PCI1420
PC Card Controllers
Data Manual

Literature Number: SCPS047
April 1999

IMPORTANT NOTICE

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

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

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

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

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

Contents

List of Illustrations

List of Tables

1 Introduction

1.1 Description

The TI PCI1420, the industry's first 208-pin controller to meet the *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges*, is a high-performance PCI-to-CardBus controller that supports two independent card sockets compliant with the *1997 PC Card Standard*. The PCI1420 provides features that make it the best choice for bridging between PCI and PC Cards in both notebook and desktop computers. The *1997 PC Card Standard* retains the 16-bit PC Card specification defined in *PCI Local Bus Specification* and defines the new 32-bit PC Card, CardBus, capable of full 32-bit data transfers at 33 MHz. The PCI1420 supports any combination of 16-bit and CardBus PC Cards in the two sockets, powered at 5 V or 3.3 V, as required.

The PCI1420 is compliant with the *PCI Local Bus Specification*, and its PCI interface can act as either a PCI master device or a PCI slave device. The PCI bus mastering is initiated during 16-bit PC Card DMA transfers or CardBus PC Card bridging transactions. The PCI1420 is also compliant with the latest *PCI Bus Power Management Interface Specification*.

All card signals are internally buffered to allow hot insertion and removal without external buffering. The PCI1420 is register compatible with the Intel 82365SL-DF and 82365SL ExCA controllers. The PCI1420 internal data path logic allows the host to access 8-, 16-, and 32-bit cards using full 32-bit PCI cycles for maximum performance. Independent buffering and a pipeline architecture provide an unsurpassed performance level with sustained bursting. The PCI1420 can also be programmed to accept fast posted writes to improve system-bus utilization.

Multiple system-interrupt signaling options are provided, including: parallel PCI, parallel ISA, serialized ISA, and serialized PCI. Furthermore, general-purpose inputs and outputs are provided for the board designer to implement sideband functions. Many other features designed into the PCI1420, such as socket activity light-emitting diode (LED) outputs, are discussed in detail throughout the design specification.

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

1.2 Features

The PCI1420 supports the following features:

- Fully compatible with the Intel™ 430TX (Mobile Triton II) chipset
- A 208-Pin Low-Profile QFP (PDV) or MicroStar Ball Grid Array (GHK) package
- 3.3-V core logic with universal PCI interfaces compatible with 3.3-V and 5-V PCI signaling environments
- Mix-and-match 5-V/3.3-V 16-bit PC Cards and 3.3-V CardBus Cards
- Two PC Card or CardBus slots with hot insertion and removal
- Uses serial interface to TI™ TPS2206/2216 dual-slot PC Card power switch
- Burst transfers to maximize data throughput with CardBus Cards
- Parallel PCI interrupts, parallel ISA IRQ and parallel PCI interrupts, serial ISA IRQ with parallel PCI interrupts, and serial ISA IRQ and PCI interrupts
- Serial EEPROM interface for loading subsystem ID and subsystem vendor ID
- Pipelined architecture allows greater than 130M bps throughput from CardBus-to-PCI and from PCI-to-CardBus

- Up to five general-purpose I/Os
- Programmable output select for $\overline{\text{CLKRUN}}$
- Multifunction PCI device with separate configuration space for each socket
- Five PCI memory windows and two I/O windows available for each R2 socket
- Two I/O windows and two memory windows available to each CardBus socket
- Exchangeable Card Architecture (ExCA) compatible registers are mapped in memory and I/O space
- Intel 82365SL-DF and 82365SL register compatible
- Distributed DMA (DDMA) and PC/PCI DMA
- 16-Bit DMA on both PC Card sockets
- Ring indicate, $\overline{\text{SUSPEND}}$, PCI $\overline{\text{CLKRUN}}$, and CardBus $\overline{\text{CCLKRUN}}$
- Socket activity LED pins
- PCI Bus Lock ($\overline{\text{LOCK}}$)
- Advanced Submicron, Low-Power CMOS Technology
- Internal Ring Oscillator

1.3 Related Documents

- *Advanced Configuration and Power Interface (ACPI) Specification* (Revision 1.0)
- *PCI Bus Power Management Interface Specification* (Revision 1.1)
- *PCI Bus Power Management Interface Specification for PCI to CardBus Bridges* (Revision 0.6)
- *PCI to PCMCIA CardBus Bridge Register Description* (Yenta) (Revision 2.1)
- *PCI Local Bus Specification* (Revision 2.2)
- *PCI Mobile Design Guide* (Revision 1.0)
- *PCI14xx Implemenation Guide for D3 Wake-Up*
- *1997 PC Card™ Standard*
- *PC 99*
- *Serialized IRQ Support for PCI Systems* (Revision 6)

1.4 Ordering Information

ORDERING NUMBER	NAME	VOLTAGE	PACKAGE
PCI1420	PC Card Controller	3.3 V, 5-V Tolerant I/Os	208-pin LQFP 209-ball PBGA

2 Terminal Descriptions

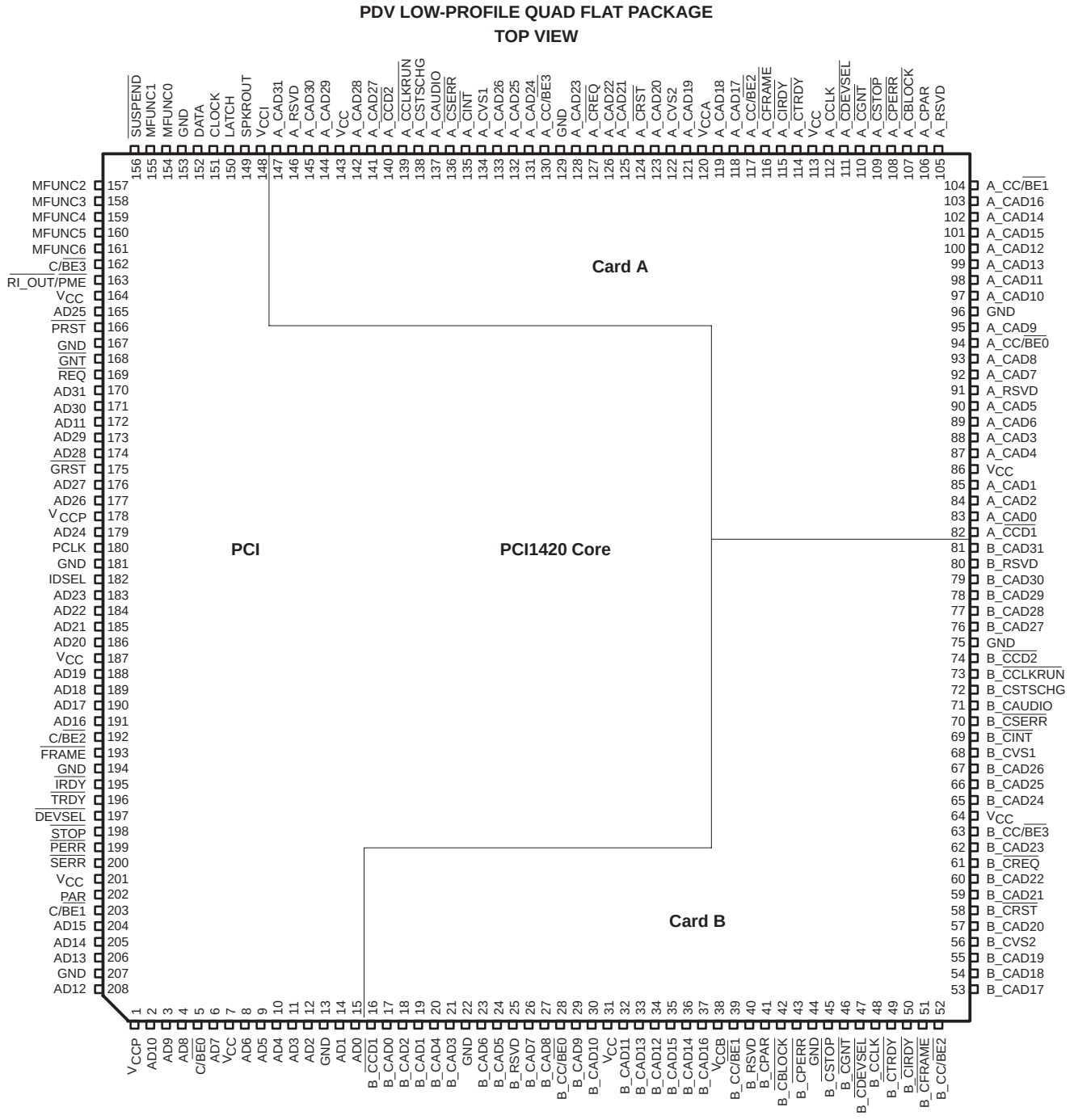


Figure 2–1. PCI-to-CardBus Pin Diagram

PDV LOW-PROFILE QUAD FLAT PACKAGE
TOP VIEW

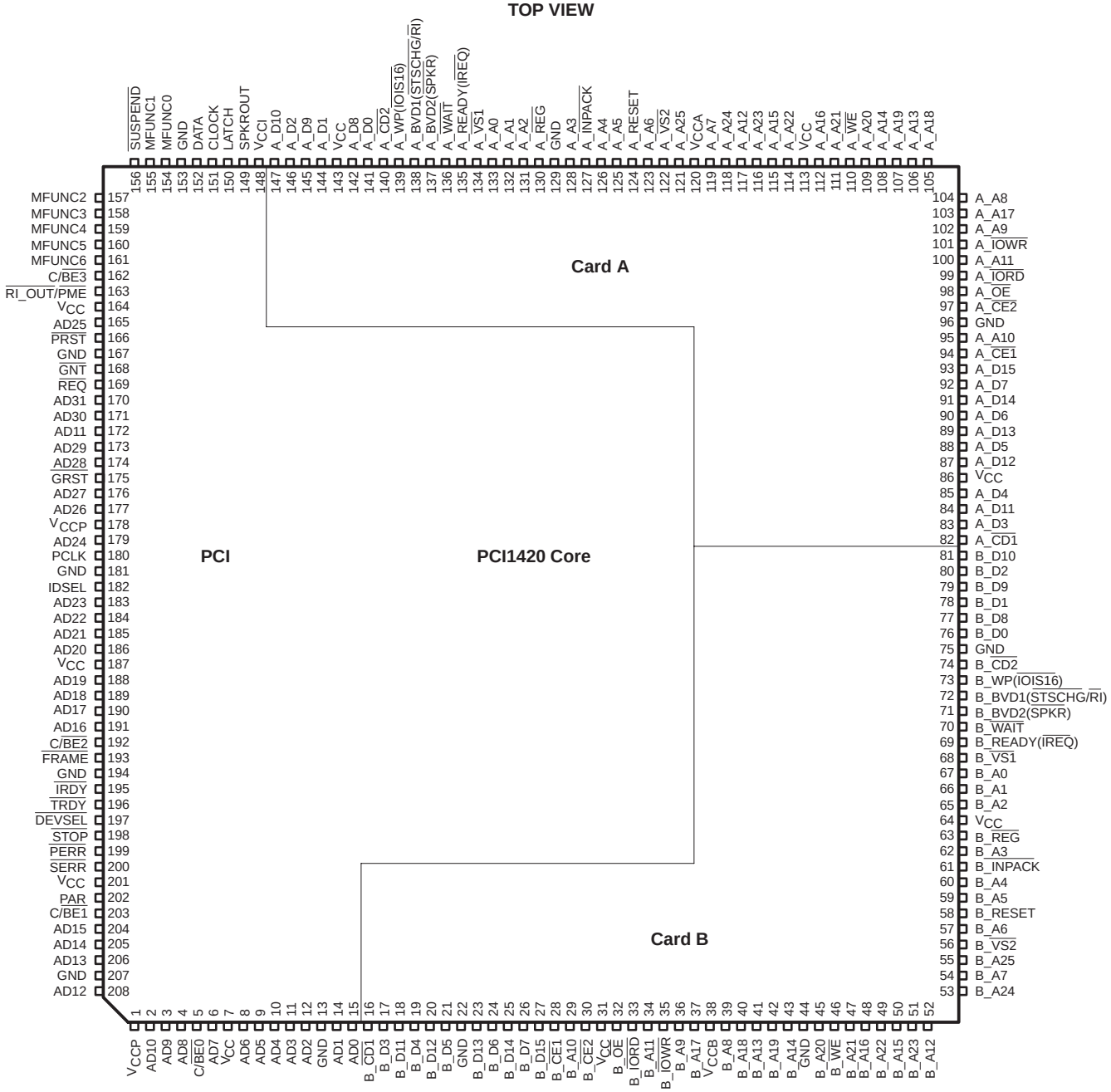


Figure 2–2. PCI-to-PC Card (16-Bit) Diagram

Table 2–1 and Table 2–2 show the terminal assignments for the CardBus PC Card; Table 2–3 and Table 2–4 show the terminal assignments for the 16-bit PC Card. Table 2–1 and Table 2–3 show the CardBus PC Card and the 16-bit PC Card terminals sorted alphanumerically by the associated GHK package terminal number. Table 2–2 and Table 2–4 show the CardBus PC Card and the 16-bit PC Card terminals sorted alphanumerically by the signal name and its associated terminal numbers. Pin E5 is a no connection identification ball.

Table 2–1. CardBus PC Card Signal Names by GHK/PDV Pin Number

PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME
GHK	PDV		GHK	PDV		GHK	PDV		GHK	PDV	
A4	208	AD12	E3	2	AD10	G19	143	V_{CC}	L18	124	A_$\overline{CRST}$
A5	203	C/$\overline{BE1}$	E6	206	AD13	H1	18	B_CAD2	L19	123	A_CAD20
A6	199	$\overline{PERR}$	E7	201	V_{CC}	H2	17	B_CAD0	M1	34	B_CAD12
A7	195	$\overline{IRDY}$	E8	194	GND	H3	16	B_$\overline{CCD1}$	M2	35	B_CAD15
A8	190	AD17	E9	189	AD18	H5	15	AD0	M3	36	B_CAD14
A9	185	AD21	E10	183	AD23	H6	11	AD3	M5	38	V_{CCB}
A10	180	PCLK	E11	178	V_{CCP}	H14	141	A_CAD27	M6	37	B_CAD16
A11	175	$\overline{GRST}$	E12	171	AD30	H15	142	A_CAD28	M14	115	A_$\overline{CIRDY}$
A12	174	AD28	E13	165	AD25	H17	140	A_$\overline{CCD2}$	M15	119	A_CAD18
A13	170	AD31	E14	159	MFUNC4	H18	139	A_$\overline{CCLKRUN}$	M17	120	V_{CCA}
A14	166	$\overline{PRST}$	E17	155	MFUNC1	H19	138	A_CSTSCHG	M18	121	A_CAD19
A15	162	C/$\overline{BE3}$	E18	153	GND	J1	19	B_CAD1	M19	122	A_CVS2
A16	157	MFUNC2	E19	151	CLOCK	J2	20	B_CAD4	N1	39	B_CC/$\overline{BE1}$
B5	205	AD14	F1	10	AD4	J3	21	B_CAD3	N2	40	B_RSVD
B6	200	$\overline{SERR}$	F2	8	AD6	J5	22	GND	N3	41	B_CPAR
B7	196	$\overline{TRDY}$	F3	7	V_{CC}	J6	23	B_CAD6	N5	45	B_$\overline{CSTOP}$
B8	191	AD16	F5	3	AD9	J14	136	A_$\overline{CSERR}$	N6	42	B_$\overline{CBLOCK}$
B9	186	AD20	F6	204	AD15	J15	137	A_CAUDIO	N14	108	A_$\overline{CPERR}$
B10	181	GND	F7	198	$\overline{STOP}$	J17	135	A_$\overline{CINT}$	N15	113	V_{CC}
B11	176	AD27	F8	193	$\overline{FRAME}$	J18	134	A_CVS1	N17	116	A_$\overline{CFRAME}$
B12	173	AD29	F9	188	AD19	J19	133	A_CAD26	N18	117	A_CC/$\overline{BE2}$
B13	169	$\overline{REQ}$	F10	184	AD22	K1	24	B_CAD5	N19	118	A_CAD17
B14	164	V_{CC}	F11	179	AD24	K2	25	B_RSVD	P1	43	B_$\overline{CPERR}$
B15	161	MFUNC6/$\overline{CLKRUN}$	F12	167	GND	K3	26	B_CAD7	P2	44	GND
C5	207	GND	F13	160	MFUNC5	K5	27	B_CAD8	P3	46	B_$\overline{CGNT}$
C6	202	PAR	F14	152	DATA	K6	28	B_CC/$\overline{BE0}$	P5	50	B_$\overline{CIRDY}$
C7	197	$\overline{DEVSEL}$	F15	154	MFUNC0	K14	132	A_CAD25	P6	48	B_CCLK
C8	192	C/$\overline{BE2}$	F17	150	LATCH	K15	131	A_CAD24	P7	56	B_CVS2
C9	187	V_{CC}	F18	148	V_{CCI}	K17	130	A_CC/$\overline{BE3}$	P8	63	B_CC/$\overline{BE3}$
C10	182	IDSEL	F19	147	A_CAD31	K18	129	GND	P9	75	GND
C11	177	AD26	G1	14	AD1	K19	128	A_CAD23	P10	80	B_RSVD
C12	172	AD11	G2	13	GND	L1	29	B_CAD9	P11	84	A_CAD2
C13	168	$\overline{GNT}$	G3	12	AD2	L2	30	B_CAD10	P12	89	A_CAD6
C14	163	$\overline{RI\_OUT}$/$\overline{PME}$	G5	9	AD5	L3	31	V_{CC}	P13	94	A_CC/$\overline{BE0}$
C15	158	MFUNC3	G6	4	AD8	L5	33	B_CAD13	P14	100	A_CAD12
D1	1	V_{CCP}	G14	146	A_RSVD	L6	32	B_CAD11	P15	107	A_$\overline{CBLOCK}$
D19	156	$\overline{SUSPEND}$	G15	149	SPKROUT	L14	127	A_$\overline{CREQ}$	P17	111	A_$\overline{CDEVSEL}$
E1	6	AD7	G17	145	A_CAD30	L15	126	A_CAD22	P18	112	A_CCLK
E2	5	C/$\overline{BE0}$	G18	144	A_CAD29	L17	125	A_CAD21	P19	114	A_$\overline{CTRDY}$

Table 2–1. CardBus PC Card Signal Names by GHK/PDV Pin Number (Continued)

PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME
GHK	PDV		GHK	PDV		GHK	PDV		GHK	PDV	
R1	47	B_$\overline{\text{CDEVSEL}}$	R18	109	A_$\overline{\text{CSTOP}}$	U14	98	A_CAD11	W4	53	B_CAD17
R2	49	B_$\overline{\text{CTRDY}}$	R19	110	A_$\overline{\text{CGNT}}$	U15	103	A_CAD16	W5	58	B_$\overline{\text{CRST}}$
R3	51	B_$\overline{\text{CFRAME}}$	T1	52	B_CC/$\overline{\text{BE2}}$	V5	57	B_CAD20	W6	62	B_CAD23
R6	55	B_CAD19	T19	105	A_RSVD	V6	60	B_CAD22	W7	66	B_CAD25
R7	61	B_$\overline{\text{CREQ}}$	U5	54	B_CAD18	V7	65	B_CAD24	W8	70	B_$\overline{\text{CSERR}}$
R8	67	B_CAD26	U6	59	B_CAD21	V8	69	B_$\overline{\text{CINT}}$	W9	71	B_CAUDIO
R9	74	B_$\overline{\text{CCD2}}$	U7	64	V_{CC}	V9	72	B_CSTSCHG	W10	76	B_CAD27
R10	79	B_CAD30	U8	68	B_CVS1	V10	77	B_CAD28	W11	81	B_CAD31
R11	85	A_CAD1	U9	73	B_$\overline{\text{CCLKRUN}}$	V11	82	A_$\overline{\text{CCD1}}$	W12	86	V_{CC}
R12	90	A_CAD5	U10	78	B_CAD29	V12	87	A_CAD4	W13	91	A_RSVD
R13	97	A_CAD10	U11	83	A_CAD0	V13	92	A_CAD7	W14	95	A_CAD9
R14	102	A_CAD14	U12	88	A_CAD3	V14	96	GND	W15	99	A_CAD13
R17	106	A_CPAR	U13	93	A_CAD8	V15	101	A_CAD15	W16	104	A_CC/$\overline{\text{BE1}}$

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

SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.	
	GHK	PDV		GHK	PDV		GHK	PDV		GHK	PDV
A_CAD0	U11	83	A_CAD27	H14	141	A_$\overline{\text{CTRDY}}$	P19	114	AD21	A9	185
A_CAD1	R11	85	A_CAD28	H15	142	A_CVS1	J18	134	AD22	F10	184
A_CAD2	P11	84	A_CAD29	G18	144	A_CVS2	M19	122	AD23	E10	183
A_CAD3	U12	88	A_CAD30	G17	145	A_RSVD	G14	146	AD24	F11	179
A_CAD4	V12	87	A_CAD31	F19	147	A_RSVD	T19	105	AD25	E13	165
A_CAD5	R12	90	A_CAUDIO	J15	137	A_RSVD	W13	91	AD26	C11	177
A_CAD6	P12	89	A_$\overline{\text{CBLOCK}}$	P15	107	AD0	H5	15	AD27	B11	176
A_CAD7	V13	92	A_CC/$\overline{\text{BE0}}$	P13	94	AD1	G1	14	AD28	A12	174
A_CAD8	U13	93	A_CC/$\overline{\text{BE1}}$	W16	104	AD2	G3	12	AD29	B12	173
A_CAD9	W14	95	A_CC/$\overline{\text{BE2}}$	N18	117	AD3	H6	11	AD30	E12	171
A_CAD10	R13	97	A_CC/$\overline{\text{BE3}}$	K17	130	AD4	F1	10	AD31	A13	170
A_CAD11	U14	98	A_$\overline{\text{CCD1}}$	V11	82	AD5	G5	9	B_CAD0	H2	17
A_CAD12	P14	100	A_$\overline{\text{CCD2}}$	H17	140	AD6	F2	8	B_CAD1	J1	19
A_CAD13	W15	99	A_CCLK	P18	112	AD7	E1	6	B_CAD2	H1	18
A_CAD14	R14	102	A_$\overline{\text{CCLKRUN}}$	H18	139	AD8	G6	4	B_CAD3	J3	21
A_CAD15	V15	101	A_$\overline{\text{CDEVSEL}}$	P17	111	AD9	F5	3	B_CAD4	J2	20
A_CAD16	U15	103	A_$\overline{\text{CFRAME}}$	N17	116	AD10	E3	2	B_CAD5	K1	24
A_CAD17	N19	118	A_$\overline{\text{CGNT}}$	R19	110	AD11	C12	172	B_CAD6	J6	23
A_CAD18	M15	119	A_$\overline{\text{CINT}}$	J17	135	AD12	A4	208	B_CAD7	K3	26
A_CAD19	M18	121	A_$\overline{\text{CIRDY}}$	M14	115	AD13	E6	206	B_CAD8	K5	27
A_CAD20	L19	123	A_CPAR	R17	106	AD14	B5	205	B_CAD9	L1	29
A_CAD21	L17	125	A_$\overline{\text{CPERR}}$	N14	108	AD15	F6	204	B_CAD10	L2	30
A_CAD22	L15	126	A_$\overline{\text{CREQ}}$	L14	127	AD16	B8	191	B_CAD11	L6	32
A_CAD23	K19	128	A_$\overline{\text{CRST}}$	L18	124	AD17	A8	190	B_CAD12	M1	34
A_CAD24	K15	131	A_$\overline{\text{CSERR}}$	J14	136	AD18	E9	189	B_CAD13	L5	33
A_CAD25	K14	132	A_$\overline{\text{CSTOP}}$	R18	109	AD19	F9	188	B_CAD14	M3	36
A_CAD26	J19	133	A_CSTSCHG	H19	138	AD20	B9	186	B_CAD15	M2	35

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

SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.	
	GHK	PDV		GHK	PDV		GHK	PDV		GHK	PDV
B_CAD16	M6	37	B_$\overline{\text{CCLKRUN}}$	U9	73	$\overline{\text{DEVSEL}}$	C7	197	PAR	C6	202
B_CAD17	W4	53	B_$\overline{\text{CDEVSEL}}$	R1	47	$\overline{\text{FRAME}}$	F8	193	PCLK	A10	180
B_CAD18	U5	54	B_$\overline{\text{CFRAME}}$	R3	51	GND	B10	181	$\overline{\text{PERR}}$	A6	199
B_CAD19	R6	55	B_$\overline{\text{CGNT}}$	P3	46	GND	C5	207	$\overline{\text{PRST}}$	A14	166
B_CAD20	V5	57	B_$\overline{\text{CINT}}$	V8	69	GND	E8	194	$\overline{\text{REQ}}$	B13	169
B_CAD21	U6	59	B_$\overline{\text{CIRDY}}$	P5	50	GND	E18	153	$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	C14	163
B_CAD22	V6	60	B_CPAR	N3	41	GND	F12	167	$\overline{\text{SERR}}$	B6	200
B_CAD23	W6	62	B_$\overline{\text{CPERR}}$	P1	43	GND	G2	13	SPKROUT	G15	149
B_CAD24	V7	65	B_$\overline{\text{CREQ}}$	R7	61	GND	J5	22	$\overline{\text{STOP}}$	F7	198
B_CAD25	W7	66	B_$\overline{\text{CRST}}$	W5	58	GND	K18	129	$\overline{\text{SUSPEND}}$	D19	156
B_CAD26	R8	67	B_$\overline{\text{CSERR}}$	W8	70	GND	P2	44	$\overline{\text{TRDY}}$	B7	196
B_CAD27	W10	76	B_$\overline{\text{CSTOP}}$	N5	45	GND	P9	75	V_{CC}	B14	164
B_CAD28	V10	77	B_CSTSCHG	V9	72	GND	V14	96	V_{CC}	C9	187
B_CAD29	U10	78	B_$\overline{\text{CTRDY}}$	R2	49	$\overline{\text{GNT}}$	C13	168	V_{CC}	E7	201
B_CAD30	R10	79	B_CVS1	U8	68	$\overline{\text{GRST}}$	A11	175	V_{CC}	F3	7
B_CAD31	W11	81	B_CVS2	P7	56	IDSEL	C10	182	V_{CC}	G19	143
B_CAUDIO	W9	71	B_RSVD	K2	25	$\overline{\text{IRDY}}$	A7	195	V_{CC}	L3	31
B_$\overline{\text{CBLOCK}}$	N6	42	B_RSVD	N2	40	LATCH	F17	150	V_{CC}	N15	113
B_CC/$\overline{\text{BE0}}$	K6	28	B_RSVD	P10	80	MFUNC0	F15	154	V_{CC}	U7	64
B_CC/$\overline{\text{BE1}}$	N1	39	C/$\overline{\text{BE0}}$	E2	5	MFUNC1	E17	155	V_{CC}	W12	86
B_CC/$\overline{\text{BE2}}$	T1	52	C/$\overline{\text{BE1}}$	A5	203	MFUNC2	A16	157	V_{CCA}	M17	120
B_CC/$\overline{\text{BE3}}$	P8	63	C/$\overline{\text{BE2}}$	C8	192	MFUNC3	C15	158	V_{CCB}	M5	38
B_$\overline{\text{CCD1}}$	H3	16	C/$\overline{\text{BE3}}$	A15	162	MFUNC4	E14	159	V_{CCI}	F18	148
B_$\overline{\text{CCD2}}$	R9	74	CLOCK	E19	151	MFUNC5	F13	160	V_{CCP}	D1	1
B_CCLK	P6	48	DATA	F14	152	MFUNC6/$\overline{\text{CLKRUN}}$	B15	161	V_{CCP}	E11	178

Table 2–3. 16-Bit PC Card Signal Names by GHK/PDV Pin Number

PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME
GHK	PDV		GHK	PDV		GHK	PDV		GHK	PDV	
A4	208	AD12	B7	196	$\overline{\text{TRDY}}$	C11	177	AD26	E11	178	V_{CCP}
A5	203	C/$\overline{\text{BE1}}$	B8	191	AD16	C12	172	AD11	E12	171	AD30
A6	199	$\overline{\text{PERR}}$	B9	186	AD20	C13	168	$\overline{\text{GNT}}$	E13	165	AD25
A7	195	$\overline{\text{IRDY}}$	B10	181	GND	C14	163	$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	E14	159	MFUNC4
A8	190	AD17	B11	176	AD27	C15	158	MFUNC3	E17	155	MFUNC1
A9	185	AD21	B12	173	AD29	D1	1	V_{CCP}	E18	153	GND
A10	180	PCLK	B13	169	$\overline{\text{REQ}}$	D19	156	$\overline{\text{SUSPEND}}$	E19	151	CLOCK
A11	175	$\overline{\text{GRST}}$	B14	164	V_{CC}	E1	6	AD7	F1	10	AD4
A12	174	AD28	B15	161	MFUNC6	E2	5	C/$\overline{\text{BE0}}$	F2	8	AD6
A13	170	AD31	C5	207	GND	E3	2	AD10	F3	7	V_{CC}
A14	166	$\overline{\text{PRST}}$	C6	202	PAR	E6	206	AD13	F5	3	AD9
A15	162	C/$\overline{\text{BE3}}$	C7	197	$\overline{\text{DEVSEL}}$	E7	201	V_{CC}	F6	204	AD15
A16	157	MFUNC2	C8	192	C/$\overline{\text{BE2}}$	E8	194	GND	F7	198	$\overline{\text{STOP}}$
B5	205	AD14	C9	187	V_{CC}	E9	189	AD18	F8	193	$\overline{\text{FRAME}}$
B6	200	$\overline{\text{SERR}}$	C10	182	IDSEL	E10	183	AD23	F9	188	AD19

Table 2–3. 16-Bit PC Card Signal Names by GHK/PDV Pin Number (Continued)

PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME
GHK	PDV		GHK	PDV		GHK	PDV		GHK	PDV	
F10	184	AD22	J18	134	A_$\overline{VS1}$	N14	108	A_A14	T1	52	B_A12
F11	179	AD24	J19	133	A_A0	N15	113	V_{CC}	T19	105	A_A18
F12	167	GND	K1	24	B_D6	N17	116	A_A23	U5	54	B_A7
F13	160	MFUNC5	K2	25	B_D14	N18	117	A_A12	U6	59	B_A5
F14	152	DATA	K3	26	B_D7	N19	118	A_A24	U7	64	V_{CC}
F15	154	MFUNC0	K5	27	B_D15	P1	43	B_A14	U8	68	B_$\overline{VS1}$
F17	150	LATCH	K6	28	B_$\overline{CE1}$	P2	44	GND	U9	73	B_WP($\overline{IOIS16}$)
F18	148	V_{CCI}	K14	132	A_A1	P3	46	B_$\overline{WE}$	U10	78	B_D1
F19	147	A_D10	K15	131	A_A2	P5	50	B_A15	U11	83	A_D3
G1	14	AD1	K17	130	A_$\overline{REG}$	P6	48	B_A16	U12	88	A_D5
G2	13	GND	K18	129	GND	P7	56	B_$\overline{VS2}$	U13	93	A_D15
G3	12	AD2	K19	128	A_A3	P8	63	B_$\overline{REG}$	U14	98	A_$\overline{OE}$
G5	9	AD5	L1	29	B_A10	P9	75	GND	U15	103	A_A17
G6	4	AD8	L2	30	B_$\overline{CE2}$	P10	80	B_D2	V5	57	B_A6
G14	146	A_D2	L3	31	V_{CC}	P11	84	A_D11	V6	60	B_A4
G15	149	SPKROUT	L5	33	B_$\overline{IORD}$	P12	89	A_D13	V7	65	B_A2
G17	145	A_D9	L6	32	B_$\overline{OE}$	P13	94	A_$\overline{CE1}$	V8	69	B_READY($\overline{IREQ}$)
G18	144	A_D1	L14	127	A_$\overline{INPACK}$	P14	100	A_A11	V9	72	B_BVD1($\overline{STSCHG}$/$\overline{R1}$)
G19	143	V_{CC}	L15	126	A_A4	P15	107	A_A19	V10	77	B_D8
H1	18	B_D11	L17	125	A_A5	P17	111	A_A21	V11	82	A_$\overline{CD1}$
H2	17	B_D3	L18	124	A_RESET	P18	112	A_A16	V12	87	A_D12
H3	16	B_$\overline{CD1}$	L19	123	A_A6	P19	114	A_A22	V13	92	A_D7
H5	15	AD0	M1	34	B_A11	R1	47	B_A21	V14	96	GND
H6	11	AD3	M2	35	B_$\overline{IOWR}$	R2	49	B_A22	V15	101	A_$\overline{IOWR}$
H14	141	A_D0	M3	36	B_A9	R3	51	B_A23	W4	53	B_A24
H15	142	A_D8	M5	38	V_{CCB}	R6	55	B_A25	W5	58	B_RESET
H17	140	A_$\overline{CD2}$	M6	37	B_A17	R7	61	B_$\overline{INPACK}$	W6	62	B_A3
H18	139	A_WP($\overline{IOIS16}$)	M14	115	A_A15	R8	67	B_A0	W7	66	B_A1
H19	138	A_BVD1($\overline{STSCHG}$/$\overline{R1}$)	M15	119	A_A7	R9	74	B_$\overline{CD2}$	W8	70	B_$\overline{WAIT}$
J1	19	B_D4	M17	120	V_{CCA}	R10	79	B_D9	W9	71	B_BVD2($\overline{SPKR}$)
J2	20	B_D12	M18	121	A_A25	R11	85	A_D4	W10	76	B_D0
J3	21	B_D5	M19	122	A_$\overline{VS2}$	R12	90	A_D6	W11	81	B_D10
J5	22	GND	N1	39	B_A8	R13	97	A_$\overline{CE2}$	W12	86	V_{CC}
J6	23	B_D13	N2	40	B_A18	R14	102	A_A9	W13	91	A_D14
J14	136	A_$\overline{WAIT}$	N3	41	B_A13	R17	106	A_A13	W14	95	A_A10
J15	137	A_BVD2($\overline{SPKR}$)	N5	45	B_A20	R18	109	A_A20	W15	99	A_$\overline{IORD}$
J17	135	A_READY($\overline{IREQ}$)	N6	42	B_A19	R19	110	A_$\overline{WE}$	W16	104	A_A8

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

SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.	
	GHK	PDV		GHK	PDV		GHK	PDV		GHK	PDV
A_A0	J19	133	A_D11	P11	84	AD26	C11	177	B_D5	J3	21
A_A1	K14	132	A_D12	V12	87	AD27	B11	176	B_D6	K1	24
A_A2	K15	131	A_D13	P12	89	AD28	A12	174	B_D7	K3	26
A_A3	K19	128	A_D14	W13	91	AD29	B12	173	B_D8	V10	77
A_A4	L15	126	A_D15	U13	93	AD30	E12	171	B_D9	R10	79
A_A5	L17	125	A_$\overline{\text{INPACK}}$	L14	127	AD31	A13	170	B_D10	W11	81
A_A6	L19	123	A_$\overline{\text{IORD}}$	W15	99	B_A0	R8	67	B_D11	H1	18
A_A7	M15	119	A_$\overline{\text{IOWR}}$	V15	101	B_A1	W7	66	B_D12	J2	20
A_A8	W16	104	A_$\overline{\text{OE}}$	U14	98	B_A2	V7	65	B_D13	J6	23
A_A9	R14	102	A_READY($\overline{\text{IREQ}}$)	J17	135	B_A3	W6	62	B_D14	K2	25
A_A10	W14	95	A_$\overline{\text{REG}}$	K17	130	B_A4	V6	60	B_D15	K5	27
A_A11	P14	100	A_RESET	L18	124	B_A5	U6	59	B_$\overline{\text{INPACK}}$	R7	61
A_A12	N18	117	A_$\overline{\text{VS1}}$	J18	134	B_A6	V5	57	B_$\overline{\text{IORD}}$	L5	33
A_A13	R17	106	A_$\overline{\text{VS2}}$	M19	122	B_A7	U5	54	B_$\overline{\text{IOWR}}$	M2	35
A_A14	N14	108	A_$\overline{\text{WAIT}}$	J14	136	B_A8	N1	39	B_$\overline{\text{OE}}$	L6	32
A_A15	M14	115	A_$\overline{\text{WE}}$	R19	110	B_A9	M3	36	B_READY($\overline{\text{IREQ}}$)	V8	69
A_A16	P18	112	A_WP($\overline{\text{IOIS16}}$)	H18	139	B_A10	L1	29	B_$\overline{\text{REG}}$	P8	63
A_A17	U15	103	AD0	H5	15	B_A11	M1	34	B_RESET	W5	58
A_A18	T19	105	AD1	G1	14	B_A12	T1	52	B_$\overline{\text{VS1}}$	U8	68
A_A19	P15	107	AD2	G3	12	B_A13	N3	41	B_$\overline{\text{VS2}}$	P7	56
A_A20	R18	109	AD3	H6	11	B_A14	P1	43	B_$\overline{\text{WAIT}}$	W8	70
A_A21	P17	111	AD4	F1	10	B_A15	P5	50	B_$\overline{\text{WE}}$	P3	46
A_A22	P19	114	AD5	G5	9	B_A16	P6	48	B_WP($\overline{\text{IOIS16}}$)	U9	73
A_A23	N17	116	AD6	F2	8	B_A17	M6	37	C/$\overline{\text{BE0}}$	E2	5
A_A24	N19	118	AD7	E1	6	B_A18	N2	40	C/$\overline{\text{BE1}}$	A5	203
A_A25	M18	121	AD8	G6	4	B_A19	N6	42	C/$\overline{\text{BE2}}$	C8	192
A_BVD1($\overline{\text{STSCHG}}$/$\overline{\text{R1}}$)	H19	138	AD9	F5	3	B_A20	N5	45	C/$\overline{\text{BE3}}$	A15	162
A_BVD2($\overline{\text{SPKR}}$)	J15	137	AD10	E3	2	B_A21	R1	47	CLOCK	E19	151
A_$\overline{\text{CD1}}$	V11	82	AD11	C12	172	B_A22	R2	49	DATA	F14	152
A_$\overline{\text{CD2}}$	H17	140	AD12	A4	208	B_A23	R3	51	$\overline{\text{DEVSEL}}$	C7	197
A_$\overline{\text{CE1}}$	P13	94	AD13	E6	206	B_A24	W4	53	$\overline{\text{FRAME}}$	F8	193
A_$\overline{\text{CE2}}$	R13	97	AD14	B5	205	B_A25	R6	55	GND	B10	181
A_D0	H14	141	AD15	F6	204	B_BVD1($\overline{\text{STSCHG}}$/$\overline{\text{R1}}$)	V9	72	GND	C5	207
A_D1	G18	144	AD16	B8	191	B_BVD2($\overline{\text{SPKR}}$)	W9	71	GND	E8	194
A_D2	G14	146	AD17	A8	190	B_$\overline{\text{CD1}}$	H3	16	GND	E18	153
A_D3	U11	83	AD18	E9	189	B_$\overline{\text{CD2}}$	R9	74	GND	F12	167
A_D4	R11	85	AD19	F9	188	B_$\overline{\text{CE1}}$	K6	28	GND	G2	13
A_D5	U12	88	AD20	B9	186	B_$\overline{\text{CE2}}$	L2	30	GND	J5	22
A_D6	R12	90	AD21	A9	185	B_D0	W10	76	GND	K18	129
A_D7	V13	92	AD22	F10	184	B_D1	U10	78	GND	P2	44
A_D8	H15	142	AD23	E10	183	B_D2	P10	80	GND	P9	75
A_D9	G17	145	AD24	F11	179	B_D3	H2	17	GND	V14	96
A_D10	F19	147	AD25	E13	165	B_D4	J1	19	$\overline{\text{GNT}}$	C13	168

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

SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.		SIGNAL NAME	PIN NO.	
	GHK	PDV		GHK	PDV		GHK	PDV		GHK	PDV
$\overline{\text{GRST}}$	A11	175	MFUNC5	F13	160	SPKROUT	G15	149	V_{CC}	L3	31
IDSEL	C10	182	MFUNC6	B15	161	$\overline{\text{STOP}}$	F7	198	V_{CC}	N15	113
$\overline{\text{IRDY}}$	A7	195	PAR	C6	202	$\overline{\text{SUSPEND}}$	D19	156	V_{CC}	U7	64
LATCH	F17	150	PCLK	A10	180	$\overline{\text{TRDY}}$	B7	196	V_{CC}	W12	86
MFUNC0	F15	154	$\overline{\text{PERR}}$	A6	199	V_{CC}	B14	164	V_{CCA}	M17	120
MFUNC1	E17	155	$\overline{\text{PRST}}$	A14	166	V_{CC}	C9	187	V_{CCB}	M5	38
MFUNC2	A16	157	$\overline{\text{REQ}}$	B13	169	V_{CC}	E7	201	V_{CCI}	F18	148
MFUNC3	C15	158	$\overline{\text{RI\_OUT}}/\overline{\text{PME}}$	C14	163	V_{CC}	F3	7	V_{CCP}	D1	1
MFUNC4	E14	159	$\overline{\text{SERR}}$	B6	200	V_{CC}	G19	143	V_{CCP}	E11	178

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

Table 2–5. Power Supply

TERMINAL NAME	TERMINAL NO. PDV	TERMINAL NO. GHK	DESCRIPTION
GND	13, 22, 44, 75, 96, 129, 153, 167, 181, 194, 207	B10, C5, E8, E18, F12, G2, J5, K18, P2, P9, V14	Device ground terminals
V_{CC}	7, 31, 64, 86, 113, 143, 164, 187, 201	B14, C9, E7, F3, G19, L3, N15, U7, W12	Power supply terminal for core logic (3.3 V)
V_{CCA}	120	M17	Clamp voltage for PC Card A interface. Matches Card A signaling environment, 5 V or 3.3 V.
V_{CCB}	38	M5	Clamp voltage for PC Card B interface. Matches Card B signaling environment, 5 V or 3.3 V.
V_{CCI}	148	F18	Clamp voltage for interrupt subsystem interface and miscellaneous I/O, 5 V or 3.3 V
V_{CCP}	1, 178	D1, E11	Clamp voltage for PCI signaling, 5 V or 3.3 V

Table 2–6. PC Card Power Switch

TERMINAL NAME	TERMINAL NO. PDV	TERMINAL NO. GHK	I/O	DESCRIPTION
CLOCK	151	E19	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 PCI1420 output by using bit 27 (P2CCLK) in the system control register (see Section 4.29). The TPS2206 defines the maximum frequency of this signal to be 2 MHz. If a system design defines this terminal as an output, then this terminal requires an external pulldown resistor. The frequency of the PCI1420 output CLOCK is derived from dividing the PCI CLK by 36.
DATA	152	F14	O	Power switch data. DATA is used to serially communicate socket power control information to the power switch.
LATCH	150	F17	O	Power switch latch. LATCH is asserted by the PCI1420 to indicate to the power switch that the data on the DATA line is valid. When a pulldown resistor is implemented on this terminal, the MFUNC1 and MFUNC4 terminals provide the serial EEPROM SDA and SCL interface.

Table 2–7. PCI System

TERMINAL NAME	TERMINAL NO. PDV	TERMINAL NO. GHK	I/O	DESCRIPTION
$\overline{\text{GRST}}$	175	A11	I	Global reset. When the global reset is asserted, the $\overline{\text{GRST}}$ signal causes the PCI1420 to place all output buffers in a high-impedance state and reset all internal registers. When $\overline{\text{GRST}}$ is asserted, the device is completely in its default state. For systems that require wake-up from D3, $\overline{\text{GRST}}$ will normally be asserted only during initial boot. $\overline{\text{PRST}}$ should be asserted following initial boot so that PME context is retained when transitioning from D3 to D0. For systems that do not require wake-up from D3, $\overline{\text{GRST}}$ should be tied to $\overline{\text{PRST}}$. When the $\overline{\text{SUSPEND}}$ mode is enabled, the device is protected from the $\overline{\text{GRST}}$, and the internal registers are preserved. All outputs are placed in a high-impedance state, but the contents of the registers are preserved.
PCLK	180	A10	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.
$\overline{\text{PRST}}$	166	A14	I	PCI reset. When the PCI bus reset is asserted, $\overline{\text{PRST}}$ causes the PCI1420 to place all output buffers in a high-impedance state and reset internal registers. When $\overline{\text{PRST}}$ is asserted, the device is completely nonfunctional. After $\overline{\text{PRST}}$ is deasserted, the PCI1420 is in a default state. When the $\overline{\text{SUSPEND}}$ mode is enabled, the device is protected from the $\overline{\text{PRST}}$, and the internal registers are preserved. All outputs are placed in a high-impedance state, but the contents of the registers are preserved.

Table 2–8. PCI Address and Data

TERMINAL NAME	TERMINAL NO. PDV	TERMINAL NO. GHK	I/O	DESCRIPTION
AD31	170	A13	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	171	E12		
AD29	173	B12		
AD28	174	A12		
AD27	176	B11		
AD26	177	C11		
AD25	165	E13		
AD24	179	F11		
AD23	183	E10		
AD22	184	F10		
AD21	185	A9		
AD20	186	B9		
AD19	188	F9		
AD18	189	E9		
AD17	190	A8		
AD16	191	B8		
AD15	204	F6		
AD14	205	B5		
AD13	206	E6		
AD12	208	A4		
AD11	172	C12		
AD10	2	E3		
AD9	3	F5		
AD8	4	G6		
AD7	6	E1		
AD6	8	F2		
AD5	9	G5		
AD4	10	F1		
AD3	11	H6		
AD2	12	G3		
AD1	14	G1		
AD0	15	H5		
C/$\overline{BE3}$	162	A15	I/O	PCI bus commands and byte enables. These signals are multiplexed on the same PCI terminals. During the address phase of a primary bus PCI cycle, C/$\overline{BE3}$–C/$\overline{BE0}$ define the bus command. During the data phase, this 4-bit bus is used as byte enables. The byte enables determine which byte paths of the full 32-bit data bus carry meaningful data. C/$\overline{BE0}$ applies to byte 0 (AD7–AD0), C/$\overline{BE1}$ applies to byte 1 (AD15–AD8), C/$\overline{BE2}$ applies to byte 2 (AD23–AD16), and C/$\overline{BE3}$ applies to byte 3 (AD31–AD24).
C/$\overline{BE2}$	192	C8		
C/$\overline{BE1}$	203	A5		
C/$\overline{BE0}$	5	E2		
PAR	202	C6	I/O	PCI bus parity. In all PCI bus read and write cycles, the PCI1420 calculates even parity across the AD31–AD0 and C/$\overline{BE3}$–C/$\overline{BE0}$ buses. As an initiator during PCI cycles, the PCI1420 outputs this parity indicator with a one-PCLK delay. As a target during PCI cycles, the calculated parity is compared to the initiator's parity indicator. A compare error results in the assertion of a parity error ($\overline{PERR}$).

Table 2–9. PCI Interface Control

TERMINAL NAME	TERMINAL NO. PDV	TERMINAL NO. GHK	I/O	DESCRIPTION
$\overline{\text{DEVSEL}}$	197	C7	I/O	PCI device select. The PCI1420 asserts $\overline{\text{DEVSEL}}$ to claim a PCI cycle as the target device. As a PCI initiator on the bus, the PCI1420 monitors $\overline{\text{DEVSEL}}$ until a target responds. If no target responds before timeout occurs, then the PCI1420 terminates the cycle with an initiator abort.
$\overline{\text{FRAME}}$	193	F8	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.
$\overline{\text{GNT}}$	168	C13	I	PCI bus grant. $\overline{\text{GNT}}$ is driven by the PCI bus arbiter to grant the PCI1420 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	182	C10	I	Initialization device select. IDSEL selects the PCI1420 during configuration space accesses. IDSEL can be connected to one of the upper 24 PCI address lines on the PCI bus.
$\overline{\text{IRDY}}$	195	A7	I/O	PCI initiator ready. $\overline{\text{IRDY}}$ indicates the PCI bus initiator's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCLK where both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. Until $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are both sampled asserted, wait states are inserted.
$\overline{\text{PERR}}$	199	A6	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 (see Section 4.4).
$\overline{\text{REQ}}$	169	B13	O	PCI bus request. $\overline{\text{REQ}}$ is asserted by the PCI1420 to request access to the PCI bus as an initiator.
$\overline{\text{SERR}}$	200	B6	O	PCI system error. $\overline{\text{SERR}}$ is an output that is pulsed from the PCI1420 when enabled through bit 8 of the command register (see Section 4.4) indicating a system error has occurred. The PCI1420 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.
$\overline{\text{STOP}}$	198	F7	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.
$\overline{\text{TRDY}}$	196	B7	I/O	PCI target ready. $\overline{\text{TRDY}}$ indicates the primary bus target's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCLK when both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. Until both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted, wait states are inserted.

Table 2–10. Multifunction and Miscellaneous Pins

TERMINAL NAME	TERMINAL NO. PDV	TERMINAL NO. GHK	I/O	DESCRIPTION
MFUNC0	154	F15	I/O	Multifunction terminal 0. MFUNC0 can be configured as parallel PCI interrupt $\overline{\text{INTA}}$, GPI0, GPO0, socket activity LED output, ZV switching outputs, CardBus audio PWM, $\overline{\text{GPE}}$, or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details.
MFUNC1	155	E17	I/O	Multifunction terminal 1. MFUNC1 can be configured as parallel PCI interrupt $\overline{\text{INTB}}$, GPI1, GPO1, socket activity LED output, ZV switching outputs, CardBus audio PWM, $\overline{\text{GPE}}$, or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details. Serial data (SDA). When LATCH is detected low after a PCI reset, the MFUNC1 terminal provides the SDA signaling for the serial bus interface. The two-pin serial interface loads the subsystem identification and other register defaults from an EEPROM after a PCI reset. See Section 3.6.1, *Serial Bus Interface Implementation*, for details on other serial bus applications.
MFUNC2	157	A16	I/O	Multifunction terminal 2. MFUNC2 can be configured as PC/PCI DMA request, GPI2, GPO2, socket activity LED output, ZV switching outputs, CardBus audio PWM, $\overline{\text{GPE}}$, RI_OUT, or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details.
MFUNC3	158	C15	I/O	Multifunction terminal 3. MFUNC3 can be configured as a parallel IRQ or the serialized interrupt signal IRQSER. See Section 4.30, *Multifunction Routing Register*, for configuration details.
MFUNC4	159	E14	I/O	Multifunction terminal 4. MFUNC4 can be configured as PCI $\overline{\text{LOCK}}$, GPI3, GPO3, socket activity LED output, ZV switching outputs, CardBus audio PWM, $\overline{\text{GPE}}$, RI_OUT, or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details. Serial clock (SCL). When LATCH is detected low after a PCI reset, the MFUNC4 terminal provides the SCL signaling for the serial bus interface. The two-pin serial interface loads the subsystem identification and other register defaults from an EEPROM after a PCI reset. See Section 3.6.1, *Serial Bus Interface Implementation*, for details on other serial bus applications.
MFUNC5	160	F13	I/O	Multifunction terminal 5. MFUNC5 can be configured as PC/PCI DMA grant, GPI4, GPO4, socket activity LED output, ZV switching outputs, CardBus audio PWM, $\overline{\text{GPE}}$, or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details.
MFUNC6	161	B15	I/O	Multifunction terminal 6. MFUNC6 can be configured as a PCI $\overline{\text{CLKRUN}}$ or a parallel IRQ. See Section 4.30, *Multifunction Routing Register*, for configuration details.
$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$	163	C14	O	Ring indicate out and power management event output. Terminal provides an output for ring-indicate or $\overline{\text{PME}}$ signals.
SPKROUT	149	G15	O	Speaker output. SPKROUT is the output to the host system that can carry $\overline{\text{SPKR}}$ or CAUDIO through the PCI1420 from the PC Card interface. SPKROUT is driven as the exclusive-OR combination of card $\overline{\text{SPKR}}$//CAUDIO inputs.
$\overline{\text{SUSPEND}}$	156	D19	I	Suspend. $\overline{\text{SUSPEND}}$ protects the internal registers from clearing when the $\overline{\text{GRST}}$ or $\overline{\text{PRST}}$ signal is asserted. See Section 3.8.4, *Suspend Mode*, for details.

Table 2–11. 16-Bit PC Card Address and Data (Slots A and B)

TERMINAL					I/O	DESCRIPTION
NAME	NUMBER					
	SLOT A†		SLOT B‡			
	PDV	GHK	PDV	GHK		
A25	121	M18	55	R6	O	PC Card address. 16-bit PC Card address lines. A25 is the most significant bit.
A24	118	N19	53	W4		
A23	116	N17	51	R3		
A22	114	P19	49	R2		
A21	111	P17	47	R1		
A20	109	R18	45	N5		
A19	107	P15	42	N6		
A18	105	T19	40	N2		
A17	103	U15	37	M6		
A16	112	P18	48	P6		
A15	115	M14	50	P5		
A14	108	N14	43	P1		
A13	106	R17	41	N3		
A12	117	N18	52	T1		
A11	100	P14	34	M1		
A10	95	W14	29	L1		
A9	102	R14	36	M3		
A8	104	W16	39	N1		
A7	119	M15	54	U5		
A6	123	L19	57	V5		
A5	125	L17	59	U6		
A4	126	L15	60	V6		
A3	128	K19	62	W6		
A2	131	K15	65	V7		
A1	132	K14	66	W7		
A0	133	J19	67	R8		
D15	93	U13	27	K5	I/O	PC Card data. 16-bit PC Card data lines. D15 is the most significant bit.
D14	91	W13	25	K2		
D13	89	P12	23	J6		
D12	87	V12	20	J2		
D11	84	P11	18	H1		
D10	147	F19	81	W11		
D9	145	G17	79	R10		
D8	142	H15	77	V10		
D7	92	V13	26	K3		
D6	90	R12	24	K1		
D5	88	U12	21	J3		
D4	85	R11	19	J1		
D3	83	U11	17	H2		
D2	146	G14	80	P10		
D1	144	G18	78	U10		
D0	141	H14	76	W10		

† Terminal name for slot A is preceded with A_. For example, the full name for terminals 121 and M18 are A_A25.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 55 and R6 are B_A25.

Table 2–12. 16-Bit PC Card Interface Control (Slots A and B)

TERMINAL NAME	TERMINAL NUMBER SLOT A† PDV	TERMINAL NUMBER SLOT A† GHK	TERMINAL NUMBER SLOT B‡ PDV	TERMINAL NUMBER SLOT B‡ GHK	I/O	DESCRIPTION
BVD1 (STSCHG/RI)	138	H19	72	V9	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 dead condition of a 16-bit I/O PC Card. Ring indicate. RI is used by 16-bit modem cards to indicate a ring detection.
BVD2 (SPKR)	137	J15	71	W9	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 PCI1420 and are output on SPKROUT. DMA request. BVD2 can be used as the DMA request signal during DMA operations to a 16-bit PC Card that supports DMA. The PC Card asserts BVD2 to indicate a request for a DMA operation.
CD1 CD2	82 140	V11 H17	16 74	H3 R9	I	Card detect 1 and Card detect 2. CD1 and CD2 are internally connected to ground on the PC Card. When a PC Card is inserted into a socket, CD1 and CD2 are pulled low. For signal status, see Section 5.2, *ExCA Interface Status Register.*
CE1 CE2	94 97	P13 R13	28 30	K6 L2	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	127	L14	61	R7	I	Input acknowledge. INPACK is asserted by the PC Card when it can respond to an I/O read cycle at the current address. DMA request. INPACK can be used as the DMA request signal during DMA operations from a 16-bit PC Card that supports DMA. If it is used as a strobe, then the PC Card asserts this signal to indicate a request for a DMA operation.
IORD	99	W15	33	L5	O	I/O read. IORD is asserted by the PCI1420 to enable 16-bit I/O PC Card data output during host I/O read cycles. DMA write. IORD is used as the DMA write strobe during DMA operations from a 16-bit PC Card that supports DMA. The PCI1420 asserts IORD during DMA transfers from the PC Card to host memory.
IOWR	101	V15	35	M2	O	I/O write. IOWR is driven low by the PCI1420 to strobe write data into 16-bit I/O PC Cards during host I/O write cycles. DMA read. IOWR is used as the DMA write strobe during DMA operations from a 16-bit PC Card that supports DMA. The PCI1420 asserts IOWR during transfers from host memory to the PC Card.

† Terminal name for slot A is preceded with A_. For example, the full name for terminals 127 and L14 are A_INPACK.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 61 and R7 are B_INPACK.

Table 2–12. 16-Bit PC Card Interface Control (Slots A and B) (Continued)

TERMINAL NAME	TERMINAL NUMBER SLOT A† PDV	TERMINAL NUMBER SLOT A† GHK	TERMINAL NUMBER SLOT B‡ PDV	TERMINAL NUMBER SLOT B‡ GHK	I/O	DESCRIPTION
$\overline{OE}$	98	U14	32	L6	O	Output enable. $\overline{OE}$ is driven low by the PCI1420 to enable 16-bit memory PC Card data output during host memory read cycles. DMA terminal count. $\overline{OE}$ is used as terminal count (TC) during DMA operations to a 16-bit PC Card that supports DMA. The PCI1420 asserts $\overline{OE}$ to indicate TC for a DMA write operation.
READY ($\overline{IREQ}$)	135	J17	69	V8	I	Ready. The ready function is provided by READY when the 16-bit PC Card and the host socket are configured for the memory-only interface. READY is driven low by the 16-bit memory PC Cards to indicate that the memory card circuits are busy processing a previous write command. READY is driven high when the 16-bit memory PC Card is ready to accept a new data transfer command. Interrupt request. $\overline{IREQ}$ is asserted by a 16-bit I/O PC Card to indicate to the host that a device on the 16-bit I /O PC Card requires service by the host software. $\overline{IREQ}$ is high (deasserted) when no interrupt is requested.
$\overline{REG}$	130	K17	63	P8	O	Attribute memory select. $\overline{REG}$ remains high for all common memory accesses. When $\overline{REG}$ is asserted, access is limited to attribute memory ($\overline{OE}$ or $\overline{WE}$ active) and to the I/O space ($\overline{IORD}$ or $\overline{IOWR}$ active). Attribute memory is a separately accessed section of card memory and is generally used to record card capacity and other configuration and attribute information. DMA acknowledge. $\overline{REG}$ is used as a DMA acknowledge ($\overline{DACK}$) during DMA operations to a 16-bit PC Card that supports DMA. The PCI1420 asserts $\overline{REG}$ to indicate a DMA operation. $\overline{REG}$ is used in conjunction with the DMA read ($\overline{IOWR}$) or DMA write ($\overline{IORD}$) strobes to transfer data.
RESET	124	L18	58	W5	O	PC Card reset. RESET forces a hard reset to a 16-bit PC Card.
$\overline{WAIT}$	136	J14	70	W8	I	Bus cycle wait. $\overline{WAIT}$ is driven by a 16-bit PC Card to extend the completion of the memory or I/O cycle in progress.
$\overline{WE}$	110	R19	46	P3	O	Write enable. $\overline{WE}$ is used to strobe memory write data into 16-bit memory PC Cards. $\overline{WE}$ is also used for memory PC Cards that employ programmable memory technologies. DMA terminal count. $\overline{WE}$ is used as $\overline{TC}$ during DMA operations to a 16-bit PC Card that supports DMA. The PCI1420 asserts $\overline{WE}$ to indicate $\overline{TC}$ for a DMA read operation.
WP ($\overline{IOIS16}$)	139	H18	73	U9	I	Write protect. WP applies to 16-bit memory PC Cards. WP reflects the status of the write-protect switch on 16-bit memory PC Cards. For 16-bit I/O cards, WP is used for the 16-bit port ($\overline{IOIS16}$) function. I/O is 16 bits. $\overline{IOIS16}$ applies to 16-bit I/O PC Cards. $\overline{IOIS16}$ is asserted by the 16-bit PC Card when the address on the bus corresponds to an address to which the 16-bit PC Card responds, and the I/O port that is addressed is capable of 16-bit accesses. DMA request. WP can be used as the DMA request signal during DMA operations to a 16-bit PC Card that supports DMA. If used, then the PC Card asserts WP to indicate a request for a DMA operation.
$\overline{VS1}$ $\overline{VS2}$	134 122	J18 M19	68 56	U8 P7	I/O	Voltage sense 1 and voltage sense 2. $\overline{VS1}$ and $\overline{VS2}$, when used in conjunction with each other, determine the operating voltage of the PC Card.

† Terminal name for slot A is preceded with A_. For example, the full name for terminals 110 and R19 are A_$\overline{WE}$.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 46 and P3 are B_$\overline{WE}$.

Table 2–13. CardBus PC Card Interface System (Slots A and B)

TERMINAL					I/O	DESCRIPTION
NAME	NUMBER					
	SLOT A†		SLOT B‡			
	PDV	GHK	PDV	GHK		
CCLK	112	P18	48	P6	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.
$\overline{\text{CCLKRUN}}$	139	H18	73	U9	O	CardBus clock run. $\overline{\text{CCLKRUN}}$ is used by a CardBus PC Card to request an increase in the CCLK frequency, and by the PCI1420 to indicate that the CCLK frequency is going to be decreased.
$\overline{\text{CRST}}$	124	L18	58	W5	I/O	CardBus reset. $\overline{\text{CRST}}$ brings CardBus PC Card-specific registers, sequencers, and signals to a known state. When $\overline{\text{CRST}}$ is asserted, all CardBus PC Card signals are placed in a high-impedance state, and the PCI1420 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 112 and P18 are A_CCLK.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 48 and P6 are B_CCLK.

Table 2–14. CardBus PC Card Address and Data (Slots A and B)

TERMINAL NAME	TERMINAL NUMBER SLOT A† PDV	TERMINAL NUMBER SLOT A† GHK	TERMINAL NUMBER SLOT B‡ PDV	TERMINAL NUMBER SLOT B‡ GHK	I/O	DESCRIPTION
CAD31	147	F19	81	W11	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	145	G17	79	R10		
CAD29	144	G18	78	U10		
CAD28	142	H15	77	V10		
CAD27	141	H14	76	W10		
CAD26	133	J19	67	R8		
CAD25	132	K14	66	W7		
CAD24	131	K15	65	V7		
CAD23	128	K19	62	W6		
CAD22	126	L15	60	V6		
CAD21	125	L17	59	U6		
CAD20	123	L19	57	V5		
CAD19	121	M18	55	R6		
CAD18	119	M15	54	U5		
CAD17	118	N19	53	W4		
CAD16	103	U15	37	M6		
CAD15	101	V15	35	M2		
CAD14	102	R14	36	M3		
CAD13	99	W15	33	L5		
CAD12	100	P14	34	M1		
CAD11	98	U14	32	L6		
CAD10	97	R13	30	L2		
CAD9	95	W14	29	L1		
CAD8	93	U13	27	K5		
CAD7	92	V13	26	K3		
CAD6	89	P12	23	J6		
CAD5	90	R12	24	K1		
CAD4	87	V12	20	J2		
CAD3	88	U12	21	J3		
CAD2	84	P11	18	H1		
CAD1	85	R11	19	J1		
CAD0	83	U11	17	H2		
CC/$\overline{BE3}$	130	K17	63	P8	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/$\overline{BE2}$	117	N18	52	T1		
CC/$\overline{BE1}$	104	W16	39	N1		
CC/$\overline{BE0}$	94	P13	28	K6		
CPAR	106	R17	41	N3	I/O	CardBus parity. In all CardBus read and write cycles, the PCI1420 calculates even parity across the CAD and CC/$\overline{BE}$ buses. As an initiator during CardBus cycles, the PCI1420 outputs CPAR with a one-CCLK delay. As a target during CardBus cycles, the calculated parity is compared to the initiator's parity indicator; a compare error results in a parity error assertion.

† Terminal name for slot A is preceded with A_. For example, the full name for terminals 106 and R17 are A_CPAR.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 41 and N3 are B_CPAR.

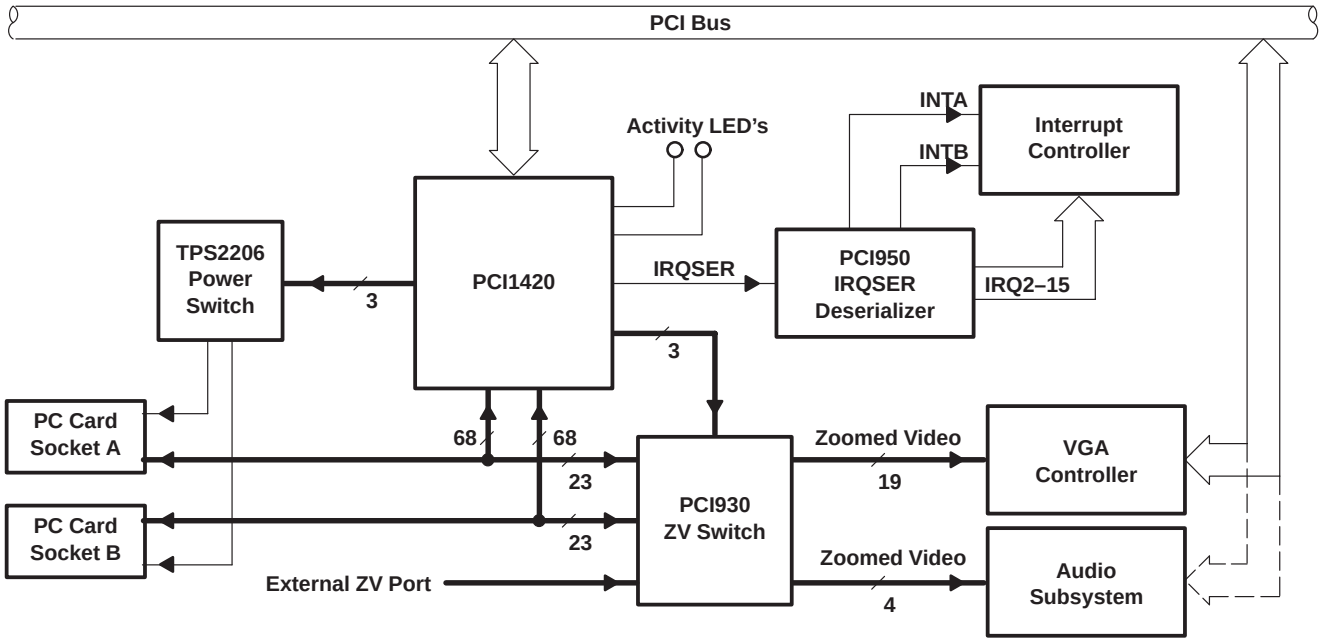
Table 2–15. CardBus PC Card Interface Control (Slots A and B)

TERMINAL NAME	TERMINAL NUMBER SLOT A† PDV	TERMINAL NUMBER SLOT A† GHK	TERMINAL NUMBER SLOT B‡ PDV	TERMINAL NUMBER SLOT B‡ GHK	I/O	DESCRIPTION
CAUDIO	137	J15	71	W9	I	CardBus audio. CAUDIO is a digital input signal from a PC Card to the system speaker. The PCI1420 supports the binary audio mode and outputs a binary signal from the card to SPKROUT.
$\overline{\text{CBLOCK}}$	107	P15	42	N6	I/O	CardBus lock. $\overline{\text{CBLOCK}}$ is used to gain exclusive access to a target.
$\overline{\text{CCD1}}$ $\overline{\text{CCD2}}$	82 140	V11 H17	16 74	H3 R9	I	CardBus detect 1 and CardBus detect 2. $\overline{\text{CCD1}}$ and $\overline{\text{CCD2}}$ are used in conjunction with CVS1 and CVS2 to identify card insertion and interrogate cards to determine the operating voltage and card type.
$\overline{\text{CDEVSEL}}$	111	P17	47	R1	I/O	CardBus device select. The PCI1420 asserts $\overline{\text{CDEVSEL}}$ to claim a CardBus cycle as the target device. As a CardBus initiator on the bus, the PCI1420 monitors $\overline{\text{CDEVSEL}}$ until a target responds. If no target responds before timeout occurs, then the PCI1420 terminates the cycle with an initiator abort.
$\overline{\text{CFRAME}}$	116	N17	51	R3	I/O	CardBus cycle frame. $\overline{\text{CFRAME}}$ is driven by the initiator of a CardBus bus cycle. $\overline{\text{CFRAME}}$ is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When $\overline{\text{CFRAME}}$ is deasserted, the CardBus bus transaction is in the final data phase.
$\overline{\text{CGNT}}$	110	R19	46	P3	I	CardBus bus grant. $\overline{\text{CGNT}}$ is driven by the PCI1420 to grant a CardBus PC Card access to the CardBus bus after the current data transaction has been completed.
$\overline{\text{CINT}}$	135	J17	69	V8	I	CardBus interrupt. $\overline{\text{CINT}}$ is asserted low by a CardBus PC Card to request interrupt servicing from the host.
$\overline{\text{CIRDY}}$	115	M14	50	P5	I/O	CardBus initiator ready. $\overline{\text{CIRDY}}$ indicates the CardBus initiator's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of CCLK when both $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are asserted. Until $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are both sampled asserted, wait states are inserted.
$\overline{\text{CPERR}}$	108	N14	43	P1	I/O	CardBus parity error. $\overline{\text{CPERR}}$ reports parity errors during CardBus transactions, except during special cycles. It is driven low by a target two clocks following that data when a parity error is detected.
$\overline{\text{CREQ}}$	127	L14	61	R7	I	CardBus request. $\overline{\text{CREQ}}$ indicates to the arbiter that the CardBus PC Card desires use of the CardBus bus as an initiator.
$\overline{\text{CSERR}}$	136	J14	70	W8	I	CardBus system error. $\overline{\text{CSERR}}$ reports address parity errors and other system errors that could lead to catastrophic results. $\overline{\text{CSERR}}$ is driven by the card synchronous to CCLK, but deasserted by a weak pullup, and may take several CCLK periods. The PCI1420 can report $\overline{\text{CSERR}}$ to the system by assertion of $\overline{\text{SERR}}$ on the PCI interface.
$\overline{\text{CSTOP}}$	109	R18	45	N5	I/O	CardBus stop. $\overline{\text{CSTOP}}$ is driven by a CardBus target to request the initiator to stop the current CardBus transaction. $\overline{\text{CSTOP}}$ is used for target disconnects, and is commonly asserted by target devices that do not support burst data transfers.
CSTSCHG	138	H19	72	V9	I	CardBus status change. CSTSCHG alerts the system to a change in the card's status, and is used as a wake-up mechanism.
$\overline{\text{CTRDY}}$	114	P19	49	R2	I/O	CardBus target ready. $\overline{\text{CTRDY}}$ indicates the CardBus target's ability to complete the current data phase of the transaction. A data phase is completed on a rising edge of CCLK, when both $\overline{\text{CIRDY}}$ and $\overline{\text{CTRDY}}$ are asserted; until this time, wait states are inserted.
CVS1 CVS2	134 122	J18 M19	68 56	U8 P7	I/O	CardBus voltage sense 1 and CardBus voltage sense 2. CVS1 and CVS2 are used in conjunction with $\overline{\text{CCD1}}$ and $\overline{\text{CCD2}}$ to identify card insertion and interrogate cards to determine the operating voltage and card type.

† Terminal name for slot A is preceded with A_. For example, the full name for terminals 137 and J15 are A_CAUDIO.
‡ Terminal name for slot B is preceded with B_. For example, the full name for terminals 71 and W9 are B_CAUDIO.

3 Feature/Protocol Descriptions

The following sections give an overview of the PCI1420. Figure 3–1 shows a simplified block diagram of the PCI1420. The PCI interface includes all address/data and control signals for PCI protocol. The interrupt interface includes terminals for parallel PCI, parallel ISA, and serialized PCI and ISA signaling. Miscellaneous system interface terminals include multifunction terminals: $\overline{\text{SUSPEND}}$, $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ (power management control signal), and SPKROUT.



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

Figure 3–1. PCI1420 Simplified Block Diagram

3.1 Power Supply Sequencing

The PCI1420 contains 3.3-V I/O buffers with 5-V tolerance requiring a core power supply and clamp voltages. The core power supply is always 3.3 V. The clamp voltages can be either 3.3 V or 5 V, depending on the interface. The following power-up and power-down sequences are recommended.

The power-up sequence is:

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

The power-down sequence is:

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

3.2 I/O Characteristics

Figure 3–2 shows a 3-state bidirectional buffer. Section 8.2, *Recommended Operating Conditions*, provides the electrical characteristics of the inputs and outputs.

NOTE: The PCI1420 meets the ac specifications of the *1997 PC Card Standard* and *PCI Local Bus Specification.*

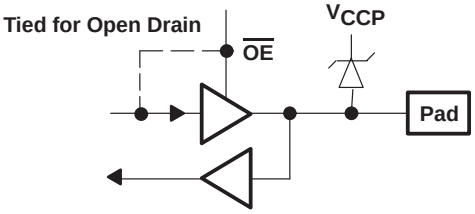


Figure 3–2. 3-State Bidirectional Buffer

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

3.3 Clamping Voltages

The clamping voltages are set to match whatever external environment the PCI1420 will be interfaced with: 3.3 V or 5 V. The I/O sites can be pulled through a clamping diode to a voltage rail that protects the core from external signals. The core power supply is always 3.3 V and is independent of the clamping voltages. For example, PCI signaling can be either 3.3 V or 5 V, and the PCI1420 must reliably accommodate both voltage levels. This is accomplished by using a 3.3-V I/O buffer that is 5-V tolerant, with the applicable clamping voltage applied. If a system designer desires a 5-V PCI bus, then V_{CCP} can be connected to a 5-V power supply.

The PCI1420 requires four separate clamping voltages because it supports a wide range of features. The four voltages are listed and defined in Section 8.2, *Recommended Operating Conditions*.

3.4 Peripheral Component Interconnect (PCI) Interface

The PCI1420 is fully compliant with the *PCI Local Bus Specification*. The PCI1420 provides all required signals for PCI master or slave operation, and may operate in either a 5-V or 3.3-V signaling environment by connecting the V_{CCP} terminals to the desired voltage level. In addition to the mandatory PCI signals, the PCI1420 provides the optional interrupt signals $\overline{\text{INTA}}$ and $\overline{\text{INTB}}$.

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

The bus-locking protocol defined in the *PCI Local Bus Specification* is not highly recommended, but is provided on the PCI1420 as an additional compatibility feature. The PCI $\overline{\text{LOCK}}$ signal can be routed to the MFUNC4 terminal via the multifunction routing register. See Section 4.30, *Multifunction Routing Register*, for details. Note that the use of $\overline{\text{LOCK}}$ is only supported by PCI-to-CardBus bridges in the downstream direction (away from the processor).

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

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

The PCI bus arbiter may be designed to support only complete bus locks using the $\overline{\text{LOCK}}$ protocol. In this scenario, the arbiter will not grant the bus to any other agent (other than the $\overline{\text{LOCK}}$ master) while $\overline{\text{LOCK}}$ is asserted. A complete

bus lock may have a significant impact on the performance of the video. The arbiter that supports complete bus lock must grant the bus to the cache to perform a writeback due to a snoop to a modified line when a locked operation is in progress.

The PCI1420 supports all $\overline{\text{LOCK}}$ protocol associated with PCI-to-PCI bridges, as also defined for PCI-to-CardBus bridges. This includes disabling write posting while a locked operation is in progress, which can solve a potential deadlock when using devices such as PCI-to-PCI bridges. The potential deadlock can occur if a CardBus target supports delayed transactions and blocks access to the target until it completes a delayed read. This target characteristic is prohibited by the *PCI Local Bus Specification*, and the issue is resolved by the PCI master using $\overline{\text{LOCK}}$.

3.4.2 Loading Subsystem Identification

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

The PCI1420 offers two mechanisms to load a read-only value into the subsystem registers. The first mechanism relies upon the system BIOS providing the subsystem ID value. The default access mode to the subsystem registers is read-only, but can be made read/write by setting bit 5 (SUBSYSRW) in the system control register (see Section 4.29) at PCI offset 80h. Once this bit is set, the BIOS can write a subsystem identification value into the registers at PCI offset 40h. The BIOS must clear the SUBSYSRW bit such that the subsystem vendor ID register and subsystem ID register is limited to read-only access. This approach saves the added cost of implementing the serial electrically erasable programmable ROM (EEPROM).

In some conditions, such as in a docking environment, the subsystem vendor ID register and subsystem ID register must be loaded with a unique identifier via a serial EEPROM. The PCI1420 loads the data from the serial EEPROM after a reset of the primary bus. Note that the $\overline{\text{SUSPEND}}$ input gates the PCI reset from the entire PCI1420 core, including the serial bus state machine (see Section 3.8.4, *Suspend Mode*, for details on using $\overline{\text{SUSPEND}}$).

The PCI1420 provides a two-line serial bus host controller that can interface to a serial EEPROM. See Section 3.6, *Serial Bus Interface*, for details on the two-wire serial bus controller and applications.

3.5 PC Card Applications

This section describes the PC Card interfaces of the PCI1420:

- Card insertion/removal and recognition
- P^2C power-switch interface
- Zoomed video support
- Speaker and audio applications
- LED socket activity indicators
- PC Card-16 DMA support
- CardBus socket registers

3.5.1 PC Card Insertion/Removal and Recognition

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

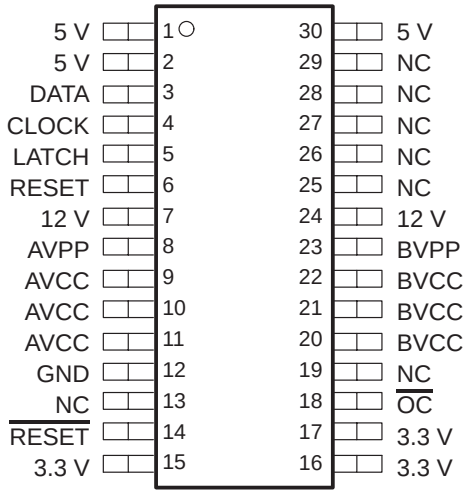
The scheme uses the card detect and voltage sense signals. The configuration of these four terminals identifies the card type and voltage requirements of the PC Card interface. The encoding scheme is defined in the *1997 PC Card Standard* and in Table 3–1.

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

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

3.5.2 P^{2}C Power-Switch Interface (TPS2206/2216)

The PCI1420 provides a P^{2}C (PCMCIA Peripheral Control) interface for control of the PC Card power switch. The CLOCK, DATA, and LATCH terminals interface with the TI TPS2206/2216 dual-slot PC Card power interface switches to provide power switch support. Figure 3–3 shows the terminal assignments of the TPS2206, and Figure 3–4 illustrates a typical application where the PCI1420 represents the PCMCIA controller.



NC – No internal connection

Figure 3–3. TPS2206 Terminal Assignments

The CLOCK terminal on the PCI1420 can be an input or an output. The PCI1420 defaults the CLOCK terminal as an input to control the serial interface and the internal state machine. Bit 27 (P2CCLK) in the system control register (see Section 4.29) can be set by the platform BIOS to enable the PCI1420 to generate and drive the CLOCK internally from the PCI clock. When the system design implements CLOCK as an output from the PCI1420, an external pulldown resistor is required.

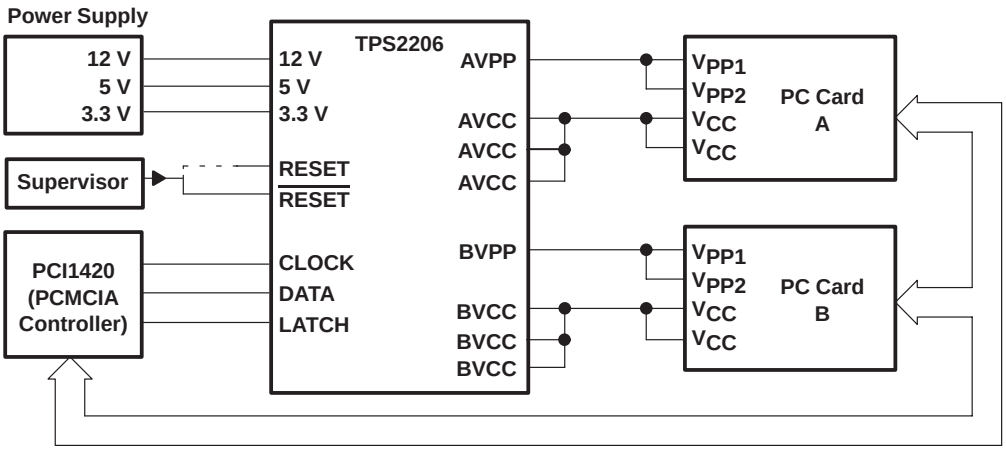


Figure 3–4. TPS2206 Typical Application

3.5.3 Zoomed Video Support

The PCI1420 allows for the implementation of zoomed video for PC Cards. Zoomed video is supported by setting bit 6 (ZVENABLE) in the card control register (see Section 4.32) on a per socket function basis. Setting this bit puts PC Card 16 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 3–5 illustrates a PCI1420 ZV implementation.

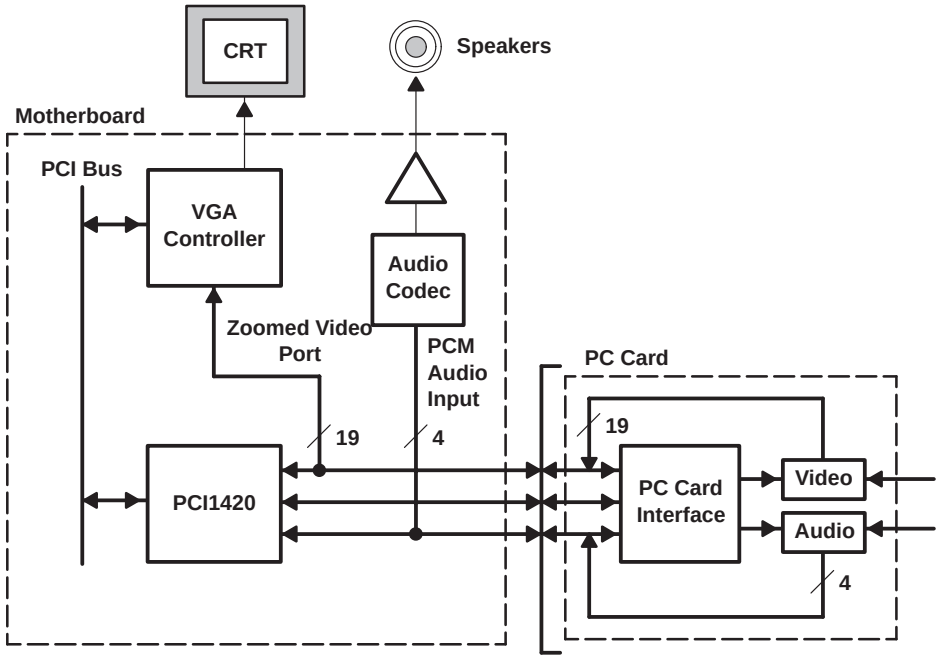


Figure 3–5. Zoomed Video Implementation Using PCI1420

Not shown in Figure 3–5 is the multiplexing scheme used to route either socket 0 or socket 1 ZV source to the graphics controller. The PCI1420 provides ZVSTAT, $\overline{\text{ZVSEL0}}$, and $\overline{\text{ZVSEL1}}$ signals on the multifunction terminals to switch external bus drivers. Figure 3–6 shows an implementation for switching between three ZV streams using external logic.

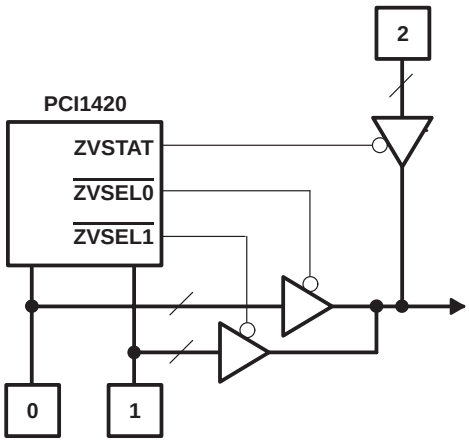


Figure 3–6. Zoomed Video Switching Application

Figure 3–6 illustrates an implementation using standard three-state bus drivers with active-low output enables. $\overline{\text{ZVSEL0}}$ is an active-low output indicating that the Socket 0 ZV mode is enabled, and $\overline{\text{ZVSEL1}}$ is an active-low output indicating that Socket 1 ZV is enabled. When both sockets have ZV mode enabled, the PCI1420 defaults to indicating socket 0 enabled through $\overline{\text{ZVSEL0}}$; however, bit 5 (PORT_SEL) in the card control register (see Section 4.32) allows software to select the socket ZV source priority. Table 3–2 illustrates the functionality of the ZV output signals.

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

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

Also shown in Figure 3–6 is a third ZV source that may be provided from a source such as a high-speed serial bus like IEEE1394. The ZVSTAT signal provides a mechanism to switch the third ZV source. ZVSTAT is an active-high output indicating that one of the PCI1420 sockets is enabled for ZV mode. The implementation shown in Figure 3–6 can be used if PC Card ZV is prioritized over other sources.

3.5.4 Ultra Zoomed Video

Ultra zoomed video is an enhancement to the PCI1420's DMA engine and is intended to improve the 16-bit bandwidth for MPEG I and MPEG II decoder PC Cards. This enhancement allows the PCI1420 to fetch 32 bits of data from memory versus the 11XX/12XX 16-bit fetch capability. This enhancement allows a higher sustained throughput to the 16-bit PC Card because the PCI1420 prefetches an extra 16 bits (32 bits total) during each PCI read transaction. If the PCI bus becomes busy, then the PCI1420 has an extra 16 bits of data to perform back-to-back 16-bit transactions to the PC Card before having to fetch more data. This feature is built into the DMA engine and software is not required to enable this enhancement.

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

3.5.5 Internal Ring Oscillator

The internal ring oscillator provides an internal clock source for the PCI1420 so that neither the PCI clock nor an external clock is required in order for the PCI1420 to power down a socket or interrogate a PC Card. This internal oscillator operates nominally at 16 kHz and can be enabled by setting bit 27 (P2CCLK) of the system control register (see Section 4.29) at PCI offset 80h to a 1. This function is disabled by default.

3.5.6 Integrated Pullup Resistors

The *1997 PC Card Standard* requires pullup resistors on various terminals to support both CardBus and 16-bit card configurations. Unlike the PCI1220/1225 which required external pullup resistors, the PCI1420 has integrated all of these pullup resistors.

SIGNAL NAME	PIN NUMBER SOCKET A		PIN NUMBER SOCKET B	
	GHK	PDV	GHK	PDV
A14/$\overline{\text{CPERR}}$	N14	108	P1	43
READY/$\overline{\text{CINT}}$	J17	135	V8	69
A15/$\overline{\text{CIRDY}}$	M14	115	P5	50
$\overline{\text{CD1}}$/$\overline{\text{CCD1}}$	V11	82	H3	16
$\overline{\text{VS1}}$/CVS1	J18	134	U8	68
A19/$\overline{\text{CBLOCK}}$	P15	107	N6	42
A20/$\overline{\text{CSTOP}}$	R18	109	N5	45
A21/$\overline{\text{CDEVSEL}}$	P17	111	R1	47
A22/$\overline{\text{CTRDY}}$	P19	114	R2	49
$\overline{\text{VS2}}$/CVS2	M19	122	P7	56
RESET/$\overline{\text{CRST}}$	L18	124	W5	58
$\overline{\text{WAIT}}$/$\overline{\text{CSERR}}$	J14	136	W8	70
$\overline{\text{INPACK}}$/$\overline{\text{CREQ}}$	L14	127	R7	61
BVD2($\overline{\text{SPKR}}$)/CAUDIO	J15	137	W9	71
BVD1($\overline{\text{STSCHG}}$)/CSTSCHG	H19	138	V9	72
$\overline{\text{CD2}}$/$\overline{\text{CCD2}}$	H17	140	R9	74

3.5.7 SPKROUT and CAUDPWM Usage

SPKROUT carries the digital audio signal from the PC Card to the system. When a 16-bit PC Card is configured for I/O mode, the BVD2 pin becomes $\overline{\text{SPKR}}$. This terminal is also used in CardBus binary audio applications, and is referred to as CAUDIO. $\overline{\text{SPKR}}$ passes a TTL level digital audio signal to the PCI1420. The CardBus CAUDIO signal also can pass a single-amplitude binary waveform. The binary audio signals from the two PC Card sockets are XOR'ed in the PCI1420 to produce SPKROUT. This output is enabled by bit 1 (SPKROUTEN) in the card control register (see Section 4.32).

Older controllers support CAUDIO in binary or PWM mode but use the same pin (SPKROUT). Some audio chips may not support both modes on one pin and may have a separate pin for binary and PWM. The PCI1420 implementation includes a signal for PWM, CAUDPWM, which can be routed to a MFUNC terminal. Bit 2 (AUD2MUX) located in the card control register is programmed on a per socket function basis to route a CardBus CAUDIO PWM terminal to CAUDPWM. If both CardBus functions enable CAUDIO PWM routing to CAUDPWM, then socket 0 audio takes precedence. See Section 4.30, *Multifunction Routing Register*, for details on configuring the MFUNC terminals.

Figure 3–7 provides an illustration of a sample application using SPKROUT and CAUDPWM.

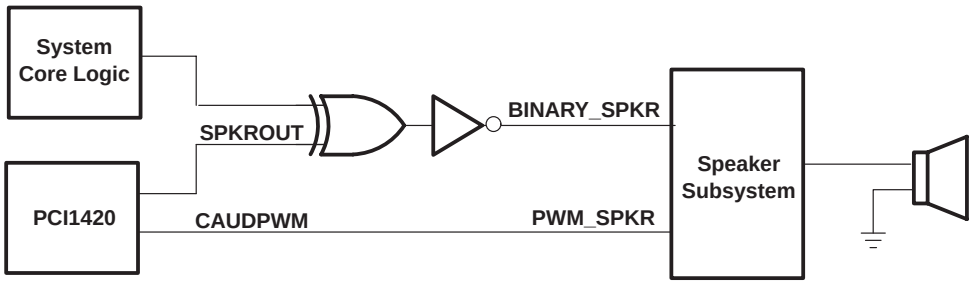


Figure 3–7. Sample Application of SPKROUT and CAUDPWM

3.5.8 LED Socket Activity Indicators

The socket activity LEDs are provided to indicate when a PC Card is being accessed. The LEDA1 and LEDA2 signals can be routed to the multifunction terminals. When configured for LED outputs, these terminals output an active high signal to indicate socket activity. LEDA1 indicates socket 0 (card A) activity, and LEDA2 indicates socket 1 (card B) activity. The LED_SKT output indicates socket activity to either socket 0 or socket 1. See Section 4.30, *Multifunction Routing Register*, for details on configuring the multifunction terminals.

The LED signal is active high and is driven for 64-ms durations. When the LED is not being driven high, it is driven to a low state. Either of the two circuits shown in Figure 3–8 can be implemented to provide LED signaling and it is left for the board designer to 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.

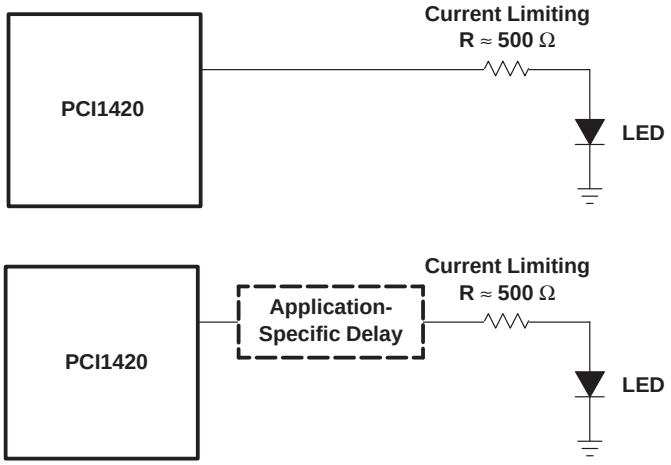


Figure 3–8. Two Sample LED Circuits

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 PC Card-16 Distributed DMA Support

The PCI1420 supports a distributed DMA slave engine for 16-bit PC Card DMA support. The distributed DMA (DDMA) slave register set provides the programmability necessary for the slave DDMA engine. The DDMA register configuration is provided in Table 3–3.

Two socket function dependent PCI configuration header registers that are critical for DDMA are the socket DMA register 0 (see Section 4.35) and the socket DMA register 1 (see Section 4.36). Distributed DMA is enabled through socket DMA register 0 and the contents of this register configure the PC Card-16 terminal ($\overline{\text{SPKR}}$, $\overline{\text{IOIS16}}$, or $\overline{\text{INPACK}}$) which is used for the DMA request signal, $\overline{\text{DREQ}}$. The base address of the DDMA slave registers and the transfer size (bytes or words) are programmed through the socket DMA register 1. Refer to the programming model and register descriptions for details.

Table 3–3. Distributed DMA Registers

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

The DDMA registers contain control and status information consistent with the 8237 DMA controller; however, the register locations are reordered and expanded in some cases. While the DDMA register definitions are identical to those in the 8237 DMA controller of the same name, some register bits defined in the 8237 DMA controller do not apply to distributed DMA in a PCI environment. In such cases, the PCI1420 implements these obsolete register bits as read-only, nonfunctional bits. The reserved registers shown in Table 3–3 are implemented as read-only and return 0s when read. Write transactions to reserved registers have no effect.

The DDMA transfer is prefaced by several configuration steps that are specific to the PC Card and must be completed after the PC Card is inserted and interrogated. These steps include setting the proper $\overline{\text{DREQ}}$ signal assignment, setting the data transfer width, and mapping and enabling the DDMA register set. As discussed above, this is done through socket DMA register 0 and socket DMA register 1. The DMA register set is then programmed similarly to an 8237 controller, and the PCI1420 awaits a $\overline{\text{DREQ}}$ assertion from the PC Card requesting a DMA transfer.

DMA writes transfer data from the PC Card-to-PCI memory addresses. The PCI1420 accepts data 8 or 16 bits at a time, depending on the programmed data width, and then requests access to the PCI bus by asserting its $\overline{\text{REQ}}$ signal. Once the PCI bus is granted in an idle state, the PCI1420 initiates a PCI memory write command to the current memory address and transfers the data in a single data phase. After terminating the PCI cycle, the PCI1420 accepts the next byte(s) from the PC Card until the transfer count expires.

DMA reads transfer data from PCI memory addresses to the PC Card application. Upon the assertion of $\overline{\text{DREQ}}$, the PCI1420 asserts $\overline{\text{REQ}}$ to acquire the PCI bus. Once the bus is granted in an idle state, the PCI1420 initiates a PCI memory read operation to the current memory address and accepts 8 or 16 bits of data, depending on the programmed data width. After terminating the PCI cycle, the data is passed onto the PC Card. After terminating the PC Card cycle, the PCI1420 requests access to the PCI bus again until the transfer count has expired.

The PCI1420 target interface acts normally during this procedure and accepts I/O reads and writes to the DDMA registers. While a DDMA transfer is in progress and the host resets the DMA channel, the PCI1420 asserts TC and ends the PC Card cycle(s). TC is indicated in the DMA status register (see Section 7.5). At the PC Card interface, the PCI1420 supports demand mode transfers. The PCI1420 asserts DACK during the transfer unless $\overline{\text{DREQ}}$ is deasserted before TC. TC is mapped to the $\overline{\text{OE}}$ PC Card terminal for DMA write operations and is mapped to $\overline{\text{WE}}$ PC Card terminal for DMA read operations. The DACK signal is mapped to the PC Card $\overline{\text{REG}}$ signal in all transfers, and the $\overline{\text{DREQ}}$ terminal is routed to one of three options which is programmed through socket DMA register 0.

3.5.10 PC Card-16 PC/PCI DMA

Some chip sets provide a way for legacy I/O devices to do DMA transfers on the PCI bus. In the PC/PCI DMA protocol, the PCI1420 acts as a PCI target device to certain DMA related I/O addresses. The PCI1420 $\overline{\text{PCREQ}}$ and $\overline{\text{PCGNT}}$ signals are provided as a point-to-point connection to a chipset supporting PC/PCI DMA. The $\overline{\text{PCREQ}}$ and $\overline{\text{PCGNT}}$ signals may be routed to the MFUNC2 and MFUNC5 terminals, respectively. See Section 4.30, *Multifunction Routing Register*, for details on configuring the multifunction terminals.

Under the PC/PCI protocol, a PCI DMA slave device (such as the PCI1420) requests a DMA transfer on a particular channel using a serialized protocol on $\overline{\text{PCREQ}}$. The I/O DMA bus master arbitrates for the PCI bus and grants the channel through a serialized protocol on $\overline{\text{PCGNT}}$ when it is ready for the transfer. The I/O cycle and memory cycles are then presented on the PCI bus, which performs the DMA transfers similarly to legacy DMA master devices.

PC/PCI DMA is enabled for each PC Card-16 slot by setting bit 19 (CDREQEN) in the respective system control register (see Section 4.29). On power up this bit is reset and the card PC/PCI DMA is disabled. Bit 3 (CDMA_EN) of the system control register is a global enable for PC/PCI DMA, and is set at power-up and never cleared if the PC/PCI DMA mechanism is implemented. The desired DMA channel for each PC Card-16 slot must be configured through bits 18–16 (CDMACHAN field) in the system control register. The channels are configured as indicated in Table 3–4.

Table 3–4. PC/PCI Channel Assignments

SYSTEM CONTROL REGISTER			DMA CHANNEL	CHANNEL TRANSFER DATA WIDTH
BIT 18	BIT 17	BIT16		
0	0	0	Channel 0	8-bit DMA transfers
0	0	1	Channel 1	8-bit DMA transfers
0	1	0	Channel 2	8-bit DMA transfers
0	1	1	Channel 3	8-bit DMA transfers
1	0	0	Channel 4	Not used
1	0	1	Channel 5	16-bit DMA transfers
1	1	0	Channel 6	16-bit DMA transfers
1	1	1	Channel 7	16-bit DMA transfers

As in distributed DMA, the PC Card terminal mapped to $\overline{\text{DREQ}}$ must be configured through socket DMA register 0 (see Section 4.35). The data transfer width is a function of channel number and the DDMA slave registers are not used. When a $\overline{\text{DREQ}}$ is received from a PC Card and the channel has been granted, the PCI1420 decodes the I/O addresses listed in Table 3–5 and performs actions dependent upon the address.

Table 3–5. I/O Addresses Used for PC/PCI DMA

DMA I/O ADDRESS	DMA CYCLE TYPE	TERMINAL COUNT	PCI CYCLE TYPE
00h	Normal	0	I/O read/write
04h	Normal TC	1	I/O read/write
C0h	Verify	0	I/O read
C4h	Verify TC	1	I/O read

When the PC/PCI DMA is used as a PC Card-16 DMA mechanism, it may not provide the performance levels of DDMA; however, the design of a PCI target implementing PC/PCI DMA is considerably less complex. No bus master state machine is required to support PC/PCI DMA, since the DMA control is centralized in the chipset. This DMA scheme is often referred to as centralized DMA for this reason.

3.5.11 CardBus Socket Registers

The PCI1420 contains all registers for compatibility with the latest *1997 PC Card Standard*. These registers exist as the CardBus socket registers and are listed in Table 3–6.

Table 3–6. CardBus Socket Registers

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

3.6 Serial Bus Interface

The PCI1420 provides a serial bus interface to load subsystem identification and select register defaults through a serial EEPROM and to provide a PC Card power switch interface alternative to P^{2}C. See Section 3.5.2, *P^{2}C Power-Switch Interface (TPS2206/2216)*, for details. The PCI1420 serial bus interface is compatible with various I^{2}C and SMBus components.

3.6.1 Serial Bus Interface Implementation

The PCI1420 defaults to serial bus interface are disabled. To enable the serial interface, a pulldown resistor must be implemented on the LATCH terminal and the appropriate pullup resistor must be implemented on the SDA and SCL signals, that is, the MFUNC1 and MFUNC4 terminals. When the interface is detected, bit 3 (SBDETECT) in the serial bus control and status register (see Section 4.50) is set. The SBDETECT bit is cleared by a write back of 1.

The PCI1420 implements a two-pin serial interface with one clock signal (SCL) and one data signal (SDA). When a pulldown resistor is provided on the LATCH terminal, the SCL signal is mapped to the MFUNC4 terminal and the SDA signal is mapped to the MFUNC1 terminal. The PCI1420 drives SCL at nearly 100 kHz during data transfers, which is the maximum specified frequency for standard mode I^{2}C. The serial EEPROM must be located at address A0h. Figure 3–9 illustrates an example application implementing the two-wire serial bus.

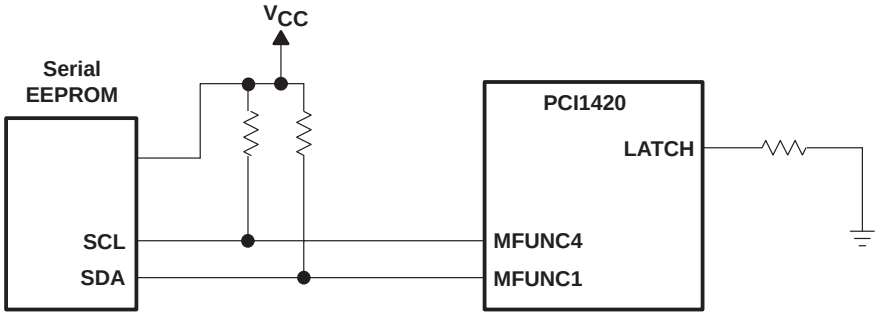


Figure 3–9. Serial EEPROM Application

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

3.6.2 Serial Bus Interface Protocol

The SCL and SDA signals are bidirectional, open-drain signals and require pullup resistors as shown in Figure 3–9. The PCI1420 supports up to 100 Kb/s data transfer rate and is compatible with standard mode I^{2}C using 7-bit addressing.

All data transfers are initiated by the serial bus master. The beginning of a data transfer is indicated by a start condition, which is signalled when the SDA line transitions to low state while SCL is in the high state, as illustrated in Figure 3–10. The end of a requested data transfer is indicated by a stop condition, which is signalled by a low-to-high transition of SDA while SCL is in the high state, as shown in Figure 3–10. Data on SDA must remain stable during the high state of the SCL signal, as changes on the SDA signal during the high state of SCL are interpreted as control signals, that is, a start or a stop condition.

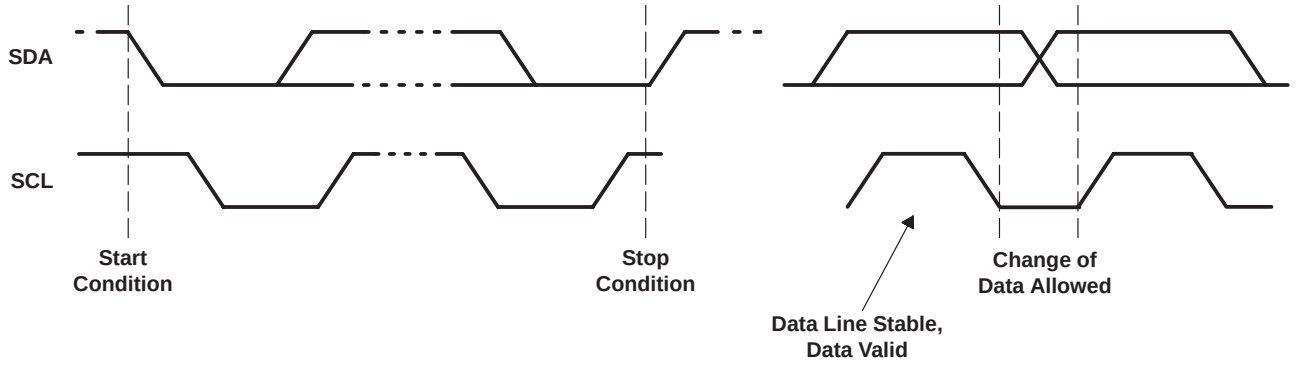


Figure 3–10. Serial Bus Start/Stop Conditions and Bit Transfers

Data is transferred serially in 8-bit bytes. The number of bytes that may be transmitted during a data transfer is unlimited, however, each byte must be completed with an acknowledge bit. An acknowledge (ACK) is indicated by the receiver pulling the SDA signal low so that it remains low during the high state of the SCL signal. Figure 3–11 illustrates the acknowledge protocol.

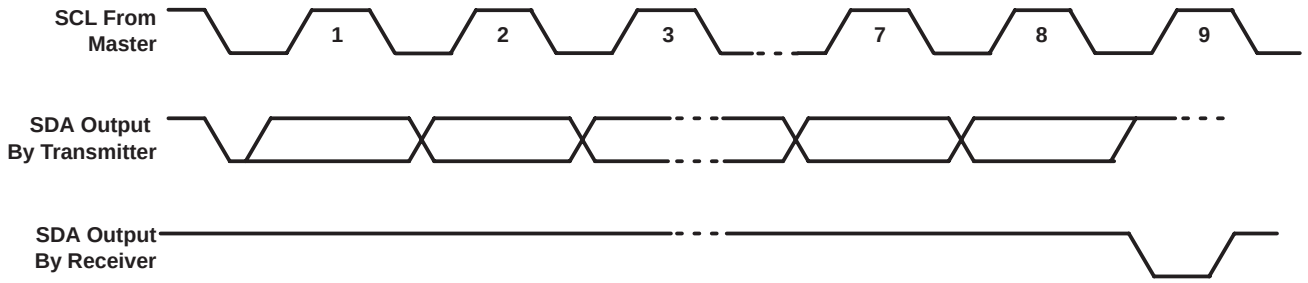


Figure 3–11. Serial Bus Protocol Acknowledge

The PCI1420 is a serial bus master; all other devices connected to the serial bus external to the PCI1420 are slave devices. As the bus master, the PCI1420 drives the SCL clock at nearly 100 kHz during bus cycles and places SCL in a high-impedance state (zero frequency) during idle states.

Typically, the PCI1420 masters byte reads and byte writes under software control. Doubleword reads are performed by the serial EEPROM initialization circuitry upon a PCI reset and may not be generated under software control. See Section 3.6.3, *Serial Bus EEPROM Application*, for details on how the PCI1420 automatically loads the subsystem identification and other register defaults through a serial bus EEPROM.

Figure 3–12 illustrates a byte write. The PCI1420 issues a start condition and sends the 7-bit slave device address and the command bit zero. A 0 in the R/$\overline{W}$ command bit indicates that the data transfer is a write. The slave device acknowledges if it recognizes the address. If there is no acknowledgment received by the PCI1420, then an appropriate status bit is set in the serial bus control and status register (see Section 4.50). The word address byte is then sent by the PCI1420 and another slave acknowledgment is expected. Then the PCI1420 delivers the data byte MSB first and expects a final acknowledgment before issuing the stop condition.

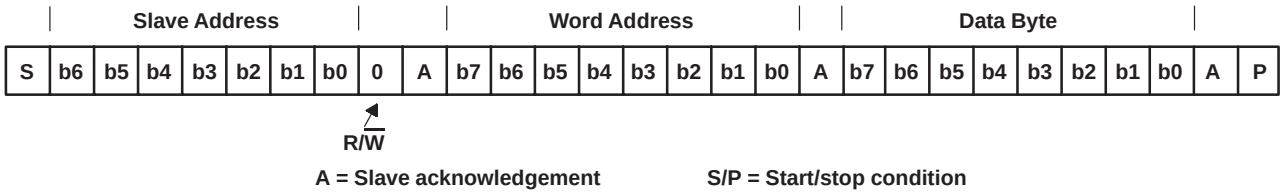


Figure 3–12. Serial Bus Protocol – Byte Write

Figure 3–13 illustrates a byte read. The read protocol is very similar to the write protocol except the R/$\overline{W}$ command bit must be set to 1 to indicate a read-data transfer. In addition, the PCI1420 master must acknowledge reception of the read bytes from the slave transmitter. The slave transmitter drives the SDA signal during read data transfers. The SCL signal remains driven by the PCI1420 master.

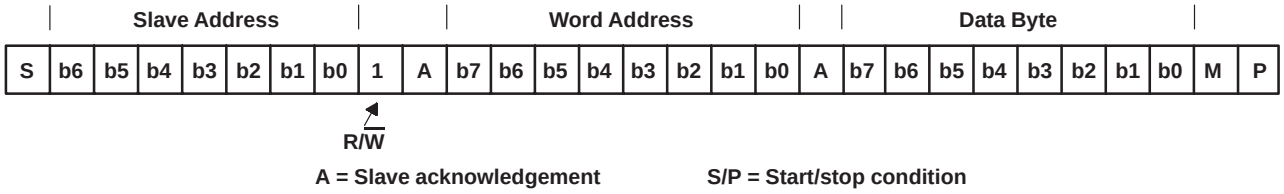


Figure 3–13. Serial Bus Protocol – Byte Read

Figure 3–14 illustrates EEPROM interface doubleword data collection protocol.

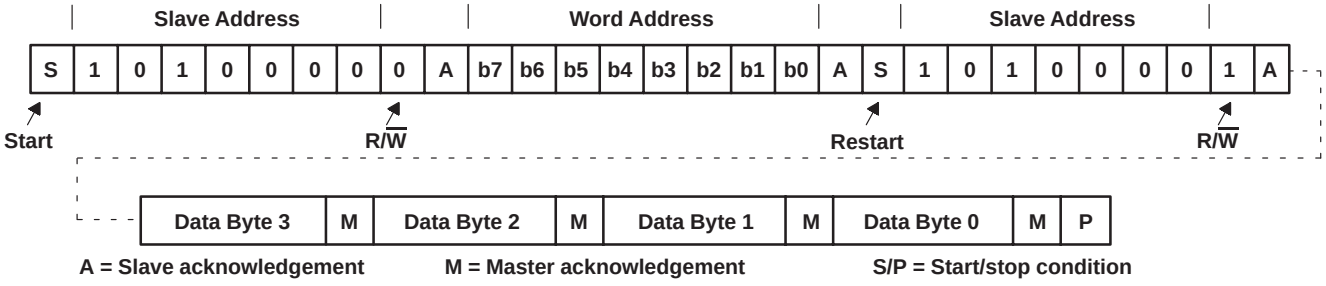


Figure 3–14. EEPROM Interface Doubleword Data Collection

3.6.3 Serial Bus EEPROM Application

When the PCI bus is reset and the serial bus interface is detected, the PCI1420 attempts to read the subsystem identification and other register defaults from a serial EEPROM. The registers and corresponding bits that may be loaded with defaults through the EEPROM are provided in Table 3–7.

Table 3–7. Registers and Bits Loadable Through Serial EEPROM

OFFSET REFERENCE	PCI OFFSET	REGISTER	BITS LOADED FROM EEPROM
01h	40h	Subsystem ID	31–0
02h	80h	System control	31–29, 27, 26, 24, 15, 14, 6–3, 1
03h	8Ch	Multifunction routing	27–0
04h	90h	Retry status, card control, device control, diagnostic	31, 28–24, 22, 19–16, 15, 13, 7, 6

Figure 3–15 details the EEPROM data format. This format must be followed for the PCI1420 to properly load initializations from a serial EEPROM. Any undefined condition results in a terminated load and sets the ROM_ERR bit in the serial bus control and status register (see Section 4.50).

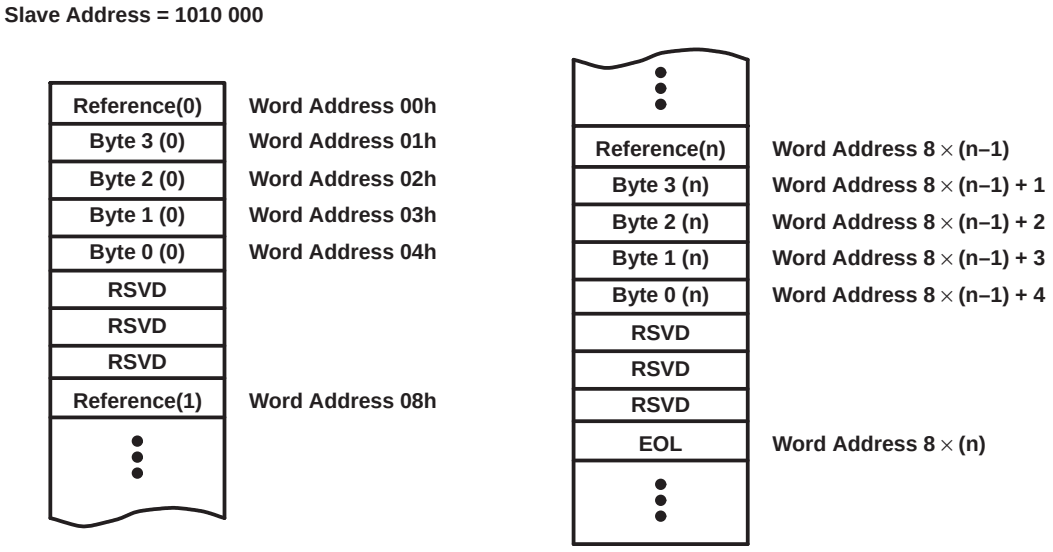


Figure 3–15. EEPROM Data Format

The byte at the EEPROM word address 00h must either contain a valid offset reference, as listed in Table 3–7, or an end-of-list (EOL) indicator. The EOL indicator is a byte value of FFh, and indicates the end of the data to load from the EEPROM. Only doubleword registers are loaded from the EEPROM, and all bit fields must be considered when programming the EEPROM.

The serial EEPROM is addressed at slave address 1010000b by the PCI1420. All hardware address bits for the EEPROM should be tied to the appropriate level to achieve this address. The serial EEPROM chip in the sample application circuit (Figure 3–9) assumes the 1010b high address nibble. The lower three address bits are terminal inputs to the chip, and the sample application shows these terminal inputs tied to GND.

When a valid offset reference is read, four bytes are read from the EEPROM, MSB first, as illustrated in Figure 3–14. The address autoincrements after every byte transfer according to the doubleword read protocol. Note that the word addresses align with the data format illustrated in Figure 3–15. The PCI1420 continues to load data from the serial EEPROM until an end-of-list indicator is read. Three reserved bytes are stuffed to maintain eight-byte data structures.

Note, the eight-byte data structure is important to provide correct addressing per the doubleword read format shown in Figure 3–14. In addition, the reference offsets must be loaded in the EEPROM in sequential order, that is 01h, 02h, 03h, 04h. If the offsets are not sequential, then the registers may be loaded incorrectly.

3.6.4 Serial Bus Power Switch Application

The PCI1420 does not automatically control a serial bus power switch transparently to host software as it does for P^2C power switches. But, the PCI1420 serial bus interface can be used in conjunction with the power status, $\overline{GPE}$, output, and support software to control a serial bus power switch. If a serial bus power switch interface is implemented, then a pulldown resistor must be provided on the PCI1420 CLOCK terminal to reduce power consumption.

The PCI1420 supports two common SMBus data write protocols, write byte and send byte formats. The write byte protocol using a word address of 00h is discussed in Section 3.6.2, *Serial Bus Interface Protocol*. The send byte protocol is shown in Figure 3–16 using a slave address 101001x. Bit 7 (PROT_SEL) in the serial bus control and status register, see Table 4–25, allows the serial bus interface to operate with the send byte protocol. For more information on programming the serial bus interface, see Section 3.6.5, *Accessing Serial Bus Devices Through Software*.

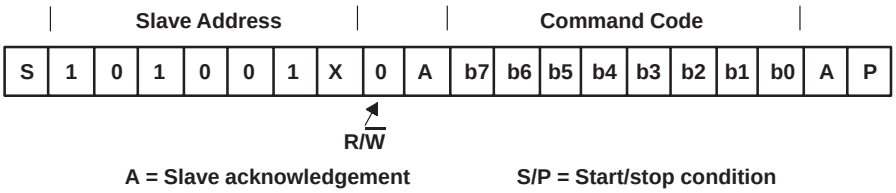


Figure 3–16. Send Byte Protocol

The power switch may support an interrupt mode to indicate over current or other power switch related events. The PCI1420 does not implement logic to respond to these events, but does implement a flexible general-purpose interface to control these events through ACPI and other handlers. See the *Advanced Configuration and Power Interface (ACPI) Specification* for details on implementing the PCI1420 in an ACPI system.

3.6.5 Accessing Serial Bus Devices Through Software

The PCI1420 provides a programming mechanism to control serial bus devices through software. The programming is accomplished through a doubleword of PCI configuration space at offset B0h. Table 3–8 lists the registers used to program a serial bus device through software.

Table 3–8. PCI1420 Registers Used to Program Serial Bus Devices

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

3.7 Programmable Interrupt Subsystem

Interrupts provide a way for I/O devices to let the microprocessor know that they require servicing. The dynamic nature of PC Cards and the abundance of PC Card I/O applications require substantial interrupt support from the PCI1420. The PCI1420 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 PCI1420 is, therefore, backward compatible with existing interrupt control register definitions, and new registers have been defined where required.

The PCI1420 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 PCI1420, PC Card interrupts are classified as either card status change (CSC) or as functional interrupts.

The method by which any type of PCI1420 interrupt is communicated to the host interrupt controller varies from system to system. The PCI1420 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 signalling is provided through the seven multifunction terminals, MFUNC0–MFUNC6.

3.7.1 PC Card Functional and Card Status Change Interrupts

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

Card status change (CSC)-type interrupts are defined as events at the PC Card interface that are detected by the PCI1420 and may warrant notification of host card and socket services software for service. CSC events include both card insertion and removal from PC Card sockets, as well as transitions of certain PC Card signals.

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

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

Table 3–9. Interrupt Mask and Flag Registers

CARD TYPE	EVENT	MASK	FLAG
16-bit memory	Battery conditions (BVD1, BVD2)	ExCA offset 05h/45h/805h bits 1 and 0	ExCA offset 04h/44h/804h bits 1 and 0
	Wait states (READY)	ExCA offset 05h/45h/805h bit 2	ExCA offset 04h/44h/804h bit 2
16-bit I/O	Change in card status ($\overline{\text{STSCHG}}$)	ExCA offset 05h/45h/805h bit 0	ExCA offset 04h/44h/804h bit 0
	Interrupt request ($\overline{\text{IREQ}}$)	Always enabled	PCI configuration offset 91h bit 0
All 16-bit PC Cards	Power cycle complete	ExCA offset 05h/45h/805h bit 3	ExCA offset 04h/44h/804h bit 3
CardBus	Change in card status (CSTSCHG)	Socket mask bit 0	Socket event bit 0
	Interrupt request ($\overline{\text{CINT}}$)	Always enabled	PCI configuration offset 91h bit 0
	Power cycle complete	Socket mask bit 3	Socket event bit 3
	Card insertion or removal	Socket mask bits 2 and 1	Socket event bits 2 and 1

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

Table 3–10. PC Card Interrupt Events and Description

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

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

The *1997 PC Card Standard* describes the power-up sequence that must be followed by the PCI1420 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 PCI1420 interrupt scheme can be used to notify the host system (see Table 3–10), denoted by the power cycle complete event. This interrupt source is considered a PCI1420 internal event because it depends on the completion of applying power to the socket rather than on a signal change at the PC Card interface.

3.7.2 Interrupt Masks and Flags

Host software may individually mask (or disable) most of the potential interrupt sources listed in Table 3–10 by setting the appropriate bits in the PCI1420. By individually masking the interrupt sources listed, software can control those events that cause a PCI1420 interrupt. Host software has some control over the system interrupt the PCI1420 asserts by programming the appropriate routing registers. The PCI1420 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 PCI1420, the interrupt service routine must determine which of the events listed in Table 3–9 caused the interrupt. Internal registers in the PCI1420 provide flags that report the source of an interrupt. By reading these status bits, the interrupt service routine can determine the action to be taken.

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

Notice that there is not a mask bit to stop the PCI1420 from passing PC Card functional interrupts through to the appropriate interrupt scheme. These interrupts are not valid until the card is properly powered, and there should never be a card interrupt that does not require service after proper initialization.

Table 3–9 lists the various methods of clearing the interrupt flag bits. The flag bits in the ExCA registers (16-bit PC Card-related interrupt flags) can be cleared using two different methods. One method is an explicit write of 1 to the flag bit to clear and the other is by reading the flag bit register. The selection of flag bit clearing is made by bit 2 (IFCMODE) in the ExCA global control register (see Section 5.22), located at ExCA offset 1Eh/5Eh/81Eh, and defaults to the *flag cleared on read* method.

The CardBus-related interrupt flags can be cleared by an explicit write of 1 to the interrupt flag in the socket event register (see Section 6.1). Although some of the functionality is shared between the CardBus registers and the ExCA registers, software should not program the chip through both register sets when a CardBus card is functioning.

3.7.3 Using Parallel IRQ Interrupts

The seven multifunction terminals, MFUNC6–MFUNC0, implemented in the PCI1420 may 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 (see Section 4.33), located at PCI offset 92h, to select the parallel IRQ signaling scheme. See Section 4.30, *Multifunction Routing Register*, for details on configuring the multifunction terminals.

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

As an example, suppose the six IRQs used by legacy PC Card applications are IRQ3, IRQ4, IRQ5, IRQ10, IRQ11, and IRQ15. The multifunction routing register must be programmed to a value of 0x0FBA5432. This value routes the MFUNC0 terminal to $\overline{\text{INTA}}$ signaling and routes the remaining terminals as illustrated in Figure 3–17. Not shown is that $\overline{\text{INTA}}$ must also be routed to the programmable interrupt controller (PIC), or to some circuitry that provides parallel PCI interrupts to the host.

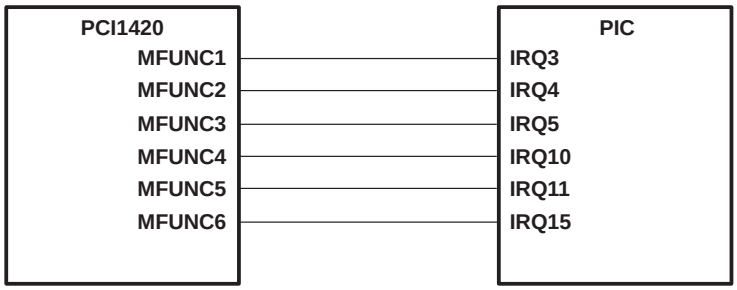


Figure 3–17. IRQ Implementation

Power-on software is responsible for programming the multifunction routing register to reflect the IRQ configuration of a system implementing the PCI1420. The multifunction routing register is shared between the two PCI1420 functions, and only one write to function 0 or 1 is necessary to configure the MFUNC6–MFUNC0 signals. Writing to only function 0 is recommended. See Section 4.30, *Multifunction Routing Register*, for details on configuring the multifunction terminals.

The parallel ISA type IRQ signaling from the MFUNC6–MFUNC0 terminals is compatible with those input directly into 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 PCI1420 makes available.

3.7.4 Using Parallel PCI Interrupts

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

The INTRTIE bit affects the read-only value provided through accesses to the interrupt pin register (see Section 4.24). When INTRTIE bit is set, both functions return a value of 0x01 on reads from the interrupt pin register for both parallel and serial PCI interrupts. Table 3–11 summarizes the interrupt signalling modes.

Table 3–11. Interrupt Pin Register Cross Reference

INTERRUPT SIGNALING MODE	INTRTIE BIT	INTPIN FUNCTION 0	INTPIN FUNCTION 1
Parallel PCI interrupts only	0	0x01 ($\overline{\text{INTA}}$)	0x02 ($\overline{\text{INTB}}$)
Parallel IRQ and parallel PCI interrupts	0	0x01 ($\overline{\text{INTA}}$)	0x02 ($\overline{\text{INTB}}$)
IRQ serialized (IRQSER) and parallel PCI interrupts	0	0x01 ($\overline{\text{INTA}}$)	0x02 ($\overline{\text{INTB}}$)
IRQ and PCI serialized (IRQSER) interrupts (default)	0	0x01 ($\overline{\text{INTA}}$)	0x02 ($\overline{\text{INTB}}$)
Parallel PCI interrupts only	1	0x01 ($\overline{\text{INTA}}$)	0x01 ($\overline{\text{INTA}}$)
Parallel IRQ and parallel PCI interrupts	1	0x01 ($\overline{\text{INTA}}$)	0x01 ($\overline{\text{INTA}}$)
IRQ serialized (IRQSER) and parallel PCI interrupts	1	0x01 ($\overline{\text{INTA}}$)	0x01 ($\overline{\text{INTA}}$)
IRQ and PCI serialized (IRQSER) interrupts	1	0x01 ($\overline{\text{INTA}}$)	0x01 ($\overline{\text{INTA}}$)

3.7.5 Using Serialized IRQSER Interrupts

The serialized interrupt protocol implemented in the PCI1420 uses a single terminal to communicate all interrupt status information to the host controller. The protocol defines a serial packet consisting of a start cycle, multiple interrupt indication cycles, and a stop cycle. All data in the packet is synchronous with the PCI clock. The packet data describes 16 parallel ISA IRQ signals and the optional 4 PCI interrupts $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, $\overline{\text{INTC}}$, and $\overline{\text{INTD}}$. For details on the IRQSER protocol refer to the document *Serialized IRQ Support for PCI Systems*.

3.7.6 SMI Support in the PCI1420

The PCI1420 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 PCI1420, when enabled, after a write cycle to either the socket control register (see Section 6.5) of the CardBus register set or the ExCA power control register (see Section 5.3) causes a power cycle change sequence sent on the power switch interface.

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

Table 3–12. SMI Control

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

If CSC SMI interrupts are selected, then the SMI interrupt is sent as the CSC on a per socket basis. The CSC interrupt can be either level or edge mode, depending upon the CSCMODE bit in the ExCA global control register (see Section 5.22).

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 (see Section 4.30).

3.8 Power Management Overview

In addition to the low-power CMOS technology process used for the PCI1420, various features are designed into the device to allow implementation of popular power-saving techniques. These features and techniques are discussed in this section.

3.8.1 Clock Run Protocol

The PCI $\overline{\text{CLKRUN}}$ feature is the primary method of power management on the PCI interface of the PCI1420. $\overline{\text{CLKRUN}}$ signalling is provided through the MFUNC6 terminal. Since some chip sets do not implement $\overline{\text{CLKRUN}}$, this is not always available to the system designer, and alternate power savings features are provided. For details on the $\overline{\text{CLKRUN}}$ protocol see the *PCI Mobile Design Guide*.

The PCI1420 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 (see Section 4.29) is set.
- The PC Card-16 resource manager is busy.
- The PCI1420 CardBus master state machine is busy. A cycle may be in progress on CardBus.
- The PCI1420 master is busy. There may be posted data from CardBus to PCI in the PCI1420.
- Interrupts are pending.
- The CardBus CCLK for either socket has not been stopped by the PCI1420 $\overline{\text{CCLKRUN}}$ manager.

The PCI1420 restarts the PCI clock using the CLKRUN protocol under any of the following conditions:

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

3.8.2 CardBus PC Card Power Management

The PCI1420 implements its own card power management engine that can turn off the CCLK to a socket when there is no activity to the CardBus PC Card. The PCI clock-run protocol is followed on the CardBus $\overline{\text{CCLKRUN}}$ interface to control this clock management.

3.8.3 16-Bit PC Card Power Management

The COE (bit 7, ExCA power control register) and PWRDWN (bit 0, ExCA global control register) 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 will reset the PC Card when used, and the PWRDWN bit will not. Furthermore, the PWRDWN bit is an automatic COE, that is, the PWRDWN performs the COE function when there is no card activity.

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

3.8.4 Suspend Mode

The $\overline{\text{SUSPEND}}$ signal, provided for backward compatibility, gates the $\overline{\text{PRST}}$ (PCI reset) signal and the $\overline{\text{GRST}}$ (global reset) signal from the PCI1420. Besides gating $\overline{\text{PRST}}$ and $\overline{\text{GRST}}$, $\overline{\text{SUSPEND}}$ also gates PCLK inside the PCI1420 in order to minimize power consumption.

Gating PCLK does not create any issues with respect to the power switch interface in the PCI1420. This is because the PCI1420 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 PCI1420:

- Use an external clock to the PCI1420 CLOCK pin
- Use the internal oscillator

It should also be noted that asynchronous signals, such as card status change interrupts and RI_OUT, can be passed to the host system without a PCI clock. However, if card status change interrupts are routed over the serial interrupt stream, then the PCI clock will have to be restarted in order to pass the interrupt, because neither the internal oscillator nor an external clock is routed to the serial interrupt state machine. Figure 3–18 is a functional implementation diagram.

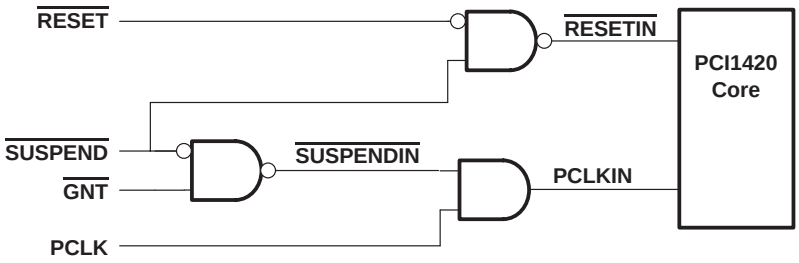


Figure 3–18. Suspend Functional Implementation

Figure 3–19 is a signal diagram of the suspend function.

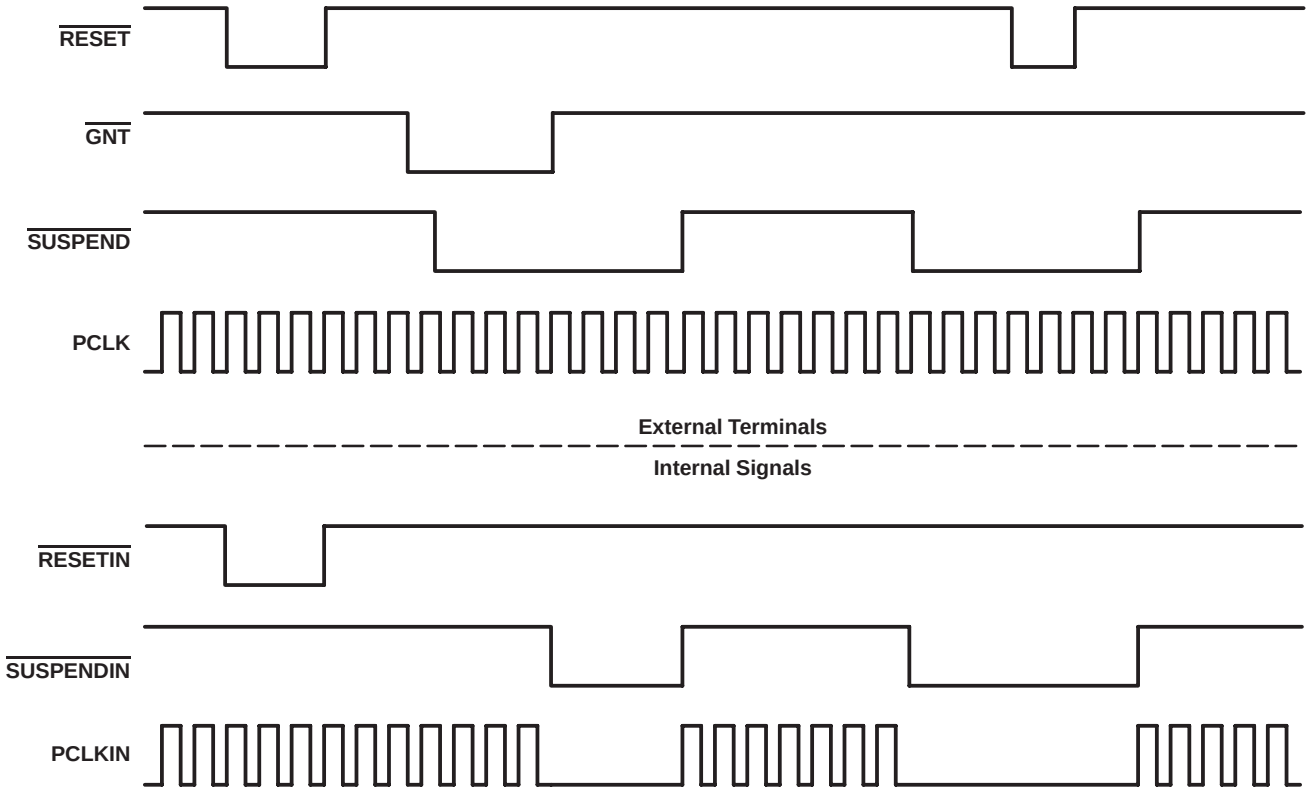


Figure 3–19. Signal Diagram of Suspend Function

3.8.5 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 PCI1420 by software. Asserting the SUSPEND signal places the controller's PCI outputs 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 PCI1420 when SUSPEND is asserted because the outputs are in a high impedance state.

The GPIOs, MFUNC signals, and RI_OUT signals are all active during SUSPEND, unless they are disabled in the appropriate PCI1420 registers.

3.8.6 Ring Indicate

The $\overline{\text{RI\_OUT}}$ output is an important feature in power management, allowing a system to go into a suspended mode and wake up on modem rings and other card events. TI designed flexibility permits this signal to fit wide platform requirements. $\overline{\text{RI\_OUT}}$ on the PCI1420 can be asserted under any of the following conditions:

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

Figure 3–20 shows various enable bits for the PCI1420 $\overline{\text{RI\_OUT}}$ function; however, it does not show the masking of CSC events. See Table 3–9 for a detailed description of CSC interrupt masks and flags.

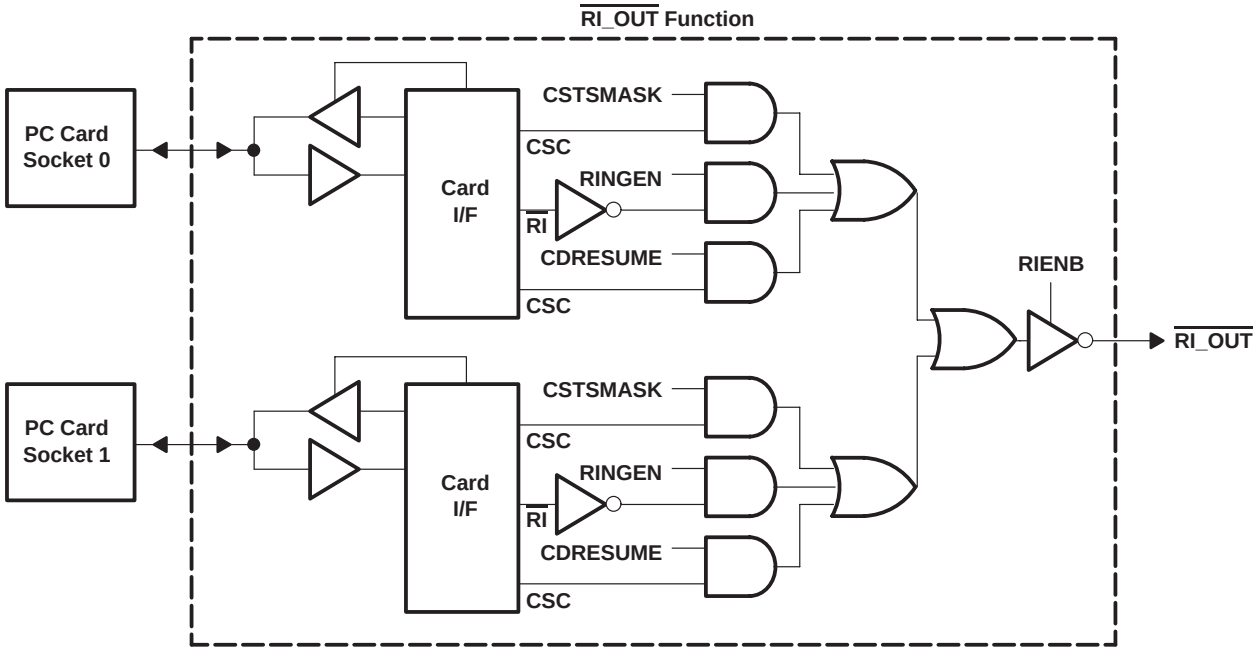


Figure 3–20. $\overline{\text{RI\_OUT}}$ Functional Diagram

$\overline{\text{RI}}$ from the 16-bit PC Card interface is masked by bit 7 (RINGEN) in the ExCA interrupt and general control register (see Section 5.4). This is programmed on a per-socket basis and is only applicable when a 16-bit card is powered in the socket.

The CBWAKE signaling to $\overline{\text{RI\_OUT}}$ is enabled through the same mask as the CSC event for CSTSCHG. The mask bit (bit 0, CSTSMASK) is programmed through the socket mask register (see Section 6.2) in the CardBus socket registers.

3.8.7 PCI Power Management

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

The four power management states of PCI functions are:

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

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

For the operating system (OS) to power manage the device power states on the PCI bus, the PCI function should support four power management operations. These 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 (see Section 4.5).

The capabilities pointer provides access to the first item in the linked list of capabilities. For the PCI1420, a CardBus bridge with PCI configuration space header type 2, the capabilities pointer is mapped to an offset of 14h. The first byte of each capability register block is required to be a unique ID of that capability. PCI power management has been assigned an ID of 01h. The next byte is a pointer to the next pointer item in the list of capabilities. If there are no more items in the list, then the next item pointer should be set to 0. The registers following the next item pointer are specific to the function's capability. The PCI power management capability implements the register block outlined in Table 3–13.

Table 3–13. Power Management Registers

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

The power management capabilities register (see Section 4.39) is a static read-only register that provides information on the capabilities of the function related to power management. The PMCSR register (see Section 4.40) enables control of power management states and enables/monitors power management events. The data register is an optional register that can provide dynamic data.

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

3.8.8 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 $\overline{PME}$ 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 when transitioning from D3 to D0. Some method to preserve wake-up context must be implemented so that the reset does not clear the $\overline{PME}$ context registers.
- Power source in $D3_{cold}$ if wake-up support is required from this state.

The Texas Instruments PCI1420 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 PCI1420 in its default state and requires BIOS to configure the device before becoming fully functional.
  - PCI reset ($\overline{\text{PRST}}$) now has dual functionality based on whether $\overline{\text{PME}}$ is enabled or not. If $\overline{\text{PME}}$ is enabled, then $\overline{\text{PME}}$ context is preserved. If $\overline{\text{PME}}$ is not enabled, then $\overline{\text{PRST}}$ acts the same as a normal PCI reset. Please see the master list of $\overline{\text{PME}}$ context bits in Section 3.8.10.
- 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 PCI1420 V_{CC} pins. Consult the *PCI14xx Implemenation Guide for D3 Wake-Up* or the *PCI Power Management Interface Specification for PCI to CardBus Bridges* for further information.

3.8.9 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 PCI1420 offers a generic interface that is compliant with ACPI design rules.

Two doublewords of general-purpose ACPI programming bits reside in PCI1420 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 general-purpose event status (see Section 4.43) and general-purpose event enable (see Section 4.44) registers. The status and enable bits are implemented as defined by ACPI and illustrated in Figure 3–21.

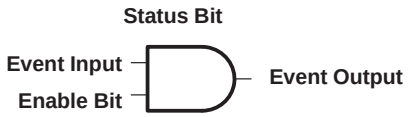


Figure 3–21. Block Diagram of a Status/Enable Cell

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

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

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

If the $\overline{\text{PME}}$ enable bit (PCI offset A4h, bit 8) is asserted, then the assertion of $\overline{\text{PRST}}$ will not clear the following $\overline{\text{PME}}$ context bits. If the $\overline{\text{PME}}$ enable bit is not asserted, then the $\overline{\text{PME}}$ context bits are cleared with $\overline{\text{PRST}}$. The $\overline{\text{PME}}$ context bits are:

- Bridge control register (PCI offset 3Eh): bit 22
- System control register (PCI offset 80h): bits 10, 9, 8
- Power management control/status register (PCI offset A4h): bits 15, 8
- ExCA identification and revision register (ExCA offset 800h): bits 30, 29, 23, 21, 20, 19, 17, 16
- ExCA card status change register (ExCA offset 804h): bits 11–8, 3–0
- CardBus socket event register (CardBus offset 00h): bits 3–0
- CardBus socket mask register (CardBus offset 04h): bits 3–0
- CardBus socket present state register (CardBus offset 08h): bits 13–7, 5–1
- CardBus socket control register (CardBus offset 10h): bits 6, 5, 4, 2, 1, 0

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

- Status register (PCI offset 06h): bits 15–11, 8
- Secondary status register (PCI offset 16h): bits 15–11, 8
- Interrupt pin register (PCI offset 3Dh): bits 1,0 (function 1 only)
- Subsystem vendor ID register (PCI offset 40h): bits 15–0
- Subsystem ID register (PCI offset 42h): bits 15–0
- PC Card 16-bit legacy mode base address register (PCI offset 44h): bits 31–1
- System control register (PCI offset 80h): bits 31–29, 27–13, 11, 6–0
- Multifunction routing register (PCI offset 8Ch): bits 27–0
- Retry status register (PCI offset 90h): bits 7–5, 3, 1
- Card control register (PCI offset 91h): bits 7–5, 2–0
- Device control register (PCI offset 92h): bits 7–5, 3–0
- Diagnostic register (PCI offset 93h): bits 7–0
- Socket DMA register 0 (PCI offset 94h): bits 1–0
- Socket DMA register 1 (PCI offset 98h): bits 15–4, 2–0
- Power management capabilities register (PCI offset A2h): bit 15
- General-purpose event status register (PCI offset A8h): bits 15, 14
- General-purpose event enable register (PCI offset AAh): bits 15, 14, 11, 8, 4–0
- General-purpose output (PCI offset AEh): bits 4–0
- Serial bus data (PCI offset B0h): bits 7–0
- Serial bus index (PCI offset B1h): bits 7–0
- Serial bus slave address register (PCI offset B2h): bits 7–0
- Serial bus control and status register (PCI offset B3h): bits 7, 5–0
- ExCA identification and revision register (ExCA offset 00h): bits 7–0
- ExCA global control register (ExCA offset 1Eh): bits 2–0
- Socket present state register (CardBus offset 08h): bit 29
- Socket power management (CardBus offset 20h): bits 25, 24

4 PC Card Controller Programming Model

This section describes the PCI1420 PCI configuration registers that make up the 256-byte PCI configuration header for each PCI1420 function. As noted, some bits are global in nature and are accessed only through function 0.

4.1 PCI Configuration Registers (Functions 0 and 1)

The PCI1420 is a multifunction PCI device, and the PC Card controller is integrated as PCI functions 0 and 1. The configuration header is compliant with the *PCI Local Bus Specification* as a CardBus bridge header and is PC 99 compliant as well. Table 4–1 shows the PCI configuration header, which includes both the predefined portion of the configuration space and the user-definable registers.

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

REGISTER NAME				OFFSET
Device ID		Vendor ID		00h
Status		Command		04h
Class code			Revision ID	08h
BIST	Header type	Latency timer	Cache line size	0Ch
CardBus socket/ExCA base address				10h
Secondary status		Reserved	Capability pointer	14h
CardBus latency timer	Subordinate bus number	CardBus bus number	PCI bus number	18h
CardBus Memory base register 0				1Ch
CardBus Memory limit register 0				20h
CardBus Memory base register 1				24h
CardBus Memory limit register 1				28h
CardBus I/O base register 0				2Ch
CardBus I/O limit register 0				30h
CardBus I/O base register 1				34h
CardBus I/O limit register 1				38h
Bridge control		Interrupt pin	Interrupt line	3Ch
Subsystem ID		Subsystem vendor ID		40h
PC Card 16-bit I/F legacy-mode base address				44h
Reserved				48h–7Ch
System control				80h
Reserved				84h–88h
Multifunction routing				8Ch
Diagnostic	Device control	Card control	Retry status	90h
Socket DMA register 0				94h
Socket DMA register 1				98h
Reserved				9Ch
Power management capabilities		Next-item pointer	Capability ID	A0h
PM data	PMCSR bridge support extensions	Power management control/status		A4h
General-purpose event enable		General-purpose event status		A8h
General-purpose output		General-purpose input		ACh
Serial bus control/status	Serial bus slave address	Serial bus index	Serial bus data	B0h
Reserved				B4h–FCh

4.2 Vendor ID Register

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

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

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

4.3 Device ID Register

This 16-bit register contains a value assigned to the PCI1420 by TI. The device identification for the PCI1420 is AC51h.

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

Register: **Device ID**
Type: Read-only
Offset: 02h (functions 0, 1)
Default: AC51h

4.4 Command Register

The command register provides control over the PCI1420 interface to the PCI bus. All bit functions adhere to the definitions in *PCI Local Bus Specification*. None of the bit functions in this register are shared between the two PCI1420 PCI functions. Two command registers exist in the PCI1420, one for each function. Software must manipulate the two PCI1420 functions as separate entities when enabling functionality through the command register. The SERR_EN and PERR_EN enable bits in this register are internally wired-OR between the two functions, and these control bits appear separately according to their software function. See Table 4–2 for the 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	R/W	R	R/W	R/W	R	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Table 4–2. Command Register

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 PCI1420 does not generate fast back-to-back transactions; therefore, bit 9 returns 0 when read.
8	SERR_EN	R/W	System Error ($\overline{\text{SERR}}$) enable. Bit 8 controls the enable for the $\overline{\text{SERR}}$ driver on the PCI interface. $\overline{\text{SERR}}$ can be asserted after detecting an address parity error on the PCI bus. Both bits 8 and 6 must be set for the PCI1420 to report address parity errors. 0 = Disable $\overline{\text{SERR}}$ output driver (default) 1 = Enable $\overline{\text{SERR}}$ output driver
7	STEP_EN	R	Address/data stepping control. The PCI1420 does not support address/data stepping; therefore, bit 7 is hardwired to 0.
6	PERR_EN	R/W	Parity error response enable. Bit 6 controls the PCI1420's response to parity errors through $\overline{\text{PERR}}$. Data parity errors are indicated by asserting $\overline{\text{PERR}}$, whereas address parity errors are indicated by asserting $\overline{\text{SERR}}$. 0 = PCI1420 ignores detected parity error (default) 1 = PCI1420 responds to detected parity errors
5	VGA_EN	R/W	VGA palette snoop. Bit 5 controls how PCI devices handle accesses to video graphics array (VGA) palette registers.
4	MWI_EN	R	Memory write and invalidate enable. Bit 4 controls whether a PCI initiator device can generate memory write and Invalidate commands. The PCI1420 controller does not support memory write and invalidate commands, it uses memory write commands instead; therefore, this bit is hardwired to 0.
3	SPECIAL	R	Special cycles. Bit 3 controls whether or not a PCI device ignores PCI special cycles. The PCI1420 does not respond to special cycle operations; therefore, this bit is hardwired to 0.
2	MAST_EN	R/W	Bus master control. Bit 2 controls whether or not the PCI1420 can act as a PCI bus initiator (master). The PCI1420 can take control of the PCI bus only when this bit is set. 0 = Disables the PCI1420's ability to generate PCI bus accesses (default) 1 = Enables the PCI1420's ability to generate PCI bus accesses
1	MEM_EN	R/W	Memory space enable. Bit 1 controls whether or not the PCI1420 can claim cycles in PCI memory space. 0 = Disables the PCI1420's response to memory space accesses (default) 1 = Enables the PCI1420's response to memory space accesses
0	IO_EN	R/W	I/O space control. Bit 0 controls whether or not the PCI1420 can claim cycles in PCI I/O space. 0 = Disables the PCI1420 from responding to I/O space accesses (default) 1 = Enables the PCI1420 to respond to I/O space accesses

4.5 Status Register

The status register provides device information to the host system. Bits in this register may be read normally. A bit in the status register is reset when a 1 is written to that bit location; a 0 written to a bit location has no effect. All bit functions adhere to the definitions in the *PCI Local Bus Specification*. PCI bus status is shown through each function. See Table 4–3 for the 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	R/C	R/C	R/C	R/C	R/C	R	R	R/C	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0

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

Table 4–3. Status Register

BIT	SIGNAL	TYPE	FUNCTION
15	PAR_ERR	R/C	Detected parity error. Bit 15 is set when a parity error is detected (either address or data).
14	SYS_ERR	R/C	Signaled system error. Bit 14 is set when $\overline{\text{SERR}}$ is enabled and the PCI1420 signals a system error to the host.
13	MABORT	R/C	Received master abort. Bit 13 is set when a cycle initiated by the PCI1420 on the PCI bus has been terminated by a master abort.
12	TABT_REC	R/C	Received target abort. Bit 12 is set when a cycle initiated by the PCI1420 on the PCI bus was terminated by a target abort.
11	TABT_SIG	R/C	Signaled target abort. Bit 11 is set by the PCI1420 when it terminates a transaction on the PCI bus with a target abort.
10–9	PCI_SPEED	R	$\overline{\text{DEVSEL}}$ timing. These bits encode the timing of $\overline{\text{DEVSEL}}$ and are hardwired 01b, indicating that the PCI1420 asserts PCI_SPEED at a medium speed on nonconfiguration cycle accesses.
8	DATAPAR	R/C	Data parity error detected. 0 = The conditions for setting bit 8 have not been met. 1 = A data parity error occurred, and the following conditions were met: a. $\overline{\text{PERR}}$ was asserted by any PCI device including the PCI1420. b. The PCI1420 was the bus master during the data parity error. c. The parity error response bit is set in the command.
7	FBB_CAP	R	Fast back-to-back capable. The PCI1420 cannot accept fast back-to-back transactions; therefore, bit 7 is hardwired to 0.
6	UDF	R	User-definable feature support. The PCI1420 does not support the user-definable features; therefore, bit 6 is hardwired to 0.
5	66MHZ	R	66-MHz capable. The PCI1420 operates at a maximum PCLK frequency of 33 MHz; therefore, bit 5 is hardwired to 0.
4	CAPLIST	R	Capabilities list. Bit 4 returns 1 when read. This bit indicates that capabilities in addition to standard PCI capabilities are implemented. The linked list of PCI power management capabilities is implemented in this function.
3–0	RSVD	R	Reserved. Bits 3–0 return 0s when read.

4.6 Revision ID Register

The revision ID register indicates the silicon revision of the PCI1420.

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

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

4.7 PCI Class Code Register

The class code register recognizes the PCI1420 functions 0 and 1 as a bridge device (06h) and CardBus bridge device (07h) with a 00h programming interface.

Bit	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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**
Type: Read-only
Offset: 09h (functions 0, 1)
Default: 060700h

4.8 Cache Line Size Register

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

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

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

4.9 Latency Timer Register

The latency timer register specifies the latency timer for the PCI1420 in units of PCI clock cycles. When the PCI1420 is a PCI bus initiator and asserts $\overline{\text{FRAME}}$, the latency timer begins counting from zero. If the latency timer expires before the PCI1420 transaction has terminated, then the PCI1420 terminates the transaction when its $\overline{\text{GNT}}$ is deasserted. This register is separate for each of the two PCI1420 functions. This allows platforms to prioritize the two PCI1420 functions' use of the PCI bus.

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

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

4.10 Header Type Register

This register returns 82h when read, indicating that the PCI1420 functions 0 and 1 configuration spaces adhere to the CardBus bridge PCI header. The CardBus bridge PCI header ranges from PCI register 0 to 7Fh, and 80h–FFh is user-definable extension registers.

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

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

4.11 BIST Register

Because the PCI1420 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**
Type: Read-only
Offset: 0Fh (functions 0, 1)
Default: 00h

4.12 CardBus Socket/ExCA Base-Address Register

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

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

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

4.13 Capability Pointer Register

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

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

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

4.14 Secondary Status Register

The secondary status register is compatible with the PCI-to-PCI bridge secondary status register and indicates CardBus-related device information to the host system. This register is very similar to the PCI status register (offset 06h); status bits are cleared by writing a 1.

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

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

Table 4–4. Secondary Status Register

BIT	SIGNAL	TYPE	FUNCTION
15	CBPARITY	R/C	Detected parity error. Bit 15 is set when a CardBus parity error is detected (either address or data).
14	CBSERR	R/C	Signaled system error. Bit 14 is set when $\overline{\text{CSERR}}$ is signaled by a CardBus card. The PCI1420 does not assert $\overline{\text{CSERR}}$.
13	CBMABORT	R/C	Received master abort. Bit 13 is set when a cycle initiated by the PCI1420 on the CardBus bus has been terminated by a master abort.
12	REC_CBTA	R/C	Received target abort. Bit 12 is set when a cycle initiated by the PCI1420 on the CardBus bus is terminated by a target abort.
11	SIG_CBTA	R/C	Signaled target abort. Bit 11 is set by the PCI1420 when it terminates a transaction on the CardBus bus with a target abort.
10–9	CB_SPEED	R	CDEVSEL timing. These bits encode the timing of $\overline{\text{CDEVSEL}}$ and are hardwired 01b, indicating that the PCI1420 asserts CB_SPEED at a medium speed.
8	CB_DPAR	R/C	CardBus data parity error detected. 0 = The conditions for setting bit 8 have not been met. 1 = A data parity error occurred and the following conditions were met: a. $\overline{\text{CPERR}}$ was asserted on the CardBus interface. b. The PCI1420 was the bus master during the data parity error. c. The parity error response bit is set in the bridge control.
7	CBFBB_CAP	R	Fast back-to-back capable. The PCI1420 cannot accept fast back-to-back transactions; therefore, bit 7 is hardwired to 0.
6	CB_UDF	R	User-definable feature support. The PCI1420 does not support the user-definable features; therefore, bit 6 is hardwired to 0.
5	CB66MHZ	R	66-MHz capable. The PCI1420 CardBus interface operates at a maximum CCLK frequency of 33 MHz; therefore, bit 5 is hardwired to 0.
4–0	RSVD	R	Reserved. Bits 4–0 return 0s when read.

4.15 PCI Bus Number Register

This register is programmed by the host system to indicate the bus number of the PCI bus to which the PCI1420 is connected. The PCI1420 uses this register in conjunction with the CardBus bus number and subordinate bus number registers to determine when to forward PCI configuration cycles to its secondary buses.

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

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

4.16 CardBus Bus Number Register

This register is programmed by the host system to indicate the bus number of the CardBus bus to which the PCI1420 is connected. The PCI1420 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 PCI1420 controller function.

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

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

4.17 Subordinate Bus Number Register

This register is programmed by the host system to indicate the highest-numbered bus below the CardBus bus. The PCI1420 uses this register in conjunction with the PCI bus number and CardBus bus number registers to determine when to forward PCI configuration cycles to its secondary buses. This register is separate for each CardBus controller function.

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

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

4.18 CardBus Latency Timer Register

This register is programmed by the host system to specify the latency timer for the PCI1420 CardBus interface in units of CCLK cycles. When the PCI1420 is a CardBus initiator and asserts $\overline{\text{CFRAME}}$, the CardBus latency timer begins counting. If the latency timer expires before the PCI1420 transaction has terminated, then the PCI1420 terminates the transaction at the end of the next data phase. A recommended minimum value for this register is 40h, which allows most transactions to be completed.

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

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

4.19 Memory Base Registers 0, 1

The memory base registers indicate the lower address of a PCI memory address range. These registers are used by the PCI1420 to determine when to forward a memory transaction to the CardBus bus and when to forward a CardBus cycle to PCI. Bits 31–12 of these registers are read/write and allow the memory base to be located anywhere in the 32-bit PCI memory space on 4-Kbyte boundaries. Bits 11–0 are read-only and always return 0s. Write transactions to these bits have no effect. Bits 8 and 9 of the bridge control register specify whether memory windows 0 and 1 are prefetchable or nonprefetchable. The memory base register or the memory limit register must be nonzero for the PCI1420 to claim any memory transactions through CardBus memory windows (that is, these windows are not enabled by default to pass the first 4K bytes of memory to CardBus).

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

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

4.20 Memory Limit Registers 0, 1

The memory limit registers indicate the upper address of a PCI memory address range. These registers are used by the PCI1420 to determine when to forward a memory transaction to the CardBus bus and when to forward a CardBus cycle to PCI. Bits 31–12 of these registers are read/write and allow the memory base to be located anywhere in the 32-bit PCI memory space on 4-Kbyte boundaries. Bits 11–0 are read-only and always return 0s. Write transactions to these bits have no effect. Bits 8 and 9 of the bridge control register specify whether memory windows 0 and 1 are prefetchable or nonprefetchable. The memory base register or the memory limit register must be nonzero for the PCI1420 to claim any memory transactions through CardBus memory windows (that is, these windows are not enabled by default to pass the first 4K bytes of memory to CardBus).

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

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

4.21 I/O Base Registers 0, 1

The I/O base registers indicate the lower address of a PCI I/O address range. These registers are used by the PCI1420 to determine when to forward an I/O transaction to the CardBus bus and when to forward a CardBus cycle to the PCI bus. The lower 16 bits of this register locate the bottom of the I/O window within a 64-Kbyte page, and the upper 16 bits (31–16) are a page register which locates this 64-Kbyte page in 32-bit PCI I/O address space. Bits 31–2 are read/write. Bits 1 and 0 are read-only and always return 0s, forcing I/O windows to be aligned on a natural doubleword boundary.

NOTE: Either the I/O base 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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	I/O base registers 0, 1															
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

4.22 I/O Limit Registers 0, 1

The I/O limit registers indicate the upper address of a PCI I/O address range. These registers are used by the PCI1420 to determine when to forward an I/O transaction to the CardBus bus and when to forward a CardBus cycle to PCI. The lower 16 bits of this register locate the top of the I/O window within a 64-Kbyte page, and the upper 16 bits are a page register that locates this 64-Kbyte page in 32-bit PCI I/O address space. Bits 15–2 are read/write and allow the I/O limit address to be located anywhere in the 64-Kbyte page (indicated by bits 31–16 of the appropriate I/O base) on doubleword boundaries.

Bits 31–16 are read-only and always return 0s when read. The page is set in the I/O base register. Bits 1 and 0 are read-only and always return 0s, forcing I/O windows to be aligned on a natural doubleword boundary. Write transactions to read-only bits have no effect. The PCI1420 assumes that the lower 2 bits of the limit address are 1s.

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

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

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

4.23 Interrupt Line Register

The interrupt line register communicates interrupt line routing information. Each PCI1420 function has an interrupt line register.

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

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

4.24 Interrupt Pin Register

The value read from the interrupt pin register is function dependent and depends on the interrupt signaling mode, selected through bits 2–1 (INTMODE field) of the device control register (see Section 4.33) and the state of bit 29 (INTRTIE) in the system control register (see Section 4.29). When the INTRTIE bit is set, this register reads 0x01 (INTA) for both functions. See Table 4–5 for the complete description of the register contents.

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

Register: **Interrupt pin**
Type: Read-only
Offset: 3Dh
Default: 03h

Table 4–5. Interrupt Pin Register Cross Reference

INTERRUPT SIGNALING MODE	INTRTIE BIT	INTPIN FUNCTION 0	INTPIN FUNCTION 1
Parallel PCI interrupts only	0	0x01 (INTA)	0x02 (INTB)
Parallel IRQ and parallel PCI interrupts	0	0x01 (INTA)	0x02 (INTB)
IRQ serialized (IRQSER) and parallel PCI interrupts	0	0x01 (INTA)	0x02 (INTB)
IRQ and PCI serialized (IRQSER) interrupts (default)	0	0x01 (INTA)	0x02 (INTB)
Parallel PCI interrupts only	1	0x01 (INTA)	0x01 (INTA)
Parallel IRQ and parallel PCI interrupts	1	0x01 (INTA)	0x01 (INTA)
IRQ serialized (IRQSER) and parallel PCI interrupts	1	0x01 (INTA)	0x01 (INTA)
IRQ and PCI serialized (IRQSER) interrupts	1	0x01 (INTA)	0x01 (INTA)

4.25 Bridge Control Register

The bridge control register provides control over various PCI1420 bridging functions. Some bits in this register are global and are 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	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0

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

Table 4–6. Bridge Control Register

BIT	SIGNAL	TYPE	FUNCTION
15–11	RSVD	R	Reserved. Bits 15–11 return 0s when read.
10	POSTEN	R/W	Write posting enable. Enables write posting to and from the CardBus sockets. Write posting enables posting of write data on burst cycles. Operating with write posting disabled inhibits performance on burst cycles. Note that bursted write data can be posted, but various write transactions may not. Bit 10 is socket dependent and is not shared between functions 0 and 1.
9	PREFETCH1	R/W	Memory window 1 type. Bit 9 specifies whether or not memory window 1 is prefetchable. This bit is socket dependent. Bit 9 is encoded as: 0 = Memory window 1 is nonprefetchable. 1 = Memory window 1 is prefetchable (default).
8	PREFETCH0	R/W	Memory window 0 type. Bit 8 specifies whether or not memory window 0 is prefetchable. This bit is encoded as: 0 = Memory window 0 is nonprefetchable. 1 = Memory window 0 is prefetchable (default).
7	INTR	R/W	PCI interrupt – IREQ routing enable. Bit 7 selects whether PC Card functional interrupts are routed to PCI interrupts or the IRQ specified in the ExCA registers. 0 = Functional interrupts routed to PCI interrupts (default) 1 = Functional interrupts routed by ExCAs
6	CRST	R/W	CardBus reset. When bit 6 is set, $\overline{\text{CRST}}$ is asserted on the CardBus interface. $\overline{\text{CRST}}$ can also be asserted by passing a $\overline{\text{PRST}}$ assertion to CardBus. 0 = $\overline{\text{CRST}}$ deasserted 1 = $\overline{\text{CRST}}$ asserted (default)
5†	MABTMODE	R/W	Master abort mode. Bit 5 controls how the PCI1420 responds to a master abort when the PCI1420 is an initiator on the CardBus interface. This bit is common between each socket. 0 = Master aborts not reported (default) 1 = Signal target abort on PCI and $\overline{\text{SERR}}$ (if enabled)
4	RSVD	R	Reserved. Bit 4 returns 0 when read.
3	VGAEN	R/W	VGA enable. Bit 3 affects how the PCI1420 responds to VGA addresses. When this bit is set, accesses to VGA addresses are forwarded.
2	ISAEN	R/W	ISA mode enable. Bit 2 affects how the PCI1420 passes I/O cycles within the 64-Kbyte ISA range. This bit is not common between sockets. When this bit is set, the PCI1420 does not forward the last 768 bytes of each 1K I/O range to CardBus.
1†	CSERREN	R/W	$\overline{\text{CSERR}}$ enable. Bit 1 controls the response of the PCI1420 to $\overline{\text{CSERR}}$ signals on the CardBus bus. This bit is common between the two sockets. 0 = $\overline{\text{CSERR}}$ is not forwarded to PCI $\overline{\text{SERR}}$. 1 = $\overline{\text{CSERR}}$ is forwarded to PCI $\overline{\text{SERR}}$.
0†	CPERREN	R/W	CardBus parity error response enable. Bit 0 controls the response of the PCI1420 to CardBus parity errors. This bit is common between the two sockets. 0 = CardBus parity errors are ignored. 1 = CardBus parity errors are reported using $\overline{\text{CPERR}}$.

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

4.26 Subsystem Vendor ID Register

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

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

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

4.27 Subsystem ID Register

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

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

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

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

The PCI1420 supports the index/data scheme of accessing the ExCA registers, which is mapped by this register. An address written to this register is the address for the index register and the address + 1 is the data address. Using this access method, applications requiring index/data ExCA access can be supported. The base address can be mapped anywhere in 32-bit I/O space on a word boundary; hence, bit 0 is read-only, returning 1 when read. As specified in the *PCI to PCMCIA CardBus Bridge Register Description* (Yenta), this register is shared by functions 0 and 1. See Section 5, *ExCA Compatibility Registers*, for register offsets.

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

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

4.29 System Control Register

System-level initializations are performed through programming this doubleword register. Some of the bits are global and are written 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	R/W	R/W	R/W	R	R/W	R/W	R/C	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	System control															
Type	R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0

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

Table 4–7. System Control Register

BIT	SIGNAL	TYPE	FUNCTION
31–30†	SER_STEP	R/W	Serialized PCI interrupt routing step. Bits 31 and 30 configure the serialized PCI interrupt stream signaling and accomplish an even distribution of interrupts signaled on the four PCI interrupt slots. Bits 31 and 30 are global to all PCI1420 functions. 00 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ signal in $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ IRQSER slots 01 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ signal in $\overline{\text{INTB}}$/$\overline{\text{INTC}}$ IRQSER slots 10 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ signal in $\overline{\text{INTC}}$/$\overline{\text{INTD}}$ IRQSER slots 11 = $\overline{\text{INTA}}$/$\overline{\text{INTB}}$ signal in $\overline{\text{INTD}}$/$\overline{\text{INTA}}$ IRQSER slots
29†	INTRTIE	R/W	Tie internal PCI interrupts. When this bit is set, the $\overline{\text{INTA}}$ and $\overline{\text{INTB}}$ signals are tied together internally and are signaled as $\overline{\text{INTA}}$. $\overline{\text{INTA}}$ can then be shifted by using bits 31–30 (SER_STEP). This bit is global to all PCI1420 functions. When configuring the PCI1420 functions to share PCI interrupts, multifunction terminal MFUNC3 must be configured as IRQSER prior to setting the INTRTIE bit.
28	RSVD	R	Reserved. Bit 28 returns 0 when read.
27†	P2CCLK	R/W	P2C power switch clock. The PCI1420's CLOCK is used to clock the serial interface power switch and the internal state machine. The default state for bit 27 is 0, requiring an external clock source provided to the CLOCK pin (pin number E19 for the GHK package or pin number 151 for the PDV package). Bit 27 can be set to 1 allowing the internal oscillator to provide the clock signal. 0 = CLOCK provided externally, input to PCI1420 (default) 1 = CLOCK generated by internal oscillator and driven by PCI1420.
26	SMIROUTE	R/W	SMI interrupt routing. Bit 26 is shared between functions 0 and 1, and selects whether IRQ2 or CSC is signaled when a write occurs to power a PC Card socket. 0 = PC Card power change interrupts routed to IRQ2 (default) 1 = A CSC interrupt is generated on PC Card power changes.
25	SMISTATUS	R/C	SMI interrupt status. This socket-dependent bit is set when bit 24 (SMIENB) is set and a write occurs to set the socket power. Writing a 1 to bit 25 clears the status. 0 = SMI interrupt signaled (default) 1 = SMI interrupt not signaled
24†	SMIENB	R/W	SMI interrupt mode enable. When bit 24 is set and a write to the socket power control occurs, the SMI interrupt signaling is enabled and generates an interrupt. This bit is shared and defaults to 0 (disabled).
23	PCIPMEN	R/W	PCI bus power management interface specification revision 1.1 enable. 0 = Use PCI bus power management interface specification revision 1.0 implementation (default) 1 = Use PCI bus power management interface specification revision 1.1 implementation
22	CBRSVD	R/W	CardBus reserved terminals signaling. When a CardBus card is inserted and bit 22 is set, the RSVD CardBus terminals are driven low. When this bit is 0, these signals are placed in a high-impedance state. 0 = 3-state CardBus RSVD 1 = Drive Cardbus RSVD low (default)
21	VCCPROT	R/W	V_{CC} protection enable. Bit 21 is socket dependent. 0 = V_{CC} protection enabled for 16-bit cards (default) 1 = V_{CC} protection disabled for 16-bit cards
20	REDUCEZV	R/W	Reduced zoomed video enable. When this bit is enabled, pins A25–A22 of the card interface for PC Card-16 cards are placed in the high-impedance state. This bit should not be set for normal ZV operation. This bit is encoded as: 0 = Reduced zoomed video disabled (default) 1 = Reduced zoomed video enabled
19	CDREQEN	R/W	PC/PCI DMA card enable. When bit 19 is set, the PCI1420 allows 16-bit PC Cards to request PC/PCI DMA using the $\overline{\text{DREQ}}$ signaling. $\overline{\text{DREQ}}$ is selected through the socket DMA register 0. 0 = Ignore $\overline{\text{DREQ}}$ signaling from PC Cards (default) 1 = Signal DMA request on $\overline{\text{DREQ}}$
18–16	CDMACHAN	R/W	PC/PCI DMA channel assignment. Bits 18–16 are encoded as: 0–3 = 8-bit DMA channels 4 = PCI master; not used (default). 5–7 = 16-bit DMA channels

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

Table 4–7. System Control Register (Continued)

BIT	SIGNAL	TYPE	FUNCTION
15†	MRBURSTDN	R/W	Memory read burst enable downstream. When bit 15 is set, memory read transactions are allowed to burst downstream. 0 = Downstream memory read burst is disabled. 1 = Downstream memory read burst is enabled (default).
14†	MRBURSTUP	R/W	Memory read burst enable upstream. When bit 14 is set, the PCI1420 allows memory read transactions to burst upstream. 0 = Upstream memory read burst is disabled (default). 1 = Upstream memory read burst is enabled.
13	SOCACTIVE	R	Socket activity status. When set, bit 13 indicates access has been performed to or from a PC card and is cleared upon read of this status bit. This bit is socket-dependent. 0 = No socket activity (default) 1 = Socket activity
12	RSVD	R	Reserved. Bit 12 returns 1 when read.
11†	PWRSTREAM	R	Power stream in progress status bit. When set, bit 11 indicates that a power stream to the power switch is in progress and a powering change has been requested. This bit is cleared when the power stream is complete. 0 = Power stream is complete and delay has expired. 1 = Power stream is in progress.
10†	DELAYUP	R	Power-up delay in progress status. When set, bit 9 indicates that a power-up stream has been sent to the power switch and proper power may not yet be stable. This bit is cleared when the power-up delay has expired.
9†	DELAYDOWN	R	Power-down delay in progress status. When set, bit 10 indicates that a power-down stream has been sent to the power switch and proper power may not yet be stable. This bit is cleared when the power-down delay has expired.
8	INTERROGATE	R	Interrogation in progress. When set, bit 8 indicates an interrogation is in progress and clears when interrogation completes. This bit is socket dependent. 0 = Interrogation not in progress (default) 1 = Interrogation in progress
7	RSVD	R	Reserved. Bit 7 returns 0 when read.
6	PWRSAVINGS	R/W	Power savings mode enable. When this bit is set, if a CB card is inserted, idle, and without a CB clock, then the applicable CB state machine will not be clocked.
5†	SUBSYSRW	R/W	Subsystem ID (see Section 4.27), subsystem vendor ID (see Section 4.26), ExCA identification and revision (see Section 5.1) registers read/write enable. Bit 5 is shared by functions 0 and 1. 0 = Subsystem ID, subsystem vendor ID, ExCA identification and revision registers are read/write. 1 = Subsystem ID, subsystem vendor ID, ExCA identification and revision registers are read-only (default).
4†	CB_DPAR	R/W	CardBus data parity $\overline{\text{SERR}}$ signaling enable 0 = CardBus data parity not signaled on PCI $\overline{\text{SERR}}$ 1 = CardBus data parity signaled on PCI $\overline{\text{SERR}}$
3†	CDMA_EN	R/W	PC/PCI DMA enable. Bit 3 enables PC/PCI DMA when set if MFUNC0–MFUNC6 are configured for centralized DMA. 0 = Centralized DMA disabled (default) 1 = Centralized DMA enabled
2	ExCAPower	R/W	ExCA power control bit. 0 = Enables 3.3 V 1 = Enables 5 V
1†	KEEPCLK	R/W	Keep clock. This bit works with PCI and CB $\overline{\text{CLKRUN}}$ protocols. 0 = Allows normal functioning of both $\overline{\text{CLKRUN}}$ protocols.(default) 1 = Does not allow CB clock or PCI clock to be stopped using the $\overline{\text{CLKRUN}}$ protocols.
0	RIMUX	R/W	$\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ multiplex enable. 0 = $\overline{\text{RI\_OUT}}$ and $\overline{\text{PME}}$ are both routed to the $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ terminal. If both are enabled at the same time, then $\overline{\text{RI\_OUT}}$ has precedence over $\overline{\text{PME}}$. 1 = Only $\overline{\text{PME}}$ is routed to the $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ terminal.

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

4.30 Multifunction Routing Register

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

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

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

Table 4–8. Multifunction Routing Register

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

Table 4–8. Multifunction Routing Register (Continued)

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

4.31 Retry Status Register

The retry status register enables the retry timeout counters and displays the retry expiration status. The flags are set when the PCI1420 retries a PCI or CardBus master request and the master does not return within 2^{15} PCI clock cycles. The flags are cleared by writing a 1 to the bit. These bits are expected to be incorporated into the PCI command, PCI status, and bridge control registers by the PCI SIG. Access this register only through function 0. See Table 4–9 for a complete description of the register contents.

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

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

Table 4–9. Retry Status Register

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

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

4.32 Card Control Register

The card control register is provided for PCI1130 compatibility. $\overline{\text{RI\_OUT}}$ is enabled through this register, and the enable bit is shared between functions 0 and 1. See Table 4–10 for a complete description of the register contents.

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

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

Table 4–10. Card Control Register

BIT	SIGNAL	TYPE	FUNCTION
7†	RIENB	R/W	Ring indicate output enable. 0 = Disables any routing of $\overline{\text{RI\_OUT}}$ signal (default). 1 = Enables $\overline{\text{RI\_OUT}}$ signal for routing to the $\overline{\text{RI\_OUT}}$/$\overline{\text{PME}}$ terminal, when RIMUX is set to 0, and for routing to MFUNC2 or MFUNC4.
6	ZVENABLE	R/W	Compatibility ZV mode enable. When set, the corresponding PC Card socket interface ZV terminals enter a high-impedance state. This bit defaults to 0.
5	PORT_SEL	R/W	Port select. This bit controls the priority for the $\overline{\text{ZVSEL0}}$ and $\overline{\text{ZVSEL1}}$ signaling if bit 6 (ZVENABLE) is set in both functions. 0 = Socket 0 takes priority, as signaled through $\overline{\text{ZVSEL0}}$, when both sockets are in ZV mode. 1 = Socket 1 takes priority, as signaled through $\overline{\text{ZVSEL1}}$, when both sockets are in ZV mode.
4–3	RSVD	R	Reserved. Bits 4 and 3 return 0 when read.
2	AUD2MUX	R/W	CardBus audio-to-IRQMUX. When set, the CAUDIO CardBus signal is routed to the corresponding multifunction terminal which may be configured for CAUDPWM. When both socket 0 and 1 functions have AUD2MUX set, socket 0 takes precedence.
1	SPKROUTEN	R/W	Speaker out enable. When bit 1 is set, $\overline{\text{SPKR}}$ on the PC Card is enabled and is routed to SPKROUT. The $\overline{\text{SPKR}}$ signal from socket 0 is exclusive ORed with the $\overline{\text{SPKR}}$ signal from socket 1 and sent to SPKROUT. The SPKROUT terminal drives data only when either function's SPKROUTEN bit is set. This bit is encoded as: 0 = $\overline{\text{SPKR}}$ to SPKROUT not enabled 1 = $\overline{\text{SPKR}}$ to SPKROUT enabled
0	IFG	R/C	Interrupt flag. Bit 0 is the interrupt flag for 16-bit I/O PC Cards and for CardBus cards. Bit 0 is set when a functional interrupt is signaled from a PC Card interface and is socket dependent (that is, not global). Write back a 1 to clear this bit. 0 = No PC Card functional interrupt detected (default). 1 = PC Card functional interrupt detected.

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

4.33 Device Control Register

The device control register is provided for PCI1130 compatibility and contains bits that are shared between functions 0 and 1. The interrupt mode select is programmed through this register which is composed of PCI1420 global bits. The socket-capable force bits are also programmed through this register. See Table 4–11 for a complete description of the register contents.

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

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

Table 4–11. Device Control Register

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

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

4.34 Diagnostic Register

The diagnostic register is provided for internal TI test purposes. It is a read/write register, but only 0s should be written to it. See Table 4–12 for a complete description of the register contents.

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

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

Table 4–12. Diagnostic Register

BIT	SIGNAL	TYPE	FUNCTION
7†	TRUE_VAL	R/W	This bit defaults to 0. This bit is encoded as: 0 = Reads true values in PCI vendor ID and PCI device ID registers (default) 1 = Reads all 1s in reads to the PCI vendor ID and PCI device ID registers
6	AOSPMEN	R/W	Auto oscillator enable. This bit provides fail safe for the oscillator power management logic. If the problem arises with the logic, then this bit disables all the power management features of the oscillator. This bit is encoded as: 0 = Oscillator power management features enabled (default) 1 = Oscillator power management features disabled
5	CSC	R/W	CSC interrupt routing control 0 = CSC interrupts routed to PCI if ExCA 803 bit 4 = 1 1 = CSC interrupts routed to PCI if ExCA 805 bits 7–4 = 0000b. (default) In this case, the setting of ExCA 803 bit 4 is a “don’t care”
4†	DIAG4	R/W	Diagnostic RETRY_DIS. Delayed transaction disable.
3†	DIAG3	R/W	Diagnostic RETRY_EXT. Extends the latency from 16 to 64.
2†	DIAG2	R/W	Diagnostic DISCARD_TIM_SEL_CB. Set = 2^{10}, reset = 2^{15}.
1†	DIAG1	R/W	Diagnostic DISCARD_TIM_SEL_PCI. Set = 2^{10}, reset = 2^{15}.
0	ASYNC	R/W	Asynchronous interrupt enable. 0 = CSC interrupt is not generated asynchronously 1 = CSC interrupt is generated asynchronously (default)

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

4.35 Socket DMA Register 0

The socket DMA register 0 provides control over the PC Card DMA request ($\overline{\text{DREQ}}$) signaling. 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	Socket DMA register 0															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Socket DMA register 0															
Type	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Table 4–13. Socket DMA Register 0

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

4.36 Socket DMA Register 1

The socket DMA register 1 provides control over the distributed DMA (DDMA) registers and the PCI portion of DMA transfers. The DMA base address locates the DDMA registers in a 16-byte region within the first 64K bytes of PCI I/O address space. See Table 4–14 for a complete description of the register contents.

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

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

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

Table 4–14. Socket DMA Register 1

BIT	SIGNAL	TYPE	FUNCTION
31–16	RSVD	R	Reserved. Bits 31–16 return 0s when read.
15–4	DMABASE	R/W	DMA base address. Locates the socket's DMA registers in PCI I/O space. This field represents a 16-bit PCI I/O address. The upper 16 bits of the address are hardwired to 0, forcing this window to within the lower 64K bytes of I/O address space. The lower 4 bits are hardwired to 0 and are included in the address decode. Thus, the window is aligned to a natural 16-byte boundary.
3	EXTMODE	R	Extended addressing. This feature is not supported by the PCI1420 and always returns a 0.
2–1	XFERSIZE	R/W	Transfer size. Bits 2 and 1 specify the width of the DMA transfer on the PC Card interface and are encoded as: 00 = Transfers are 8 bits (default). 01 = Transfers are 16 bits. 10 = Reserved 11 = Reserved
0	DDMAEN	R/W	DDMA registers decode enable. Enables the decoding of the distributed DMA registers based on the value of DMABASE. 0 = Disabled (default) 1 = Enabled

4.37 Capability ID Register

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

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

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

4.38 Next-Item Pointer Register

The next-item pointer register indicates the next item in the linked list of the PCI power management capabilities. Because the PCI1420 functions include only one capabilities item, this register returns 0s when read.

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

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

4.39 Power Management Capabilities Register

This register contains information on the capabilities of the PC Card function related to power management. Both PCI1420 CardBus bridge functions support D0, D1, D2, and D3 power states. See Table 4–15 for a complete description of the register contents.

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

Register: **Power management capabilities**
Type: Read/Write, Read-only
Offset: A2h
Default: FE11h

Table 4–15. Power Management Capabilities Register

BIT	SIGNAL	TYPE	FUNCTION
15	PME_Support	R/W	$\overline{\text{PME}}$ support. This 5-bit field indicates the power states from which the PCI1420 device functions may assert $\overline{\text{PME}}$. A 0 (zero) for any bit indicates that the function cannot assert the $\overline{\text{PME}}$ signal while in that power state. These five bits return 11111b when read. Each of these bits is described below: Bit 15 defaults to the value 1 indicating the $\overline{\text{PME}}$ signal can be asserted from the $D3_{cold}$ state. This bit is R/W because wake-up support from $D3_{cold}$ is contingent on the system providing an auxiliary power source to the V_{CC} terminals. If the system designer chooses not to provide an auxiliary power source to the V_{CC} terminals for $D3_{cold}$ wake-up support, then BIOS should write a 0 to this bit.
14–11	PME_Support	R	Bit 14 contains the value 1, indicating that the $\overline{\text{PME}}$ signal can be asserted from $D3_{hot}$ state. Bit 13 contains the value 1, indicating that the $\overline{\text{PME}}$ signal can be asserted from D2 state. Bit 12 contains the value 1, indicating that the $\overline{\text{PME}}$ signal can be asserted from D1 state. Bit 11 contains the value 1, indicating that the $\overline{\text{PME}}$ signal can be asserted from the D0 state.
10	D2_Support	R	D2 support. Bit 10 returns a 1 when read, indicating that the CardBus function supports the D2 device power state.
9	D1_Support	R	D1 support. Bit 9 returns a 1 when read, indicating that the CardBus function supports the D1 device power state.
8–6	RSVD	R	Reserved. Bits 8–6 return 0s when read.
5	DSI	R	Device-specific initialization. Bit 5 returns 1 when read, indicating that the CardBus controller function require special initialization (beyond the standard PCI configuration header) before the generic class device driver is able to use it.
4	AUX_PWR	R	Auxiliary power source. Bit 4 is meaningful only if bit 15 (PME_Support, $D3_{cold}$) is set. When bit 4 is set, it indicates that support for $\overline{\text{PME}}$ in $D3_{cold}$ requires auxiliary power supplied by the system by way of a proprietary delivery vehicle. When bit 4 is 0, it indicates that the function supplies its own auxiliary power source.
3	PMECLK	R	$\overline{\text{PME}}$ clock. Bit 3 returns 0 when read, indicating that no host bus clock is required for the PCI1420 to generate $\overline{\text{PME}}$.
2–0	VERSION	R	Version. Bits 2–0 return 001b when read, indicating that there are four bytes of general-purpose power management (PM) registers as described in the *PCI Bus Power Management Interface Specification*.

4.40 Power Management Control/Status Register

The power management control/status register determines and changes the current power state of the PCI1420 CardBus function. The contents of this register are not affected by the internally-generated reset caused by the transition from $D3_{hot}$ to D0 state. All PCI, ExCA, and CardBus registers are reset as a result of a $D3_{hot}$ to D0 state transition. TI-specific registers, PCI power management registers, and the legacy base address register are not reset. See Table 4–16 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	R/C	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Table 4–16. Power Management Control/Status Register

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

4.41 Power Management Control/Status Register Bridge Support Extensions

The power management control/status register bridge support extensions support PCI bridge specific functionality. See Table 4–17 for a complete description of the register contents.

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

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

Table 4–17. Power Management Control/Status Register Bridge Support Extensions

BIT	SIGNAL	TYPE	FUNCTION
7	BPCC_EN	R	BPCC_Enable. Bus power/clock control enable. This bit returns 1 when read. This bit is encoded as: 0 = Bus power/clock control is disabled. 1 = Bus power/clock control is enabled (default). A 0 indicates that the bus power/clock control policies defined in the *PCI Bus Power Management Interface Specification* are disabled. When the bus power/clock control enable mechanism is disabled, the bridge's power management control/status register power state field (see Section 4.40, bits 1–0) cannot be used by the system software to control the power or the clock of the bridge's secondary bus. A 1 indicates that the bus power/clock control mechanism is enabled.
6	B2_$\overline{B3}$	R	B2/B3 support for $D3_{hot}$. The state of this bit determines the action that is to occur as a direct result of programming the function to $D3_{hot}$. This bit is only meaningful if bit 7 (BPCC_EN) is a 1. This bit is encoded as: 0 = when the bridge is programmed to $D3_{hot}$, its secondary bus will have its power removed (B3). 1 = when the bridge function is programmed to $D3_{hot}$, its secondary bus's PCI clock will be stopped (B2). (Default)
5–0	RSVD	R	Reserved. Bits 5–0 return 0s when read.

4.42 Power Management Data Register

The power management data register returns 0s when read, since 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**
Type: Read-only
Offset: A7h (functions 0, 1)
Default: 00h

4.43 General-Purpose Event Status Register

The general-purpose event status register contains status bits that are set when events occur that are controlled by the general-purpose control register. The bits in this register and the corresponding $\overline{GPE}$ are cleared by writing a 1 to the corresponding bit location. The status bits in this register do not depend upon the state of a corresponding bit in the general-purpose enable register. Access this register only through function 0. See Table 4–18 for a complete description of the register contents.

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

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

Table 4–18. General-Purpose Event Status Register

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

4.44 General-Purpose Event Enable Register

The general-purpose event enable register contains bits that are set to enable a $\overline{\text{GPE}}$ signal. The $\overline{\text{GPE}}$ signal is driven until the corresponding status bit is cleared and the event is serviced. The $\overline{\text{GPE}}$ can only be signaled if one of the multifunction terminals, MFUNC6–MFUNC0, is configured for $\overline{\text{GPE}}$ signaling. Access this register only through function 0. See Table 4–19 for a complete description of the register contents.

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

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

Table 4–19. General-Purpose Event Enable Register

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

4.45 General-Purpose Input Register

The general-purpose input register provides the logical value of the data input from the GPI terminals, MFUNC5, MFUNC4, and MFUNC2–MFUNC0. Access this register only through function 0. See Table 4–20 for a complete description of the register contents.

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

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

Table 4–20. General-Purpose Input Register

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

4.46 General-Purpose Output Register

The general-purpose output register is used for control of the general-purpose outputs. Access this register only through function 0. See Table 4–21 for a complete description of the register contents.

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

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

Table 4–21. General-Purpose Output Register

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

4.47 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 both 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 must be programmed with both the 7-bit slave address and the read/write indicator bit must be set, and bit 5 (REQBUSY) in the serial bus control and status register (see Section 4.50) must be polled until clear. Then the contents of this 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 data							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

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

Table 4–22. Serial Bus Data Register

BIT	SIGNAL	TYPE	FUNCTION
7–0	SBDATA	R/W	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.

4.48 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 must be set, and bit 5 (REQBUSY) in the serial bus control and status register (see Section 4.50) 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 index							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

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

Table 4–23. Serial Bus Index Register

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

4.49 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 must be set, and bit 5 (REQBUSY) in the serial bus control and status register (see Section 4.50) 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 4–24 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Serial bus slave address							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

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

Table 4–24. Serial Bus Slave Address Register

BIT	SIGNAL	TYPE	FUNCTION
7–1	SLAVADDR	R/W	Serial bus slave address. This bit field represents the slave address of a read or write transaction on the serial interface.
0	RWCMD	R/W	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

4.50 Serial Bus Control and Status Register

The serial bus control and status register communicates serial bus status information and select 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 4–25 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name	Serial bus control and status							
Type	R/W	R	R	R	R/C	R/W	R/C	R/C
Default	0	0	0	0	0	0	0	0

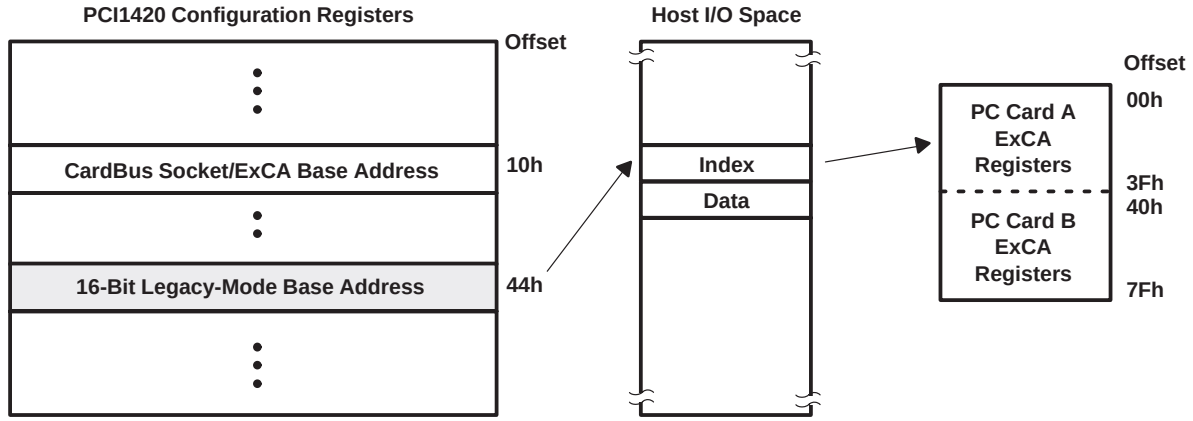
Register: **Serial bus control and status**
Type: Read-only, Read/Write, Read/Write to Clear
Offset: B3h (function 0)
Default: 00h

Table 4–25. Serial Bus Control and Status Register

BIT	SIGNAL	TYPE	FUNCTION
7	PROT_SEL	R/W	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 4.48) is not output by the PCI1420 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 4.49). Bit 5 must be polled on reads from the serial interface. After the byte read access has been requested, the read data is valid in the serial bus data register.
4	ROMBUSY	R	Serial EEPROM Busy status. Bit 4 indicates the status of the PCI1420 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	R/C	Serial bus detect. When bit 3 is set, it indicates that the serial bus interface is detected. A pulldown resistor must be implemented on the LATCH terminal for bit 3 to be set. If bit 3 is reset, then the MFUNC4 and MFUNC1 terminals can be used for alternate functions such as general-purpose inputs and outputs. 0 = Serial bus interface not detected 1 = Serial bus interface detected
2	SBTEST	R/W	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	R/C	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 write back 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	R/C	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 3.6.1, *Serial Bus Interface Implementation*, for details on EEPROM data format. Bit 0 is cleared by a write back of 1. 0 = No error detected during auto-load from serial bus EEPROM 1 = Data error detected during auto-load from serial bus EEPROM

5 ExCA Compatibility Registers (Functions 0 and 1)

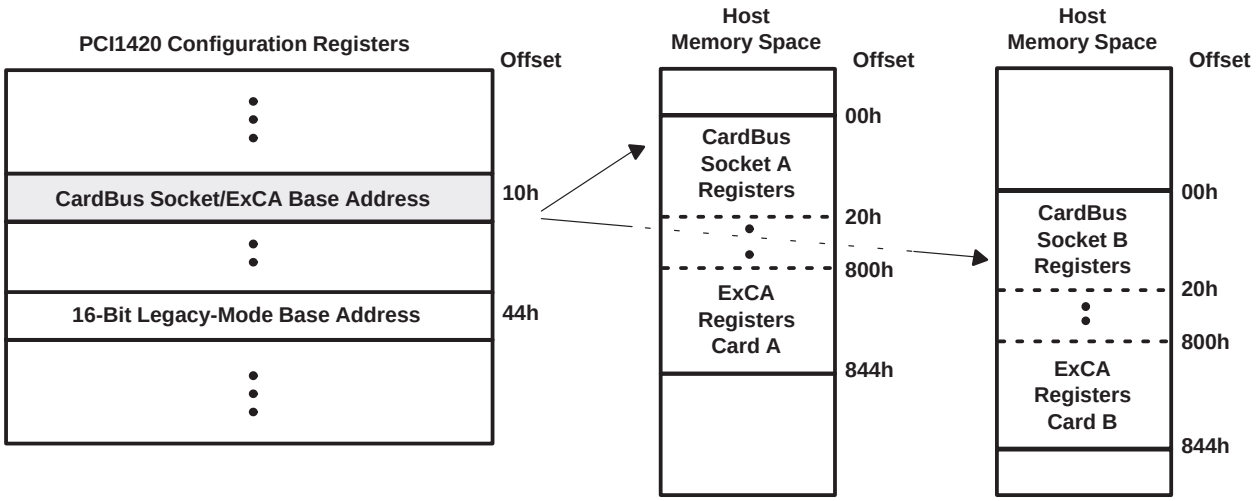
The ExCA registers implemented in the PCI1420 are register-compatible with the Intel 82365SL–DF PCMCIA controller. ExCA registers are identified by an offset value that is compatible with the legacy I/O index/data scheme used on the Intel 82365 ISA controller. The ExCA registers are accessed through this scheme by writing the register offset value into the index register (I/O base) and reading or writing the data register (I/O base + 1). The I/O base address used in the index/data scheme is programmed in the PC Card 16-bit I/F legacy mode base address register (see Section 4.28), which is shared by both card sockets. The offsets from this base address run contiguous from 00h to 3Fh for socket A, and from 40h to 7Fh for socket B. See Figure 5–1 for an ExCA I/O mapping illustration.



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

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

The TI PCI1420 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 base address register (see Section 4.12) at memory offset 800h. Each socket has a separate base address programmable by function. See Figure 5–2 for an ExCA memory mapping illustration. Note that memory offsets are 800h–844h for both functions 0 and 1. This illustration also identifies the CardBus socket register mapping, which is mapped into the same 4K window at memory offset 0h.



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

Figure 5–2. ExCA Register Access Through Memory

The interrupt registers, as defined by the 82365SL–DL Specification, in the ExCA register set control such card functions as reset, type, interrupt routing, and interrupt enables. Special attention must be paid to the interrupt routing registers and the host interrupt signaling method selected for the PCI1420 to ensure that all possible PCI1420 interrupts can potentially be routed to the programmable interrupt controller. The ExCA registers that are critical to the interrupt signaling are the ExCA interrupt and general control register (see Section 5.4) and the ExCA card status-change-interrupt configuration register (see Section 5.6).

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

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

Table 5–1. ExCA Registers and Offsets

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

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

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

5.1 ExCA Identification and Revision Register (Index 00h)

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

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

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

Table 5–2. ExCA Identification and Revision Register (Index 00h)

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 PCI1420. The PCI1420 supports both I/O and memory 16-bit PC cards.
5–4	RSVD	R/W	Reserved. Bits 5 and 4 can be used for Intel 82365SL-DF emulation.
3–0	365REV	R/W	Intel 82365SL-DF revision. This field stores the Intel 82365SL-DF revision supported by the PCI1420. Host software can read this field to determine compatibility to the Intel 82365SL-DF register set. Writing 0010b to this field puts the controller in 82365SL mode.

5.2 ExCA Interface Status Register (Index 01h)

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

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

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

Table 5–3. ExCA Interface Status Register (Index 01h)

BIT	SIGNAL	TYPE	FUNCTION
7	RSVD	R	Reserved. Bit 7 returns 0 when read.
6	CARDPWR	R	Card Power. Bit 6 indicates the current power status of the PC Card socket. This bit reflects how the ExCA power control register (see Section 5.3) is programmed. Bit 6 is encoded as: 0 = V_{CC} and V_{PP} to the socket turned off (default) 1 = V_{CC} and V_{PP} to the socket turned on
5	READY	R	Ready. Bit 5 indicates the current status of the READY signal at the PC Card interface. 0 = PC Card not ready for data transfer 1 = PC Card ready for data transfer
4	CARDWP	R	Card write protect. Bit 4 indicates the current status of WP at the PC Card interface. This signal reports to the PCI1420 whether or not the memory card is write protected. Furthermore, write protection for an entire PCI1420 16-bit memory window is available by setting the appropriate bit in the memory window offset-address high-byte register. 0 = WP is 0. PC Card is read/write. 1 = WP is 1. PC Card is read-only.
3	CDETECT2	R	Card detect 2. Bit 3 indicates the status of $\overline{CD2}$ at the PC Card interface. Software may use this and bit 2 (CDET$\overline{ECT1}$) to determine if a PC Card is fully seated in the socket. 0 = $\overline{CD2}$ is 1. No PC Card is inserted. 1 = $\overline{CD2}$ is 0. PC Card is at least partially inserted.
2	CDETECT1	R	Card detect 1. Bit 2 indicates the status of $\overline{CD1}$ at the PC Card interface. Software may use this and bit 3 (CDET$\overline{ECT2}$) to determine if a PC Card is fully seated in the socket. 0 = $\overline{CD1}$ is 1. No PC Card is inserted. 1 = $\overline{CD1}$ is 0. PC Card is at least partially inserted.
1–0	BVDSTAT	R	Battery voltage detect. When a 16-bit memory card is inserted, the field indicates the status of the battery voltage detect signals (BVD1, BVD2) at the PC Card interface, where bit 1 reflects the BVD2 status and bit 0 reflects BVD1. 00 = Battery dead 01 = Battery dead 10 = Battery low; warning 11 = Battery good When a 16-bit I/O card is inserted, this field indicates the status of $\overline{SPKR}$ (bit 1) and $\overline{STSCHG}$ (bit 0) at the PC Card interface. In this case, the two bits in this field directly reflect the current state of these card outputs.

5.3 ExCA Power Control Register (Index 02h)

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

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

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

Table 5–4. ExCA Power Control Register 82365SL Support (Index 02h)

BIT	SIGNAL	TYPE	FUNCTION
7	COE	R/W	Card output enable. Bit 7 controls the state of all of the 16-bit outputs on the PCI1420. 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	R/W	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	R/W	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 (see Section 4.29).
3–2	RSVD	R	Reserved. Bits 3 and 2 return 0s when read.
1–0	EXCAVPP	R/W	PC Card V_{PP} power control. Bits 1 and 0 are used to request changes to card V_{PP}. The PCI1420 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 82365SL-DF Support (Index 02h)

BIT	SIGNAL	TYPE	FUNCTION
7	COE	R/W	Card output enable. Bit 7 controls the state of all of the 16-bit outputs on the PCI1420. This bit is encoded as: 0 = 16-bit PC Card outputs disabled (default) 1 = 16-bit PC Card outputs enabled
6–5	RSVD	R	Reserved. Bits 6 and 5 return 0s when read.
4–3	EXCAVCC	R/W	V_{CC}. Bits 4 and 3 are used to request changes to card V_{CC}. This field is encoded as: 00 = 0 V (default) 01 = 0 V reserved 10 = 5 V 11 = 3 V
2	RSVD	R	Reserved. Bit 2 returns 0 when read.
1–0	EXCAVPP	R/W	V_{PP}. Bits 1 and 0 are used to request changes to card V_{PP}. The PCI1420 ignores this field unless V_{CC} to the socket is enabled. This field is encoded as: 00 = No connection (default) 01 = V_{CC} 10 = 12 V 11 = Reserved

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

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

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

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

Table 5–6. ExCA Interrupt and General Control Register (Index 03h)

BIT	SIGNAL	TYPE	FUNCTION
7	RINGEN	R/W	Card ring indicate enable. Bit 7 enables the ring indicate function of BVD1/$\overline{RI}$. This bit is encoded as: 0 = Ring indicate disabled (default) 1 = Ring indicate enabled
6	RESET	R/W	Card reset. Bit 6 controls the 16-bit PC Card RESET, and allows host software to force a card reset. Bit 6 affects 16-bit cards only. This bit is encoded as: 0 = RESET signal asserted (default) 1 = RESET signal deasserted
5	CARDTYPE	R/W	Card type. Bit 5 indicates the PC card type. This bit is encoded as: 0 = Memory PC Card installed (default) 1 = I/O PC Card installed
4	CSCROUTE	R/W	PCI Interrupt CSC routing enable bit. When bit 4 is set (high), the card status change interrupts are routed to PCI interrupts. When low, the card status change interrupts are routed using bits 7–4 (CSCSELECT field) in the ExCA card status change interrupt configuration register (see Section 5.6). This bit is encoded as: 0 = CSC interrupts are routed by ExCA registers (default). 1 = CSC interrupts are routed to PCI interrupts.
3–0	INTSELECT	R/W	Card interrupt select for I/O PC Card functional interrupts. Bits 3–0 select the interrupt routing for I/O PC Card functional interrupts. This field is encoded as: 0000 = No interrupt routing (default) . CSC interrupts routed to PCI interrupts. This bit setting is OR'ed with bit 4 (CSCROUTE) for backwards compatibility. 0001 = IRQ1 enabled 0010 = SMI enabled 0011 = IRQ3 enabled 0100 = IRQ4 enabled 0101 = IRQ5 enabled 0100 = IRQ6 enabled 0111 = IRQ7 enabled 1000 = IRQ8 enabled 1001 = IRQ9 enabled 1010 = IRQ10 enabled 1011 = IRQ11 enabled 1100 = IRQ12 enabled 1101 = IRQ13 enabled 1110 = IRQ14 enabled 1111 = IRQ15 enabled

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

The ExCA card status-change register controls interrupt routing for I/O interrupts as well as other critical 16-bit PC Card functions. The register enables these interrupt sources to generate an interrupt to the host. When the interrupt source is disabled, the corresponding bit in this register always reads 0. When an interrupt source is enabled, the corresponding bit in this register is set to indicate that the interrupt source is active. After generating the interrupt to the host, the interrupt service routine must read this register to determine the source of the interrupt. The interrupt service routine is responsible for resetting the bits in this register as well. Resetting a bit is accomplished by one of two methods: a read of this register or an explicit write back of 1 to the status bit. The choice of these two methods is based on bit 2 (interrupt flag clear mode select) in the ExCA global control register (see Section 5.22). 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 (Index 04h)

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 PCI1420 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 PCI1420 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 PCI1420 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 $\overline{RI}$.

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

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

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

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

Table 5–8. ExCA Card Status-Change-Interrupt Configuration Register (Index 05h)

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

5.7 ExCA Address Window Enable Register (Index 06h)

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 PCI1420 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	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

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

Table 5–9. ExCA Address Window Enable Register (Index 06h)

BIT	SIGNAL	TYPE	FUNCTION
7	IOWIN1EN	R/W	I/O window 1 enable. Bit 7 enables/disables I/O window 1 for the PC Card. This bit is encoded as: 0 = I/O window 1 disabled (default) 1 = I/O window 1 enabled
6	IOWIN0EN	R/W	I/O window 0 enable. Bit 6 enables/disables I/O window 0 for the PC Card. This bit is encoded as: 0 = I/O window 0 disabled (default) 1 = I/O window 0 enabled
5	RSVD	R	Reserved. Bit 5 returns 0 when read.
4	MEMWIN4EN	R/W	Memory window 4 enable. Bit 4 enables/disables memory window 4 for the PC Card. This bit is encoded as: 0 = Memory window 4 disabled (default) 1 = Memory window 4 enabled
3	MEMWIN3EN	R/W	Memory window 3 enable. Bit 3 enables/disables memory window 3 for the PC Card. This bit is encoded as: 0 = Memory window 3 disabled (default) 1 = Memory window 3 enabled
2	MEMWIN2EN	R/W	Memory window 2 enable. Bit 2 enables/disables memory window 2 for the PC Card. This bit is encoded as: 0 = Memory window 2 disabled (default) 1 = Memory window 2 enabled
1	MEMWIN1EN	R/W	Memory window 1 enable. Bit 1 enables/disables memory window 1 for the PC Card. This bit is encoded as: 0 = Memory window 1 disabled (default) 1 = Memory window 1 enabled
0	MEMWIN0EN	R/W	Memory window 0 enable. Bit 0 enables/disables memory window 0 for the PC Card. This bit is encoded as: 0 = Memory window 0 disabled (default) 1 = Memory window 0 enabled

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

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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

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

Table 5–10. ExCA I/O Window Control Register (Index 07h)

BIT	SIGNAL	TYPE	FUNCTION
7	WAITSTATE1	R/W	I/O window 1 wait state. Bit 7 controls the I/O window 1 wait state for 16-bit I/O accesses. Bit 7 has no effect on 8-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 16-bit cycles have standard length (default). 1 = 16-bit cycles are extended by one equivalent ISA wait state.
6	ZEROWS1	R/W	I/O window 1 zero wait state. Bit 6 controls the I/O window 1 wait state for 8-bit I/O accesses. Bit 6 has no effect on 16-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 8-bit cycles have standard length (default). 1 = 8-bit cycles are reduced to equivalent of three ISA cycles.
5	IOSIS16W1	R/W	I/O window 1 $\overline{\text{IOIS16}}$ source. Bit 5 controls the I/O window 1 automatic data sizing feature that uses $\overline{\text{IOIS16}}$ from the PC Card to determine the data width of the I/O data transfer. This bit is encoded as: 0 = Window data width determined by DATASIZE1, bit 4 (default). 1 = Window data width determined by $\overline{\text{IOIS16}}$.
4	DATASIZE1	R/W	I/O window 1 data size. Bit 4 controls the I/O window 1 data size. Bit 4 is ignored if 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	R/W	I/O window 0 wait state. Bit 3 controls the I/O window 0 wait state for 16-bit I/O accesses. Bit 3 has no effect on 8-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 16-bit cycles have standard length (default). 1 = 16-bit cycles are extended by one equivalent ISA wait state.
2	ZEROWS0	R/W	I/O window 0 zero wait state. Bit 2 controls the I/O window 0 wait state for 8-bit I/O accesses. Bit 2 has no effect on 16-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 8-bit cycles have standard length (default). 1 = 8-bit cycles are reduced to equivalent of three ISA cycles.
1	IOSIS16W0	R/W	I/O window 0 $\overline{\text{IOIS16}}$ source. Bit 1 controls the I/O window 0 automatic data sizing feature that uses $\overline{\text{IOIS16}}$ from the PC Card to determine the data width of the I/O data transfer. This bit is encoded as: 0 = Window data width is determined by DATASIZE0, bit 0 (default). 1 = Window data width is determined by $\overline{\text{IOIS16}}$.
0	DATASIZE0	R/W	I/O window 0 data size. Bit 0 controls the I/O window 0 data size. Bit 0 is ignored if 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 (Index 08h, 0Ch)

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

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

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

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

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 end address.

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

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

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

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

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

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

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

These registers contain the high byte of the 16-bit I/O window end address for I/O windows 0 and 1. The 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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

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

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

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

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

Register: **ExCA memory window 0 start-address low byte**
Offset: CardBus socket address + 810h; Card A ExCA offset 10h
Card B ExCA offset 50h
Register: **ExCA memory window 1 start-address low byte**
Offset: CardBus socket address + 818h; Card A ExCA offset 18h
Card B ExCA offset 58h
Register: **ExCA memory window 2 start-address low byte**
Offset: CardBus socket address + 820h; Card A ExCA offset 20h
Card B ExCA offset 60h
Register: **ExCA memory window 3 start-address low byte**
Offset: CardBus socket address + 828h; Card A ExCA offset 28h
Card B ExCA offset 68h
Register: **ExCA memory window 4 start-address low byte**
Offset: CardBus socket address + 830h; Card A ExCA offset 30h
Card B ExCA offset 70h
Type: Read/Write
Default: 00h
Size: One byte

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

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

Register: **ExCA memory window 0 start-address high byte**
Offset: CardBus socket address + 811h; Card A ExCA offset 11h
Card B ExCA offset 51h
Register: **ExCA memory window 1 start-address high byte**
Offset: CardBus socket address + 819h; Card A ExCA offset 19h
Card B ExCA offset 59h
Register: **ExCA memory window 2 start-address high byte**
Offset: CardBus socket address + 821h; Card A ExCA offset 21h
Card B ExCA offset 61h
Register: **ExCA memory window 3 start-address high byte**
Offset: CardBus socket address + 829h; Card A ExCA offset 29h
Card B ExCA offset 69h
Register: **ExCA memory window 4 start-address high byte**
Offset: CardBus socket address + 831h; Card A ExCA offset 31h
Card B ExCA offset 71h
Type: Read/Write
Default: 00h
Size: One byte

Table 5–11. ExCA Memory Windows 0–4 Start-Address High-Byte Registers (Index 11h, 19h, 21h, 29h, 31h)

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

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

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

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

Register: **ExCA memory window 0 end-address low byte**
Offset: CardBus socket address + 812h; Card A ExCA offset 12h
Card B ExCA offset 52h
Register: **ExCA memory window 1 end-address low byte**
Offset: CardBus socket address + 81Ah; Card A ExCA offset 1Ah
Card B ExCA offset 5Ah
Register: **ExCA memory window 2 end-address low byte**
Offset: CardBus socket address + 822h; Card A ExCA offset 22h
Card B ExCA offset 62h
Register: **ExCA memory window 3 end-address low byte**
Offset: CardBus socket address + 82Ah; Card A ExCA offset 2Ah
Card B ExCA offset 6Ah
Register: **ExCA memory window 4 end-address low byte**
Offset: CardBus socket address + 832h; Card A ExCA offset 32h
Card B ExCA offset 72h
Type: Read/Write
Default: 00h
Size: One byte

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

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

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

Table 5–12. ExCA Memory Windows 0–4 End-Address High-Byte Registers (Index 13h, 1Bh, 23h, 2Bh, 33h)

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

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

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

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

Register: **ExCA memory window 0 offset-address low byte**
Offset: CardBus socket address + 814h; Card A ExCA offset 14h
Card B ExCA offset 54h
Register: **ExCA memory window 1 offset-address low byte**
Offset: CardBus socket address + 81Ch; Card A ExCA offset 1Ch
Card B ExCA offset 5Ch
Register: **ExCA memory window 2 offset-address low byte**
Offset: CardBus socket address + 824h; Card A ExCA offset 24h
Card B ExCA offset 64h
Register: **ExCA memory window 3 offset-address low byte**
Offset: CardBus socket address + 82Ch; Card A ExCA offset 2Ch
Card B ExCA offset 6Ch
Register: **ExCA memory window 4 offset-address low byte**
Offset: CardBus socket address + 834h; Card A ExCA offset 34h
Card B ExCA offset 74h
Type: Read/Write
Default: 00h
Size: One byte

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

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

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

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

Table 5–13. ExCA Memory Windows 0–4 Offset-Address High-Byte Registers (Index 15h, 1Dh, 25h, 2Dh, 35h)

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

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

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

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

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

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

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

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

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

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

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

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

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

Table 5–14. ExCA Card Detect and General Control Register (Index 16h)

BIT	SIGNAL	TYPE	FUNCTION
7	VS2STAT	R	$\overline{VS2}$ state. Bit 7 reports the current state of $\overline{VS2}$ at the PC Card interface and, therefore, does not have a default value. 0 = $\overline{VS2}$ low 1 = $\overline{VS2}$ high
6	VS1STAT	R	$\overline{VS1}$ state. Bit 6 reports the current state of $\overline{VS1}$ at the PC Card interface and, therefore, does not have a default value. 0 = $\overline{VS1}$ low 1 = $\overline{VS1}$ high
5	SWCSC	R/W	Software card detect interrupt. If bit 3 (CDEN) in the ExCA card status-change-interrupt configuration register is set (see Section 5.6), then writing a 1 to bit 5 causes a card-detect card-status change interrupt for the associated card socket. If bit 3 (CDEN) in the ExCA card status-change-interrupt configuration register is cleared to 0 (see Section 5.6), then writing a 1 to bit 5 has no effect. A read operation of this bit always returns 0.
4	CDRESUME	R/W	Card detect resume enable. If bit 4 is set to 1, then once a card detect change has been detected on $\overline{CD1}$ and $\overline{CD2}$ inputs, $\overline{RI\_OUT}$ goes from high to low. $\overline{RI\_OUT}$ remains low until bit 0 (card status change) in the ExCA card status-change register is cleared (see Section 5.5). If this bit is a 0, then the card detect resume functionality is disabled. 0 = Card detect resume disabled (default) 1 = Card detect resume enabled
3–2	RSVD	R	Reserved. Bits 3 and 2 return 0s when read.
1	REGCONFIG	R/W	Register configuration on card removal. Bit 1 controls how the ExCA registers for the socket react to a card removal event. This bit is encoded as: 0 = No change to ExCA registers on card removal (default) 1 = Reset ExCA registers on card removal
0	RSVD	R	Reserved. Bit 0 returns 0 when read.

5.22 ExCA Global Control Register (Index 1Eh)

The ExCA global control register controls both PC Card sockets and is not duplicated for each socket. The host interrupt mode bits in this register are retained for Intel 82365SL-DF compatibility. See Table 5–15 for a complete description of the register contents.

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

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

Table 5–15. ExCA Global Control Register (Index 1Eh)

BIT	SIGNAL	TYPE	FUNCTION
7–5	RSVD	R	Reserved. Bits 7–5 return 0s when read.
4	INTMODEB	R/W	Level/edge interrupt mode select – card B. Bit 4 selects the signaling mode for the PCI1420 host interrupt for card B interrupts. This bit is encoded as: 0 = Host interrupt is edge mode (default). 1 = Host interrupt is level mode.
3	INTMODEA	R/W	Level/edge interrupt mode select – card A. Bit 3 selects the signaling mode for the PCI1420 host interrupt for card A interrupts. This bit is encoded as: 0 = Host interrupt is edge mode (default). 1 = Host interrupt is level mode.
2	IFCMODE	R/W	Interrupt flag clear mode select. Bit 2 selects the interrupt flag clear mechanism for the flags in the ExCA card status change register (see Section 5.5). This bit is encoded as: 0 = Interrupt flags are cleared by read of CSC register (default). 1 = Interrupt flags are cleared by explicit write back of 1.
1	CSCMODE	R/W	Card status change level/edge mode select. Bit 1 selects the signaling mode for the PCI1420 host interrupt for card status changes. This bit is encoded as: 0 = Host interrupt is edge mode (default). 1 = Host interrupt is level mode.
0	PWRDWN	R/W	Power-down mode select. When bit 0 is set to 1, the PCI1420 is in power-down mode. In power-down mode, the PCI1420 card outputs are high impedance until an active cycle is executed on the card interface. Following an active cycle, the outputs are again high impedance. The PCI1420 still receives DMA requests, functional interrupts, and/or card status change interrupts; however, an actual card access is required to wake up the interface. This bit is encoded as: 0 = Power-down mode is disabled (default). 1 = Power-down mode is enabled.

5.23 ExCA Memory Windows 0–4 Page Register

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

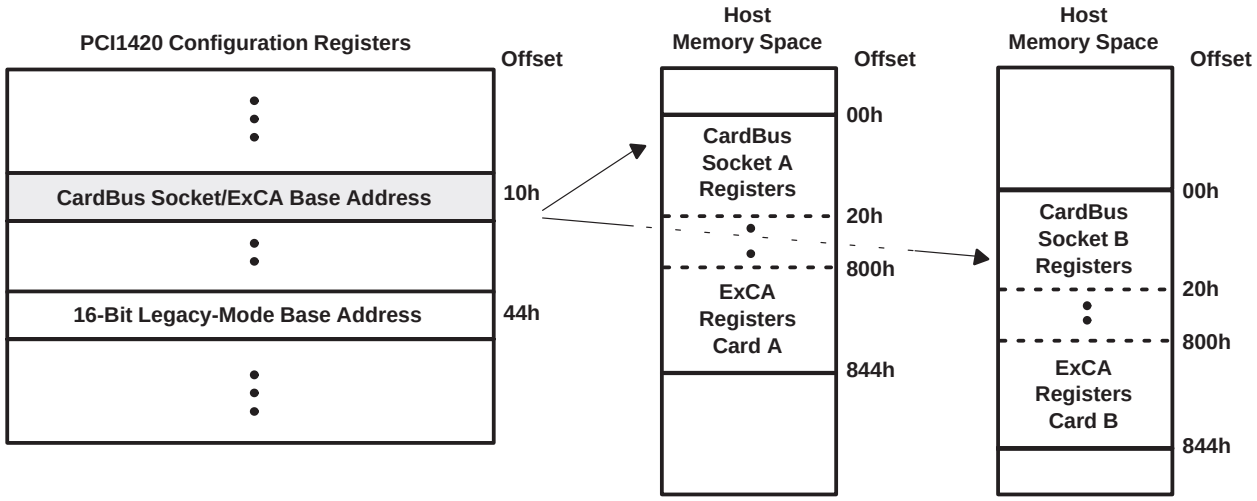
Bit	7	6	5	4	3	2	1	0
Name	ExCA memory windows 0–4 page							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0

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

6 CardBus Socket Registers (Functions 0 and 1)

The *1997 PC Card Standard* requires a CardBus socket controller to provide five 32-bit registers that report and control socket-specific functions. The PCI1420 provides the CardBus socket/ExCA base address register (see Section 4.12) to locate these CardBus socket registers in PCI memory address space. Each socket has a separate base address register for accessing the CardBus socket registers (see Figure 6–1). Table 6–1 gives the location of the socket registers in relation to the CardBus socket/ExCA base address.

The PCI1420 implements an additional register at offset 20h that provides power management control for the socket.



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

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

Table 6–1. CardBus Socket Registers

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

6.1 Socket Event Register

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

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

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

Table 6–2. Socket Event Register

BIT	SIGNAL	TYPE	FUNCTION
31–4	RSVD	R	Reserved. Bits 31–4 return 0s when read.
3	PWREVENT	R/C	Power cycle. Bit 3 is set when the PCI1420 detects that bit 3 (PWRCYCLE) in the socket present state register (see Section 6.3) has changed state. This bit is cleared by writing a 1.
2	CD2EVENT	R/C	$\overline{\text{CCD2}}$. Bit 2 is set when the PCI1420 detects that bit 2 (CDETECT2) in the socket present state register (see Section 6.3) has changed state. This bit is cleared by writing a 1.
1	CD1EVENT	R/C	$\overline{\text{CCD1}}$. Bit 3 is set when the PCI1420 detects that bit 1 (CDETECT1) in the socket present state register (see Section 6.3) has changed state. This bit is cleared by writing a 1.
0	CSTSEVENT	R/C	$\overline{\text{CSTSCHG}}$. Bit 0 is set when bit 0 (CARDSTS) in the socket present state register (see Section 6.3) has changed state. For CardBus cards, bit 0 is set on the rising edge of $\overline{\text{CSTSCHG}}$. For 16-bit PC Cards, bit 0 is set on both transitions of $\overline{\text{CSTSCHG}}$. This bit is reset by writing a 1.

6.2 Socket Mask Register

The socket mask register allows software to control the CardBus card events that generate a status change interrupt. The state of these mask bits does not prevent the corresponding bits from reacting in the socket event register (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	R/W	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Table 6–3. Socket Mask Register

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

6.3 Socket Present State Register

The socket present state register reports information about the socket interface. Write transactions to the socket force event register (see Section 6.4) are reflected here, as well as general socket interface status. Information about PC Card V_{CC} support and card type is only updated at each insertion. Also note that the PCI1420 uses $\overline{CCD1}$ and $\overline{CCD2}$ during card identification, and changes on these signals during this operation are not reflected in this register. See Table 6–4 for a complete description of the register contents.

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

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

Table 6–4. Socket Present State Register

BIT	SIGNAL	TYPE	FUNCTION
31	YVSOCKET	R	YV socket. Bit 31 indicates whether or not the socket can supply V_{CC} = Y.Y V to PC Cards. The PCI1420 does not support Y.Y-V V_{CC}; therefore, this bit is always reset unless overridden by the socket force event register (see Section 6.4). This bit is hardwired to 0.
30	XVSOCKET	R	XV socket. Bit 30 indicates whether or not the socket can supply V_{CC} = X.X V to PC Cards. The PCI1420 does not support X.X-V V_{CC}; therefore, this bit is always reset unless overridden by the socket force event register (see Section 6.4). This bit is hardwired to 0.
29	3VSOCKET	R	3-V socket. Bit 29 indicates whether or not the socket can supply V_{CC} = 3.3 V to PC Cards. The PCI1420 does support 3.3-V V_{CC}; therefore, this bit is always set unless overridden by the socket force event register (see Section 6.4).
28	5VSOCKET	R	5-V socket. Bit 28 indicates whether or not the socket can supply V_{CC} = 5 V to PC Cards. The PCI1420 does support 5-V V_{CC}; therefore, this bit is always set unless overridden by the socket force event register (see Section 6.4).
27–14	RSVD	R	Reserved. Bits 27–14 return 0s when read.
13	YVCARD	R	YV card. Bit 13 indicates whether or not the PC Card inserted in the socket supports V_{CC} = Y.Y V.
12	XVCARD	R	XV card. Bit 12 indicates whether or not the PC Card inserted in the socket supports V_{CC} = X.X V.
11	3VCARD	R	3-V card. Bit 11 indicates whether or not the PC Card inserted in the socket supports V_{CC} = 3.3 V.
10	5VCARD	R	5-V card. Bit 10 indicates whether or not the PC Card inserted in the socket supports V_{CC} = 5 V.
9	BADVCCREQ	R	Bad V_{CC} request. Bit 9 indicates that the host software has requested that the socket be powered at an invalid voltage. 0 = Normal operation (default) 1 = Invalid V_{CC} request by host software
8	DATALOST	R	Data lost. Bit 8 indicates that a PC Card removal event may have caused lost data because the cycle did not terminate properly or because write data still resides in the PCI1420. 0 = Normal operation (default) 1 = Potential data loss due to card removal
7	NOTACARD	R	Not a card. Bit 7 indicates that an unrecognizable PC Card has been inserted in the socket. This bit is not updated until a valid PC Card is inserted into the socket. 0 = Normal operation (default) 1 = Unrecognizable PC Card detected

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

BIT	SIGNAL	TYPE	FUNCTION

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

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

6.4 Socket Force Event Register

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

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

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

Table 6–5. Socket Force Event Register

BIT	SIGNAL	TYPE	FUNCTION
31–15	RSVD	R	Reserved. Bits 31–15 return 0s when read.
14	CVSTEST	W	Card VS test. When bit 14 is set, the PCI1420 re-interrogates the PC Card, updates the socket present state register (see Section 6.3), and enables the socket control register (see Section 6.5).
13	FYVCARD	W	Force YV card. Write transactions to bit 13 cause bit 13 (YVCARD) in the socket present state register to be written (see Section 6.3). When set, this bit disables the socket control register (see Section 6.5).
12	FXVCARD	W	Force XV card. Write transactions to bit 12 cause bit 12 (XVCARD) in the socket present state register to be written (see Section 6.3). When set, this bit disables the socket control register (see Section 6.5).
11	F3VCARD	W	Force 3-V card. Write transactions to bit 11 cause bit 11 (3VCARD) in the socket present state register to be written (see Section 6.3). When set, this bit disables the socket control register (see Section 6.5).
10	F5VCARD	W	Force 5-V card. Write transactions to bit 10 cause bit 10 (5VCARD) in the socket present state register to be written (see Section 6.3). When set, this bit disables the socket control register (see Section 6.5).
9	FBADVCCREQ	W	Force bad V_{CC} request. Changes to bit 9 (BADVCCREQ) in the socket present state register (see Section 6.3) can be made by writing to bit 9.
8	FDATALOST	W	Force data lost. Write transactions to bit 8 cause bit 8 (DATALOST) in the socket present state register to be written (see Section 6.3).
7	FNOTACARD	W	Force not a card. Write transactions to bit 7 cause bit 7 (NOTACARD) in the socket present state register to be written (see Section 6.3).
6	RSVD	R	Reserved. Bit 6 returns 0 when read.
5	FCBCARD	W	Force CardBus card. Write transactions to bit 5 cause bit 5 (CBCARD) in the socket present state register to be written (see Section 6.3).
4	F16BITCARD	W	Force 16-bit card. Write transactions to bit 4 cause bit 4 (16BITCARD) in the socket present state register to be written (see Section 6.3).
3	FPWRCYCLE	W	Force power cycle. Write transactions to bit 3 cause bit 3 (PWREVENT) in the socket event register to be written (see Section 6.1), and bit 3 (PWRCYCLE) in the socket present state register is unaffected (see Section 6.3).
2	FCDETECT2	W	Force $\overline{\text{CCD2}}$. Write transactions to bit 2 cause bit 2 (CD2EVENT) in the socket event register to be written (see Section 6.1), and bit 2 (CDETECT2) in the socket present state register is unaffected (see Section 6.3).
1	FCDETECT1	W	Force $\overline{\text{CCD1}}$. Write transactions to bit 1 cause bit 1 (CD1EVENT) in the socket event register to be written (see Section 6.1), and bit 1 (CDETECT1) in the socket present state register is unaffected (see Section 6.3).
0	FCARDSTS	W	Force CSTSCHG. Write transactions to bit 0 cause bit 0 (CSTSEVENT) in the socket event register to be written (see Section 6.1), and bit 0 (CARDSTS) in the socket present state register is unaffected (see Section 6.3).

6.5 Socket Control Register

The socket control register provides control of the voltages applied to the socket and instructions for CB $\overline{\text{CLKRUN}}$ protocol. The PCI1420 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	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Table 6–6. Socket Control Register

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

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

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

Table 6–7. Socket Power Management Register

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

7 Distributed DMA (DDMA) Registers

The DMA base address, programmable in PCI configuration space at offset 98h, points to a 16-byte region in PCI I/O space where the DDMA registers reside. The names and locations of these registers are summarized in Table 7–1. These PCI1420 register definitions are identical in function, but differ in location, to the 8237 DMA controller. The similarity between the register models retains some level of compatibility with legacy DMA and simplifies the translation required by the master DMA device when it forwards legacy DMA writes to DMA channels.

While the DMA register definitions are identical to those in the 8237 of the same name, some register bits defined in the 8237 do not apply to distributed DMA in a PCI environment. In such cases, the PCI1420 implements these obsolete register bits as read-only nonfunctional bits. The reserved registers shown in Table 7–1 are implemented as read-only and return 0s when read. Write transactions to reserved registers have no effect.

Table 7–1. Distributed DMA Registers

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

7.1 DMA Current Address/Base Address Register

The DMA current address/base address register sets the starting (base) memory address of a DMA transfer. Read transactions from this register indicate the current memory address of a direct memory transfer.

For the 8-bit DMA transfer mode, the current address register contents are presented on AD15–AD0 of the PCI bus during the address phase. Bits 7–0 of the DMA page register (see Section 7.2) are presented on AD23–AD16 of the PCI bus during the address phase.

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

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

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

7.2 DMA Page Register

The DMA page register sets the upper byte of the address of a DMA transfer. Details of the address represented by this register are explained in Section 7.1, *DMA Current Address/Base Address Register*.

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

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

7.3 DMA Current Count/Base Count Register

The DMA current count/base count register sets the total transfer count, in bytes, of a direct memory transfer. Read transactions to this register indicate the current count of a direct memory transfer. In the 8-bit transfer mode, the count is decremented by 1 after each transfer, and the count is decremented by 2 after each transfer in the 16-bit transfer mode.

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

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

7.4 DMA Command Register

The DMA command register enables and disables the DMA controller. Bit 2 (DMAEN) defaults to 0 enabling the DMA controller. All other bits are reserved. See Table 7–2 for a complete description of the register contents.

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

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

Table 7–2. DMA Command Register

BIT	TYPE	TYPE	FUNCTION
7–3	RSVD	R	Reserved. Bits 7–3 return 0s when read.
2	DMAEN	R/W	DMA controller enable. Bit 2 enables and disables the distributed DMA slave controller in the PCI1420 and defaults to the enabled state. 0 = DMA controller enabled (default) 1 = DMA controller disabled
1–0	RSVD	R	Reserved. Bits 1 and 0 return 0s when read.

7.5 DMA Status Register

The DMA status register indicates the terminal count and DMA request ($\overline{\text{DREQ}}$) status. See Table 7–3 for a complete description of the register contents.

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

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

Table 7–3. DMA Status Register

BIT	SIGNAL	TYPE	FUNCTION
7–4	DREQSTAT	R	Channel request. In the 8237, bits 7–4 indicate the status of $\overline{\text{DREQ}}$ of each DMA channel. In the PCI1420, these bits indicate the $\overline{\text{DREQ}}$ status of the single socket being serviced by this register. All four bits are set when the PC Card asserts $\overline{\text{DREQ}}$ and are reset when $\overline{\text{DREQ}}$ is deasserted. The status of bit 0 (MASKBIT) in the DMA multichannel/mask register (see Section 7.9) has no effect on these bits.
3–0	TC	R	Channel terminal count. The 8327 uses bits 3–0 to indicate the TC status of each of its four DMA channels. In the PCI1420, these bits report information about a single DMA channel; therefore, all four of these register bits indicate the TC status of the single socket being serviced by this register. All four bits are set when the TC is reached by the DMA channel. These bits are reset when read or the DMA channel is reset.

7.6 DMA Request Register

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

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

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

7.7 DMA Mode Register

The DMA mode register sets the DMA transfer mode. See Table 7–4 for a complete description of the register contents.

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

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

Table 7–4. DMA Mode Register

BIT	SIGNAL	TYPE	FUNCTION
7–6	DMAMODE	R/W	Mode select. The PCI1420 uses bits 7 and 6 to determine the transfer mode. 00 = Demand mode select (default) 01 = Single mode select 10 = Block mode select 11 = Reserved
5	INCDEC	R/W	Address increment/decrement. The PCI1420 uses bit 5 to select the memory address in the DMA current address/base address register to increment or decrement after each data transfer. This is in accordance with the 8237 use of this register bit and is encoded as follows: 0 = Addresses increment (default). 1 = Addresses decrement.
4	AUTOINIT	R/W	Auto initialization 0 = Auto initialization disabled (default) 1 = Auto initialization enabled
3–2	XFERTYPE	R/W	Transfer type. Bits 3 and 2 select the type of direct memory transfer to be performed. A memory write transfer moves data from the PCI1420 PC Card interface to memory and a memory read transfer moves data from memory to the PCI1420 PC Card interface. The field is encoded as: 00 = No transfer selected (default) 01 = Write transfer 10 = Read transfer 11 = Reserved
1–0	RSVD	R	Reserved. Bits 1 and 0 return 0s when read.

7.8 DMA Master Clear Register

The DMA master clear register resets the DDMA controller and all DDMA registers.

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

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

7.9 DMA Multichannel/Mask Register

The PCI1420 uses only the least significant bit of this register to mask the PC Card DMA channel. The PCI1420 sets the mask bit when the PC Card is removed. Host software is responsible for either resetting the socket's DMA controller or enabling the mask bit. See Table 7–5 for a complete description of the register contents.

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

Register: **DMA multichannel/mask**
Type: Read-only
Offset: DMA base address + 0Fh
Default: 00h
Size: One byte

Table 7–5. DMA Multichannel/Mask Register

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

8 Electrical Characteristics

8.1 Absolute Maximum Ratings Over Operating Temperature Ranges†

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

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

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

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

8.2 Recommended Operating Conditions (see Note 3)

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

† Applies to external inputs and bidirectional buffers without hysteresis

‡ Miscellaneous pins are 149, 150, 151, 152, 154, 155, 156, 157, 158, 159, 161, and 163 for the PDV packaged device and A16, B15, C14, C15, D19, E14, E17, E19, F14, F15, F17, and G15 for the GHK packaged device ($\overline{\text{SUSPEND}}$, SPKROUT, $\overline{\text{RI\_OUT}}$, multifunction terminals (MFUNC0–MFUNC6), and power switch control pins).

§ Fail-safe pins are 16, 56, 68, 74, 82, 122, 134, and 140 for the PDV packaged device and H3, H17, J18, M19, P7, R9, U8, and V11 for the GHK packaged device (card detect and voltage sense pins).

¶ Applies to external output buffers

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

☆CD pins are 16, 74, 82, and 140 for the PDV packaged device and H3, H17, R9, and V11 for the GHK packaged device.

NOTE 3: Unused pins (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		PINS	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	
		$\overline{SERR}$		I_{OL} = 12 mA		0.5	
I_{OZL}	3-state, high-impedance low-level output current	Output pins	3.6 V	$V_I = V_{CC}$		–1	µA
			5.25 V	$V_I = V_{CC}$		–1	
I_{OZH}	3-state, high-impedance high-level output current	Output pins	3.6 V	$V_I = V_{CC}$†		10	µA
			5.25 V	$V_I = V_{CC}$†		25	
I_{IL}	Low-level input current	Input pins		V_I = GND		–1	µA
		I/O pins		V_I = GND		–10	
I_{IH}§	High-level input current	Input pins	3.6 V	$V_I = V_{CC}$‡		10	µA
			5.25 V	$V_I = V_{CC}$‡		20	
		I/O pins	3.6 V	$V_I = V_{CC}$‡		10	
			5.25 V	$V_I = V_{CC}$‡		25	
		Fail-safe pins	3.6 V	$V_I = V_{CC}$		10	

† For PCI pins, $V_I = V_{CCP}$. For PC Card pins, $V_I = V_{CC(A/B)}$. For miscellaneous pins, $V_I = V_{CCI}$

‡ For I/O pins, input leakage (I_{IL} and I_{IH}) includes I_{OZ} leakage of the disabled output.

§ I_{IH} is not tested in these pins: 16, 43, 45, 47, 48, 49, 50, 56, 58, 61, 68, 69, 70, 71, 72, 74, 82, 107, 108, 109, 111, 114, 115, 122, 124, 127, 134, 135, 136, 137, 138, 140, and 150 for the PDV packaged device and F17, H3, H17, H19, J14, J15, J17, J18, L14, L18, M14, M19, N5, N19, P1, P5, P6, P7, P15, P17, P19, R1, R2, R7, R9, R18, U8, V8, V9, V11, W5, W8, and W9 for the GHK packaged device because they are pulled up with internal resistors.

8.4 PCI Clock/Reset Timing Requirements Over Recommended Ranges Of Supply Voltage And Operating Free-air Temperature

PARAMETER		ALTERNATE SYMBOL	TEST CONDITIONS	MIN	MAX	UNIT
t_c	Cycle time, PCLK	t_{cyc}		30		ns
t_{wH}	Pulse duration (width), PCLK high	t_{high}		11		ns
t_{wL}	Pulse duration (width), PCLK low	t_{low}		11		ns
$\Delta v/\Delta t$	Slew rate, PCLK	t_r, t_f		1	4	V/ns
t_w	Pulse duration (width), 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

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

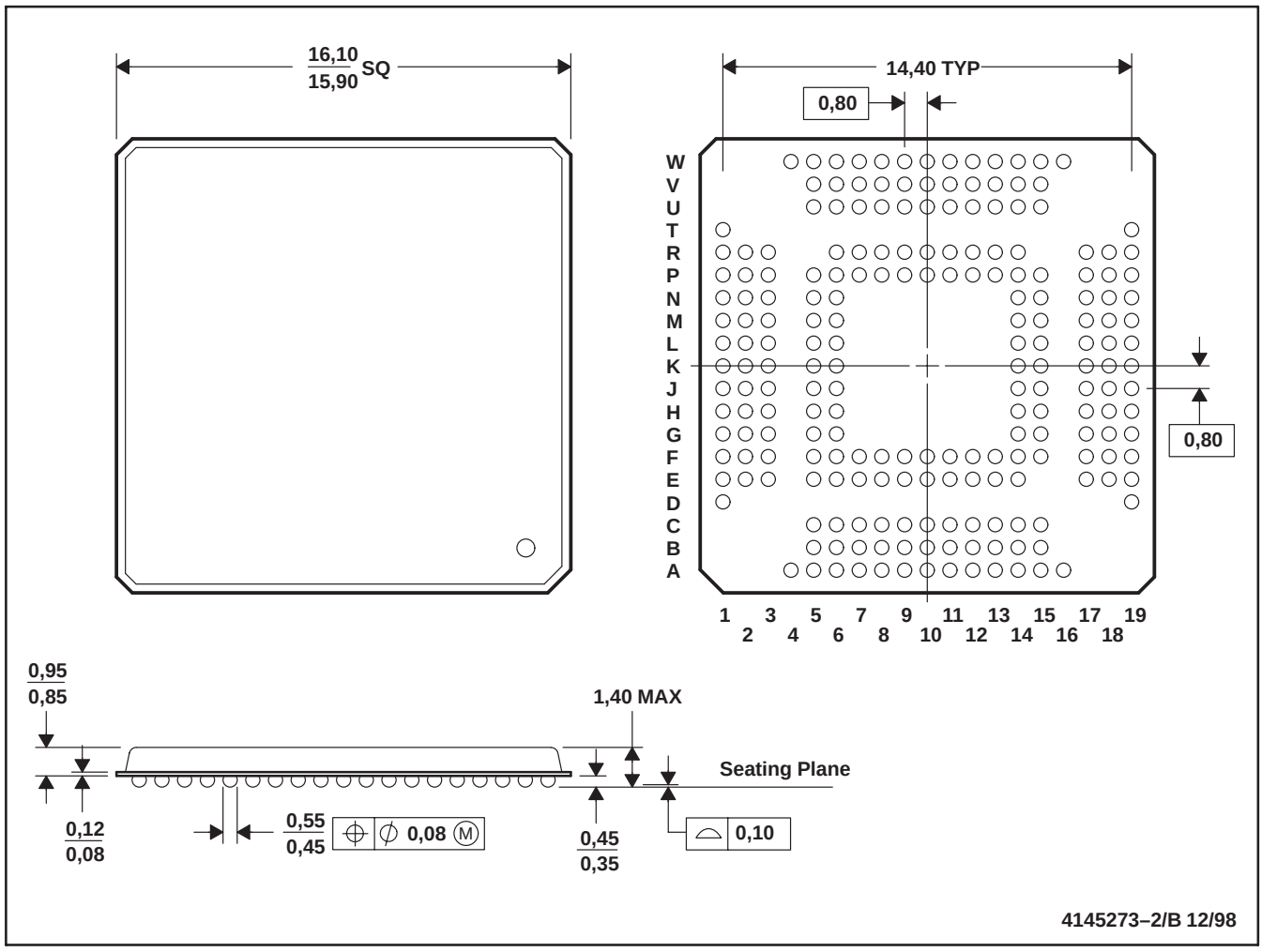
PARAMETER			ALTERNATE SYMBOL	TEST CONDITIONS	MIN	MAX	UNIT
t_{pd}	Propagation delay time, See Note 4	PCLK-to-shared signal valid delay time	t_{val}	C_L = 50 pF, See Note 4		11	ns
		PCLK-to-shared signal invalid delay time	t_{inv}		2		
t_{en}	Enable time, high impedance-to-active delay time from PCLK	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

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

9 Mechanical Information

The PCI1420 is packaged in either a 209-ball GHK BGA or a 208-pin PDV package. The following shows the mechanical dimensions for the GHK and PDV packages.

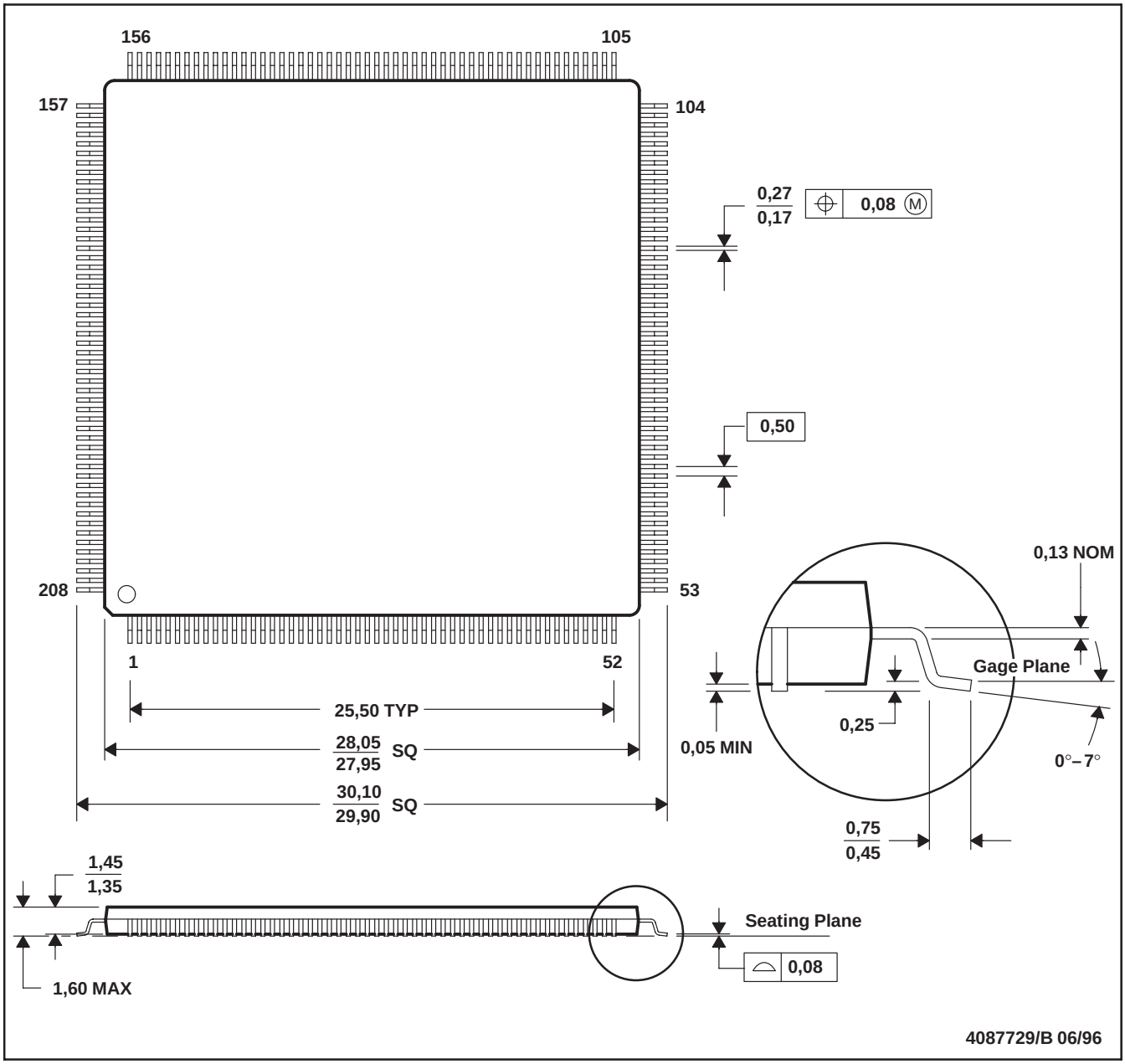
GHK (S-PBGA-N209) PLASTIC BALL GRID ARRAY



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

Micro Star is a trademark of Texas Instruments Incorporated.

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 MO-136

IMPORTANT NOTICE

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

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

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

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

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