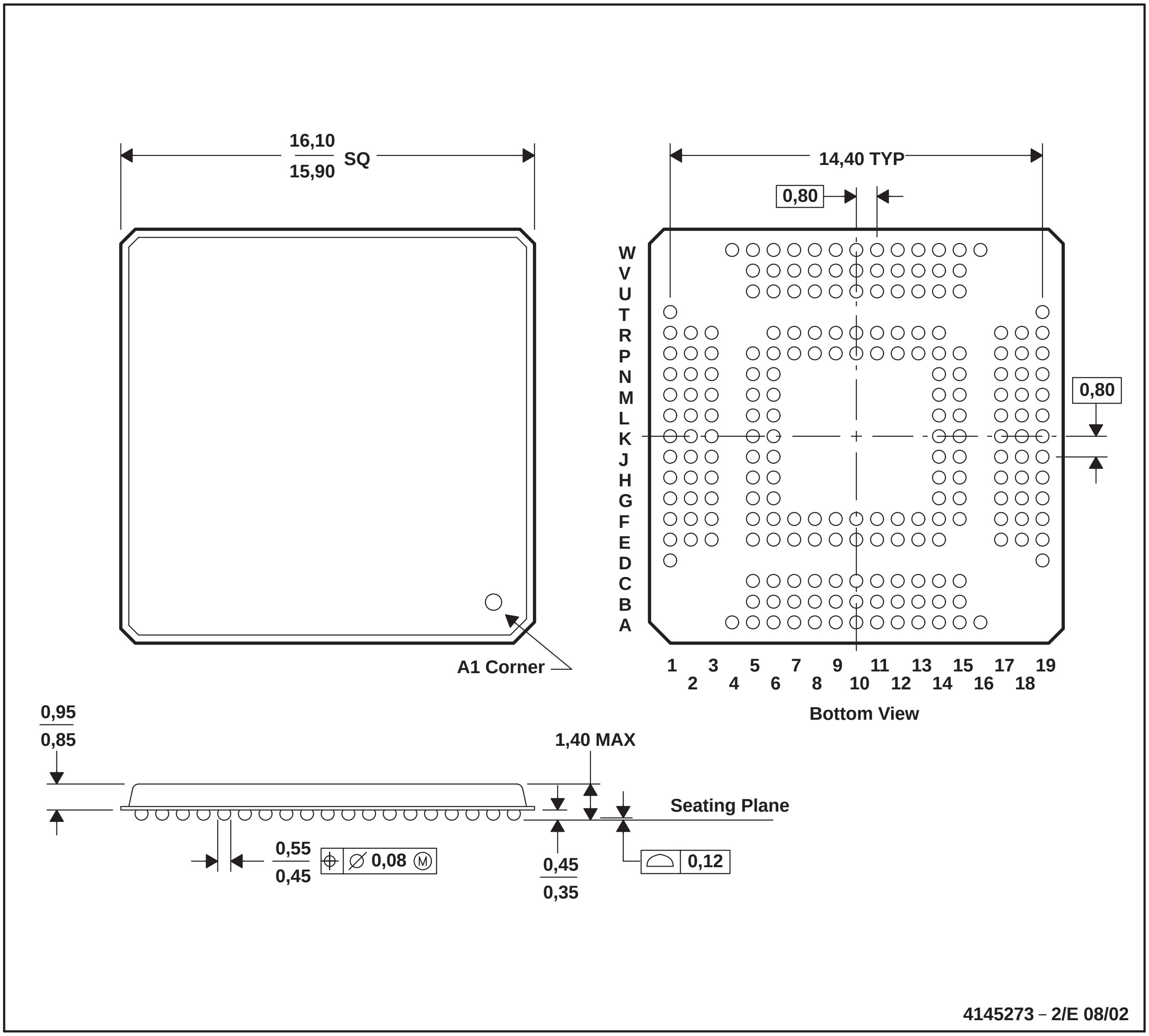


Figure 8−7. Miscellaneous PC Card Delay Times

9 Mechanical Data

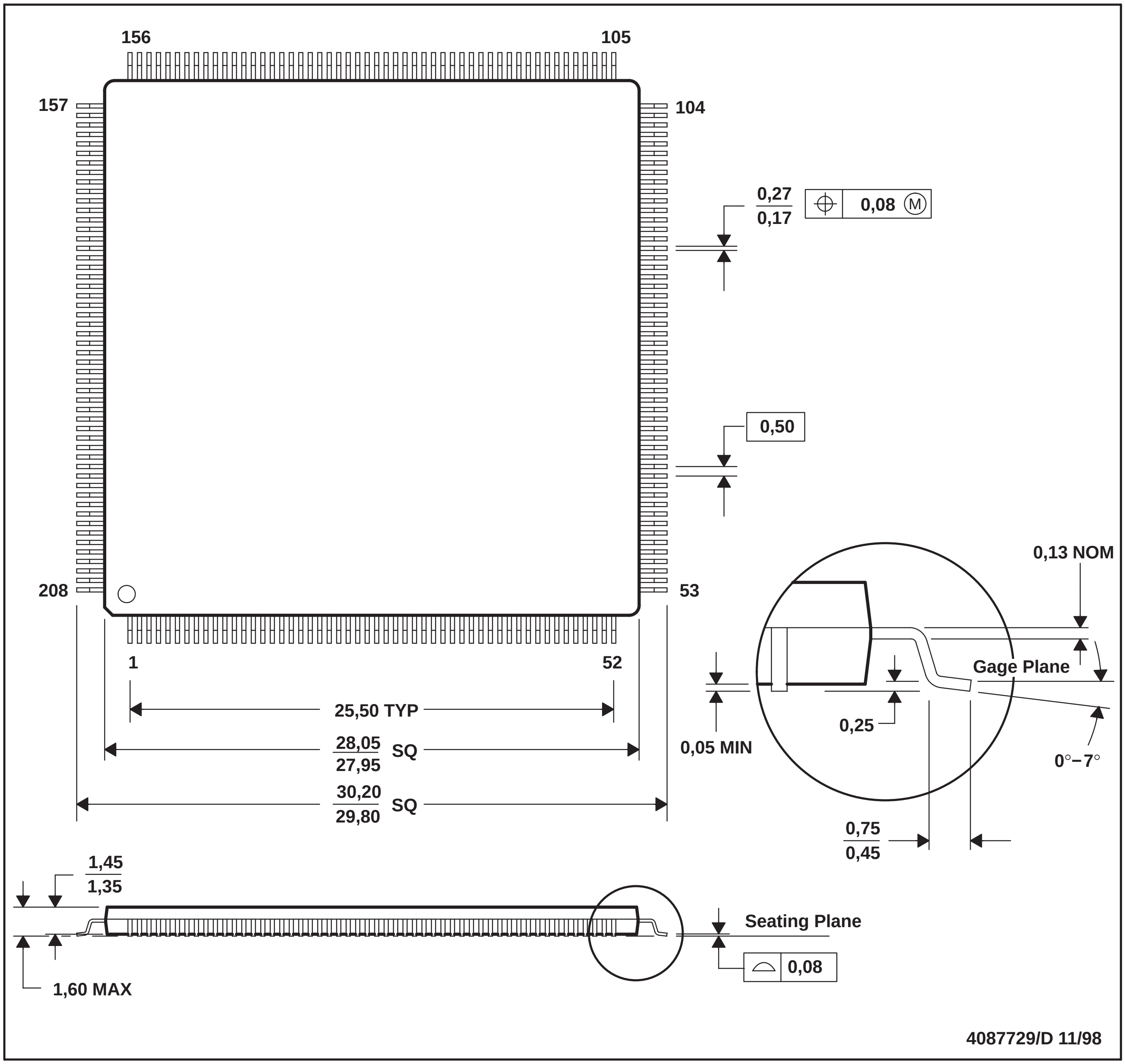
GHK (S−PBGA−N209) **PLASTIC BALL GRID ARRAY**



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.

PDV (S-PQFP-G208) PLASTIC QUAD FLATPACK



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-026