

PCI1515Single Socket CardBus Controller

Data Manual

IMPORTANT NOTICE

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

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

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

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

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

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

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

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

Mailing Address: Texas Instruments

Post Office Box 655303 Dallas, Texas 75265

Copyright © 2004, Texas Instruments Incorporated

Contents

Se	ection		Title	Page
1	Introd	duction .		1–1
	1.1	Control	ler Functional Description	1–1
		1.1.1	PCI1515 Controller	1–1
		1.1.2	Multifunctional Terminals	1–1
		1.1.3	PCI Bus Power Management	1–1
		1.1.4	Power Switch Interface	1–1
	1.2	Feature	es	1–1
	1.3	Related	Documents	1–2
	1.4	Tradem	narks	1–2
	1.5	Terms a	and Definitions	1–3
	1.6	Orderin	g Information	1–3
2			criptions	2–1
3	Featu		col Descriptions	
	3.1	Power :	Supply Sequencing	3–1
	3.2	I/O Cha	aracteristics	3–1
	3.3		ng Voltages	
	3.4	Periphe	eral Component Interconnect (PCI) Interface	3–2
		3.4.1	Device Resets	
		3.4.2	PCI Bus Lock (LOCK)	3–2
		3.4.3	Serial EEPROM I ² C Bus	
		3.4.4	Function 0 (CardBus) Subsystem Identification	
	3.5	PC Car	d Applications	
		3.5.1	PC Card Insertion/Removal and Recognition	3–4
		3.5.2	Low Voltage CardBus Card Detection	
		3.5.3	Card Detection	
		3.5.4	Power Switch Interface	
		3.5.5	Internal Ring Oscillator	
		3.5.6	Integrated Pullup Resistors for PC Card Interface	
		3.5.7	SPKROUT and CAUDPWM Usage	
		3.5.8	LED Socket Activity Indicators	
		3.5.9	CardBus Socket Registers	3–8
	3.6		EEPROM Interface	3–8
		3.6.1	Serial-Bus Interface Implementation	3–8
		3.6.2	Accessing Serial-Bus Devices Through Software	3–8
		3.6.3	Serial-Bus Interface Protocol	3–8
		3.6.4	Serial-Bus EEPROM Application	3–10

ection		Title	Page
3.7	Program	mable Interrupt Subsystem	3–12
	3.7.1	·	
	3.7.2		
	3.7.3		
	3.7.4	Using Parallel PCI Interrupts	3–14
	3.7.5	· · · · · · · · · · · · · · · · · · ·	
	3.7.6	SMI Support in the PCI1515 Controller	3–15
3.8	Power M	lanagement Overview	3–15
	3.8.1	Integrated Low-Dropout Voltage Regulator (LDO-VR)	3–16
	3.8.2	CardBus (Function 0) Clock Run Protocol	3–16
	3.8.3	CardBus PC Card Power Management	3–17
	3.8.4	16-Bit PC Card Power Management	3–17
	3.8.5	Suspend Mode	3–17
	3.8.6	Requirements for Suspend Mode	3–18
	3.8.7	Ring Indicate	3–18
	3.8.8		
	3.8.9	CardBus Bridge Power Management	3–20
	3.8.10	·	3–21
	3.8.11		0.04
DC Ca	ud Contu		
		-	
_		•	_
_			
_		· ·	
_	•	· · · · · · · · · · · · · · · · · · ·	
	•	,	
		•	
		· · · · · · · · · · · · · · · · · · ·	
		,	
	3.8	3.7 Program 3.7.1 3.7.2 3.7.3 3.7.4 3.7.5 3.7.6 3.8 Power M 3.8.1 3.8.2 3.8.3 3.8.4 3.8.5 3.8.6 3.8.7 3.8.8 3.8.9 3.8.10 3.8.11 PC Card Contr 4.1 PCI Cont 4.2 Vendor I 4.1 PCI Cont 4.2 Vendor I 4.3 Device I 4.4 Comman 4.5 Status R 4.6 Revision 4.7 Class Co 4.8 Cache L 4.9 Latency 4.10 Header 4.11 BIST Re 4.12 CardBus 4.13 Capabilit 4.14 Seconda 4.15 PCI Bus 4.16 CardBus 4.17 Subordir 4.18 CardBus 4.17 Subordir 4.18 CardBus 4.19 CardBus 4.20 CardBus	3.7.1 PC Card Functional and Card Status Change Interrupts 3.7.2 Interrupt Masks and Flags 3.7.3 Using Parallel IRQ Interrupts 3.7.4 Using Parallel IRQ Interrupts 3.7.5 Using Serialized IRQSER Interrupts 3.7.6 SMI Support in the PCI1515 Controller 3.8 Power Management Overview 3.8.1 Integrated Low-Dropout Voltage Regulator (LDO-VR) 3.8.2 CardBus (Function 0) Clock Run Protocol 3.8.3 CardBus PC Card Power Management 3.8.4 16-Bit PC Card Power Management 3.8.5 Suspend Mode 3.8.6 Requirements for Suspend Mode 3.8.7 Ring Indicate 3.8.8 PCI Power Management (Function 0) 3.8.9 CardBus Bridge Power Management 3.8.10 ACPI Support 3.8.11 Master List of PME Context Bits and Global Reset-Only Bits Bits PC Card Controller Programming Model 4.1 PCI Configuration Register Map (Function 0) 4.2 Vendor ID Register 4.3 Device ID Register Function 0 4.4 Command Register 4.5 Status Register 4.6 Revision ID Register 4.7 Class Code Register 4.8 Cache Line Size Register 4.9 Latency Timer Register 4.10 Header Type Register 4.11 BIST Register 4.12 CardBus Socket Registers/ExCA Base Address Register 4.13 Capability Pointer Register 4.14 Secondary Status Register 4.15 PCI Bus Number Register 4.16 CardBus Bus Number Register 4.17 Subordinate Bus Number Register 4.18 CardBus Latency Timer Register 4.19 CardBus Memory Base Registers 0, 1

Se	ction	Title	Page
	4.22	CardBus I/O Limit Registers 0, 1	4–13
	4.23	Interrupt Line Register	
	4.24	Interrupt Pin Register	
	4.25	Bridge Control Register	4–15
	4.26	Subsystem Vendor ID Register	4–16
	4.27	Subsystem ID Register	
	4.28	PC Card 16-Bit I/F Legacy-Mode Base-Address Register	4-16
	4.29	System Control Register	4-17
	4.30	General Control Register	4-19
	4.31	General-Purpose Event Status Register	4-20
	4.32	General-Purpose Event Enable Register	4–21
	4.33	General-Purpose Input Register	4–21
	4.34	General-Purpose Output Register	4–22
	4.35	Multifunction Routing Status Register	4–23
	4.36	Retry Status Register	4–24
	4.37	Card Control Register	
	4.38	Device Control Register	4–26
	4.39	Diagnostic Register	
	4.40	Capability ID Register	
	4.41	Next Item Pointer Register	
	4.42	Power Management Capabilities Register	
	4.43	Power Management Control/Status Register	4–30
	4.44	Power Management Control/Status Bridge Support Extensions	1 21
	4.45	Register	
	4.46	Serial Bus Data Register	
	4.47	Serial Bus Index Register	
	4.48	Serial Bus Slave Address Register	
	4.49	Serial Bus Control/Status Register	
5		Compatibility Registers (Function 0)	
•	5.1	ExCA Identification and Revision Register	
	5.2	ExCA Interface Status Register	5–6
	5.3	ExCA Power Control Register	5–7
	5.4	ExCA Interrupt and General Control Register	5–8
	5.5	ExCA Card Status-Change Register	5–9
	5.6	ExCA Card Status-Change Interrupt Configuration Register	
	5.7	ExCA Address Window Enable Register	
	5.8	ExCA I/O Window Control Register	
	5.9	ExCA I/O Windows 0 and 1 Start-Address Low-Byte Registers	
	5.10	ExCA I/O Windows 0 and 1 Start-Address High-Byte Registers	
	5.11	ExCA I/O Windows 0 and 1 End-Address Low-Byte Registers	
	5.12	ExCA I/O Windows 0 and 1 End-Address High-Byte Registers	

Se	ection	Title	Page
	5.13	ExCA Memory Windows 0–4 Start-Address Low-Byte Registers	5–14
	5.14	ExCA Memory Windows 0–4 Start-Address High-Byte Registers	5–15
	5.15	ExCA Memory Windows 0–4 End-Address Low-Byte Registers	5–16
	5.16	ExCA Memory Windows 0–4 End-Address High-Byte Registers	5–16
	5.17	ExCA Memory Windows 0–4 Offset-Address Low-Byte Registers	5–17
	5.18	ExCA Memory Windows 0–4 Offset-Address High-Byte Registers .	5–18
	5.19	ExCA Card Detect and General Control Register	5-19
	5.20	ExCA Global Control Register	5-20
	5.21	ExCA I/O Windows 0 and 1 Offset-Address Low-Byte Registers	5-21
	5.22	ExCA I/O Windows 0 and 1 Offset-Address High-Byte Registers	5-21
	5.23	ExCA Memory Windows 0–4 Page Registers	5-21
6	CardB	Bus Socket Registers (Function 0)	6–1
	6.1	Socket Event Register	6–2
	6.2	Socket Mask Register	6–3
	6.3	Socket Present State Register	6–4
	6.4	Socket Force Event Register	6–5
	6.5	Socket Control Register	6–7
	6.6	Socket Power Management Register	6–8
7	Electr	ical Characteristics	7–1
	7.1	Absolute Maximum Ratings Over Operating Temperature Ranges .	7–1
	7.2	Recommended Operating Conditions	7–1
	7.3	Electrical Characteristics Over Recommended Operating Conditions	7–3
	7.4	PCI Clock/Reset Timing Requirements Over Recommended Ranges Supply Voltage and Operating Free-Air Temperature	of 7–3
	7.5	PCI Timing Requirements Over Recommended Ranges of Supply Voltage and Operating Free-Air Temperature	7–4
	7.6	Reset Timing	7–4
8	Mecha	anical Information	8–1

List of Illustrations

Figure	Title	Page
2–1	PCI1515 GHK/ZHK-Package Terminal Diagram	2–1
3–1	PCI1515 System Block Diagram	3–1
3–2	3-State Bidirectional Buffer	3–2
3–3	Serial ROM Application	3–3
3–4	SPKROUT Connection to Speaker Driver	3–7
3–5	Sample LED Circuit	3–7
3–6	Serial-Bus Start/Stop Conditions and Bit Transfers	3-9
3–7	Serial-Bus Protocol Acknowledge	3–9
3–8	Serial-Bus Protocol—Byte Write	3–10
3–9	Serial-Bus Protocol—Byte Read	3–10
3–10	EEPROM Interface Doubleword Data Collection	3–10
3–11	IRQ Implementation	3–14
3–12	System Diagram Implementing CardBus Device Class Power	
	Management	3–16
3–13	Signal Diagram of Suspend Function	3–18
3–14	RI_OUT Functional Diagram	3–19
3–15	Block Diagram of a Status/Enable Cell	3–21
5–1	ExCA Register Access Through I/O	5–2
5–2	ExCA Register Access Through Memory	5–2
6–1	Accessing CardBus Socket Registers Through PCI Memory	6–1
7–1	Reset Timing Diagram	7–4

List of Tables

Table	Title	Page
1–1	Terms and Definitions	1–3
2–1	Signal Names by GHK Terminal Number	
2-2	CardBus PC Card Signal Names Sorted Alphabetically	
2-3	16-Bit PC Card Signal Names Sorted Alphabetically	2–7
2-4	Power Supply Terminals	2-9
2-5	PC Card Power Switch Terminals	
2-6	PCI System Terminals	2-9
2-7	PCI Address and Data Terminals	
2-8	PCI Interface Control Terminals	2-11
2-9	Multifunction and Miscellaneous Terminals	2–12
2-10	16-Bit PC Card Address and Data Terminals	2-13
2-11	16-Bit PC Card Interface Control Terminals	2-14
2–12	CardBus PC Card Interface System Terminals	2–15
2–13	CardBus PC Card Address and Data Terminals	2-16
2-14	CardBus PC Card Interface Control Terminals	2-17
3–1	PC Card—Card Detect and Voltage Sense Connections	3–5
3–2	TPS2228 Control Logic—xVPP/VCORE	3–6
3–3	TPS2228 Control Logic—xVCC	3–6
3–4	TPS2226A Control Logic—xVPP	3–6
3–5	TPS2226A Control Logic—xVCC	3–6
3–6	CardBus Socket Registers	3–8
3–7	PCI1515 Registers Used to Program Serial-Bus Devices	3–8
3–8	EEPROM Loading Map	3–11
3–9	Interrupt Mask and Flag Registers	3–12
3–10	PC Card Interrupt Events and Description	3–13
3–11	SMI Control	
3–12	Requirements for Internal/External 1.5-V Core Power Supply	3–16
3–13	Power-Management Registers	3–20
4–1	Bit Field Access Tag Descriptions	
4–2	Function 0 PCI Configuration Register Map	
4–3	Command Register Description	
4–4	Status Register Description	
4–5	Secondary Status Register Description	
4–6	Bridge Control Register Description	
4–7	System Control Register Description	
4–8	General Control Register Description	
4-9	General-Purpose Event Status Register Description	4-20

Table	Title	Page
4-10	General-Purpose Event Enable Register Description	4-21
4-11	General-Purpose Input Register Description	4-21
4-12	General-Purpose Output Register Description	4-22
4-13	Multifunction Routing Status Register Description	4-23
4–14	Retry Status Register Description	4-24
4-15	Card Control Register Description	4-25
4–16	Device Control Register Description	4-26
4–17	Diagnostic Register Description	4-27
4–18	Power Management Capabilities Register Description	4-29
4-19	Power Management Control/Status Register Description	4-30
4–20	Power Management Control/Status Bridge Support Extensions Register Description	
4-21	Serial Bus Data Register Description	
4-22	Serial Bus Index Register Description	
4-23	Serial Bus Slave Address Register Description	
4-24	Serial Bus Control/Status Register Description	
5–1	ExCA Registers and Offsets	
5–2	ExCA Identification and Revision Register Description	5–5
5–3	ExCA Interface Status Register Description	5–6
5-4	ExCA Power Control Register Description—82365SL Support	5–7
5–5	ExCA Power Control Register Description—82365SL-DF Support	5–7
5–6	ExCA Interrupt and General Control Register Description	5–8
5–7	ExCA Card Status-Change Register Description	5–9
5–8	ExCA Card Status-Change Interrupt Configuration Register	
	Description	
5–9	ExCA Address Window Enable Register Description	
5–10	ExCA I/O Window Control Register Description	5–12
5–11	ExCA Memory Windows 0–4 Start-Address High-Byte Registers Description	5–15
5–12	ExCA Memory Windows 0–4 End-Address High-Byte Registers Description	5–16
5–13	ExCA Memory Windows 0–4 Offset-Address High-Byte Registers Description	
5–14	ExCA Card Detect and General Control Register Description	5–19
5–15	ExCA Global Control Register Description	5–20
6–1	CardBus Socket Registers	6–1
6–2	Socket Event Register Description	6–2
6–3	Socket Mask Register Description	6–3
6–4	Socket Present State Register Description	6–4
6–5	Socket Force Event Register Description	6–6
6–6	Socket Control Register Description	6–7
6–7	Socket Power Management Register Description	6–8

1 Introduction

The Texas Instruments PCI1515 controller is a single-socket PC Card controller. This high-performance solution provides the latest in PC Card technology.

1.1 Controller Functional Description

1.1.1 PCI1515 Controller

The PCI1515 controller is a single-function PCI controller compliant with PCI Local Bus Specification, Revision 2.3.

Function 0 provides a PC Card socket controller compliant with the *PC Card Standard* (Release 8.1). The PCI1515 controller provides features that make it the best choice for bridging between the PCI bus and PC Cards, and supports 16-bit, CardBus, or USB custom card interface PC Cards, powered at 5 V or 3.3 V, as required.

All card signals are internally buffered to allow hot insertion and removal without external buffering. The PCI1515 controller is register compatible with the Intel 82365SL-DF ExCA controller. The PCI1515 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 PCI1515 controller can be programmed to accept posted writes to improve bus utilization.

1.1.2 Multifunctional Terminals

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

1.1.3 PCI Bus Power Management

The PCI1515 controller is compliant with the latest *PCI Bus Power Management Specification*, and provides several low-power modes, which enable the host power system to further reduce power consumption.

1.1.4 Power Switch Interface

The PCI1515 controller also has a three-pin serial interface compatible with the Texas Instruments TPS2228 (default), TPS2226A, TPS2224A, TPS2223A, and TPS2220A power switches. All five power switches provide power to the CardBus socket on the PCI1515 controller.

1.2 Features

The PCI1515 controller supports the following features:

- PC Card Standard 8.1 compliant
- PCI Bus Power Management Interface Specification 1.1 compliant
- Advanced Configuration and Power Interface (ACPI) Specification 2.0 compliant
- PCI Local Bus Specification Revision 2.3 compliant
- PC 98/99 and PC2001 compliant
- Windows Logo Program 2.0 compliant

- PCI Bus Interface Specification for PCI-to-CardBus Bridges
- 1.5-V core logic and 3.3-V I/O cells with internal voltage regulator to generate 1.5-V core V_{CC}
- Universal PCI interfaces compatible with 3.3-V and 5-V PCI signaling environments
- Supports PC Card or CardBus with hot insertion and removal
- Supports 132-MBps burst transfers to maximize data throughput on both the PCI bus and the CardBus
- Supports serialized IRQ with PCI interrupts
- Programmable multifunction terminals
- Many interrupt modes supported
- Serial ROM interface for loading subsystem ID and subsystem vendor ID
- ExCA-compatible registers are mapped in memory or I/O space
- Intel 82365SL-DF register compatible
- Supports ring indicate, SUSPEND, and PCI CLKRUN protocols and PCI bus Lock (LOCK)
- Provides VGA/palette memory and I/O, and subtractive decoding options, LED activity terminals
- Compliant with Intel Mobile Power Guideline 2000
- PCI power-management D0, D1, D2, and D3 power states
- Advanced submicron, low-power CMOS technology

1.3 Related Documents

- Advanced Configuration and Power Interface (ACPI) Specification (Revision 2.0)
- PC Card Standard (Release 8.1)
- PCI Bus Power Management Interface Specification (Revision 1.1)
- Serial Bus Protocol 2 (SBP-2)
- Serialized IRQ Support for PCI Systems
- PCI Mobile Design Guide
- PCI Bus Power Management Interface Specification for PCI to CardBus Bridges
- PCI14xx Implementation Guide for D3 Wake-Up
- PCI to PCMCIA CardBus Bridge Register Description
- Texas Instruments TPS2220A, TPS2223A, TPS2224A, and TPS2226A product data sheet, SLVS428A
- Texas Instruments TPS2228 product data sheet, SLVS419
- PCI Local Bus Specification (Revision 2.3)
- PCMCIA Proposal (262)
- ISO Standards for Identification Cards ISO/IEC 7816

1.4 Trademarks

Intel is a trademark of Intel Corporation.

TI and MicroStar BGA are trademarks of Texas Instruments.

Other trademarks are the property of their respective owners.

1.5 Terms and Definitions

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

Table 1-1. Terms and Definitions

TERM	DEFINITIONS
AT	AT (advanced technology, as in PC AT) attachment interface
CIS	Card information structure. Tuple list defined by the PC Card standard to communicate card information to the host computer.
CSR	Control and status register
PCMCIA	Personal Computer Memory Card International Association. Standards body that governs the PC Card standards.
RSVD	Reserved for future use

1.6 Ordering Information

ORDERING NUMBER	NAME	VOLTAGE	PACKAGE
PCI1515	Single Socket CardBus Controller	3.3-V, 5-V tolerant I/Os	257-ball PBGA (GHK or ZHK)

2 Terminal Descriptions

The PCI1515 controller is available in the 257-terminal MicroStar BGA™ package (GHK) or the 257-terminal lead-free (Pb, atomic number 82) MicroStar BGA™ package (ZHK). Figure 2–1 is a pin diagram of the PCI1515 package.

w		NC	NC	AD16	TRDY	SERR	AD15	VCCP	AD11	C/BE0	AD4	TEST3	NC	NC	NC	NC	NC	NC	
٧	NC	NC	NC	NC	ĪRDY	STOP	C/BE1	AD12	AD10	AD7	AD3	TEST2	NC	NC	NC	NC	NC	NC	NC
U	NC	NC	NC	NC	C/BE2	DEVSEL	PAR	AD13	AD9	AD6	AD2	TEST1	GND	GND	VCC	NC	NC	GND	VCC
Т	AD18	AD17	NC														NC	NC	NC
R	AD22	AD21	AD19			FRAME	PERR	AD14	AD8	AD5	AD0	RSVD	NC	GND			GND	NC	NC
Р	VCCP	C/BE3	AD23		AD20	VCC	GND	VCC	GND	VCC	AD1	TEST0	VCC	VCC	VR_ PORT		TEST4	NC	A_CAD0 //A_D3
N	AD26	AD25	AD24		IDSEL	GND								NC	A_CCD1		A_CAD2	A_CAD1	A_CAD4
М	AD31	AD30	AD29		AD27	AD28								GND	//A_CD1		//A_D11 A_CAD6	//A_D4 A_CAD5	//A_D12 A_RSVD
 L	PCLK	GNT	REQ		RI_OUT	vcc								vcc	//A_D5 A_CAD9		//A_D13 A_CC/BE0	//A_D6 A_CAD8	//A_D14 A_CAD7
					//PME										//A_A10 A CAD12		//A_CE1 A CAD11	//A_D15 A CAD10	//A_D7
K	VR_PORT	VR_EN	PRST		GRST	GND								GND	//A_A11		//A_OE A CAD15	//A_CE2	VR_PORT
J	MFUNC4	MFUNC5	MFUNC6		SUSPEND	VCC								VCC	//A_A9		//A_IOWR	//A_IORD	VCCA
Н	MFUNC3	MFUNC2	SPKROUT		MFUNC1	GND								A_CPAR //A_A13	A_CBLOCK //A_A19		A_RSVD //A_A18	A_CC/BE1 //A_A8	A_CAD16 //A_A17
G	MFUNC0	SCL	SDA		NC	RSVD							1	GND	A_CTRDY //A_A22		A_CGNT //A_WE	A_CSTOP //A_A20	A_CPERR //A_A14
F	RSVD	NC	NC		NC	VCC	GND	NC	vcc	GND	A_CAD29 //A_D1	vcc	GND	vcc	A_CAD17 //A_A24		A_CIRDY //A_A15	A_CCLK //A_A16	A_CDEVSEL //A_A21
Ε	NC	NC	NC		NC	NC	NC	NC	NC	A_USB_EN	A_CAD28 //A_D8	A_CINT// A_READY (IREQ)	A_CC/BE3 //A_REG	A_CAD21 //A_A5			A_CAD18 //A_A7	A_CC/BE2 //A_A12	A_CFRAME //A_A23
D	NC	NC	NC														NC	NC	A_CAD19 //A_A25
С	NC	NC	NC	NC	NC	NC	NC	NC	LATCH	A_CAD31 //A_D10	A_CAD27 //A_D0	A_CSERR //A_WAIT	A_CAD25 //A_A1	A_CREQ //A_INPACK	A_CRST //A_RESET	NC	NC	NC	NC
В	NC	NC	NC	NC	NC	NC	NC	NC	DATA	A_RSVD //A_D2	A_CCD2 //A_CD2	A_CAUDIO //A_BVD2 (SPKR)	A_CAD26 //A_A0	A_CAD23 //A_A3	A_CAD22 //A_A4	A_CVS2 //A_VS2	NC	NC	NC
Α		NC	NC	NC	NC	NC	NC	NC	CLOCK	A_CAD30 //A_D9	A_CCLKRUN //A_WP (IOIS16)	A_CSTSCHG //A_BVD1 (STSCHG/RI)	A_CVS1 //A_VS1	A_CAD24 //A_A2	VCCA	A_CAD20 //A_A6	NC	NC	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Figure 2–1. PCI1515 GHK/ZHK-Package Terminal Diagram

Table 2–1 lists the terminal assignments arranged in terminal-number order, with corresponding signal names for both CardBus and 16-bit PC Cards for the PCI1515 GHK package. Table 2–2 and Table 2–3 list the terminal assignments arranged in alphanumerical order by signal name, with corresponding terminal numbers for the GHK package; Table 2–2 is for CardBus signal names and Table 2–3 is for 16-bit PC Card signal names.

Terminal E5 on the GHK package is an identification ball used for device orientation.

Table 2-1. Signal Names by GHK Terminal Number

TERMINAL		L NAME	TERMINAL	SIGNAL NAME			
NUMBER	CardBus PC Card	16-Bit PC Card	NUMBER	CardBus PC Card	16-Bit PC Card		
A02	NC	NC	C03	NC	NC		
A03	NC	NC	C04	NC	NC		
A04	NC	NC	C05	NC	NC		
A05	NC	NC	C06	NC	NC		
A06	NC	NC	C07	NC	NC		
A07	NC	NC	C08	NC	NC		
A08	NC	NC	C09	LATCH	LATCH		
A09	CLOCK	CLOCK	C10	A_CAD31	A_D10		
A10	A_CAD30	A_D9	C11	A_CAD27	A_D0		
A11	A_CCLKRUN	A_WP(IOIS16)	C12	A_CSERR	A_WAIT		
A12	A_CSTSCHG	A_BVD1(STSCHG/RI)	C13	A_CAD25	A_A1		
A13	A_CVS1	A_VS1	C14	A_CREQ	A_INPACK		
A14	A_CAD24	A_A2	C15	A_CRST	A_RESET		
A15	VCCA	VCCA	C16	NC	NC		
A16	A_CAD20	A_A6	C17	NC	NC		
A17	NC	NC	C18	NC	NC		
A18	NC	NC	C19	NC	NC		
B01	NC	NC	D01	NC	NC		
B02	NC	NC	D02	NC	NC		
B03	NC	NC	D03	NC	NC		
B04	NC	NC	D17	NC	NC		
B05	NC	NC	D18	NC	NC		
B06	NC	NC	D19	A_CAD19	A_A25		
B07	NC	NC	E01	NC	NC		
B08	NC	NC	E02	NC	NC		
B09	DATA	DATA	E03	NC	NC		
B10	A_RSVD	A_D2	E06	NC	NC		
B11	A_CCD2	A_CD2	E07	NC	NC		
B12	A_CAUDIO	A_BVD2(SPKR)	E08	NC	NC		
B13	A_CAD26	A_A0	E09	NC	NC		
B14	A_CAD23	A_A3	E10	A_USB_EN	A_USB_EN		
B15	A_CAD22	A_A4	E11	A_CAD28	A_D8		
B16	A_CVS2	A_VS2	E12	A_CINT	A_READY(IREQ)		
B17	NC	NC	E13	A_CC/BE3	A_REG		
B18	NC	NC	E14	A_CAD21	A_A5		
B19	NC	NC	E17	A_CAD18	A_A7		
C01	NC	NC	E18	A_CC/BE2	A_A12		
C02	NC	NC	E19	A_CFRAME	A_A23		

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

TERMINAL NUMBER	SIGNAL	NAME	TERMINAL	SIGNAL NAME			
	CardBus PC Card	16-Bit PC Card	NUMBER	CardBus PC Card	16-Bit PC Card		
F01	RSVD	RSVD	J18	A_CAD13	A_IORD		
F02	NC	NC	J19	VCCA	VCCA		
F03	NC	NC	K01	VR_PORT	VR_PORT		
F05	NC	NC	K02	VR_EN	VR_EN		
F06	VCC	V _{CC}	K03	PRST	PRST		
F07	GND	GND	K05	GRST	GRST		
F08	NC	NC	K06	GND	GND		
F09	VCC	VCC	K14	GND	GND		
F10	GND	GND	K15	A_CAD12	A_A11		
F11	A_CAD29	A_D1	K17	A_CAD11	A_OE		
F12	VCC	VCC	K18	A_CAD10	A_CE2		
F13	GND	GND	K19	VR_PORT	VR_PORT		
F14	Vcc	Vcc	L01	PCLK	PCLK		
F15	A_CAD17	A_A24	L02	GNT	GNT		
F17	A_CIRDY	A_A15	L03	REQ	REQ		
F18	A_CCLK	A_A16	L05	RI_OUT/PME	RI_OUT/PME		
F19	A_CDEVSEL	A_A21	L06	Vcc	Vcc		
G01	MFUNC0	MFUNC0	L14	VCC	Vcc		
G02	SCL	SCL	L15	A_CAD9	A_A10		
G03	SDA	SDA	L17	A_CC/BE0	A_CE1		
G05	NC	NC	L18	A_CAD8	A_D15		
G06	RSVD	RSVD	L19	A_CAD7	A_D7		
G14	GND	GND	M01	AD31	AD31		
G15	A_CTRDY	A_A22	M02	AD30	AD30		
G17	A_CGNT	A_WE	M03	AD29	AD29		
G18	A_CSTOP	A_A20	M05	AD27	AD27		
G19	A_CPERR	 A_A14	M06	AD28	AD28		
H01	MFUNC3	MFUNC3	M14	GND	GND		
H02	MFUNC2	MFUNC2	M15	A_CAD3	A_D5		
H03	SPKROUT	SPKROUT	M17	A_CAD6	 A_D13		
H05	MFUNC1	MFUNC1	M18	A_CAD5	A_D6		
H06	GND	GND	M19	A RSVD	A D14		
H14	A_CPAR	A_A13	N01	AD26	AD26		
H15	A_CBLOCK	A_A19	N02	AD25	AD25		
H17	A_RSVD	A_A18	N03	AD24	AD24		
H18	A_CC/BE1	A_A8	N05	IDSEL	IDSEL		
H19	A_CAD16	A_A17	N06	GND	GND		
J01	MFUNC4	MFUNC4	N14	NC	NC		
J02	MFUNC5	MFUNC5	N15	A_CCD1	A_CD1		
J03	MFUNC6	MFUNC6	N17	A_CAD2	A_D11		
J05	SUSPEND	SUSPEND	N18	A_CAD1	A_D4		
J06			N19	A_CAD1 A_CAD4	A_D4 A_D12		
J14	Vcc	VCC	P01				
J14 J15	VCC A_CAD14	VCC A_A9	P01	VCCP C/BE3	V _{CCP} C/BE3		
J15 J17	A_CAD14 A_CAD15	A_IOWR	P02 P03	AD23	AD23		

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

TERMINAL NUMBER	SIGNAL	NAME	TERMINAL	SIGNAL NAME			
	CardBus PC Card	16-Bit PC Card	NUMBER	CardBus PC Card	16-Bit PC Card		
P05	AD20	AD20	U11	AD2	AD2		
P06	VCC	VCC	U12	TEST1	TEST1		
P07	GND	GND	U13	GND	GND		
P08	VCC	VCC	U14	GND	GND		
P09	GND	GND	U15	VCC	V _{CC}		
P10	VCC	VCC	U16	NC	NC		
P11	AD1	AD1	U17	NC	NC		
P12	TEST0	TEST0	U18	GND	GND		
P13	VCC	Vcc	U19	VCC	Vcc		
P14	Vcc	Vcc	V01	NC	NC		
P15	VR_PORT	VR_PORT	V02	NC	NC		
P17	TEST4	TEST4	V03	NC	NC		
P18	NC	NC	V04	NC	NC		
P19	A_CAD0	A_D3	V05	IRDY	ĪRDY		
R01	AD22	AD22	V06	STOP	STOP		
R02	AD21	AD21	V07	C/BE1	C/BE1		
R03	AD19	AD19	V08	AD12	AD12		
R06	FRAME	FRAME	V09	AD10	AD10		
R07	PERR	PERR	V10	AD7	AD7		
R08	AD14	AD14	V11	AD3	AD3		
R09	AD8	AD8	V12	TEST2	TEST2		
R10	AD5	AD5	V13	NC	NC		
R11	AD0	AD0	V14	NC	NC		
R12	RSVD	RSVD	V15	NC	NC		
R13	NC	NC	V16	NC	NC		
R14	GND	GND	V17	NC	NC		
R17	GND	GND	V18	NC	NC		
R18	NC	NC	V19	NC	NC		
R19	NC	NC	W02	NC	NC		
T01	AD18	AD18	W03	NC	NC		
T02	AD17	AD17	W04	AD16	AD16		
T03	NC	NC	W05	TRDY	TRDY		
T17	NC	NC	W06	SERR	SERR		
T18	NC	NC	W07	AD15	AD15		
T19	NC	NC	W08	VCCP	VCCP		
U01	NC	NC	W09	AD11	AD11		
U02	NC	NC	W10	C/BE0	C/BE0		
U03	NC	NC	W11	AD4	AD4		
U04	NC	NC	W12	TEST3	TEST3		
U05	C/BE2	C/BE2	W13	NC	NC		
U06	DEVSEL	DEVSEL	W14	NC	NC		
U07	PAR	PAR	W15	NC	NC		
U08	AD13	AD13	W16	NC	NC		
U09	AD9	AD9	W17	NC	NC		
U10	AD6	AD6	W18	NC	NC		

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

SIGNAL	TERMINAL		TERMINAL	SIGNAL	TERMINAL	SIGNAL	TERMINAL
NAME	NUMBER	SIGNAL NAME	NUMBER	NAME	NUMBER	NAME	NUMBER
AD0	R11	A_CAD11	K17	A_CTRDY	G15	NC	A02
AD1	P11	A_CAD12	K15	A_CVS1	A13	NC	A03
AD2	U11	A_CAD13	J18	A_CVS2	B16	NC	A04
AD3	V11	A_CAD14	J15	A_RSVD	B10	NC	A05
AD4	W11	A_CAD15	J17	A_RSVD	H17	NC	A06
AD5	R10	A_CAD16	H19	A_RSVD	M19	NC	A07
AD6	U10	A_CAD17	F15	A_USB_EN	E10	NC	A08
AD7	V10	A_CAD18	E17	C/BE0	W10	NC	A17
AD8	R09	A_CAD19	D19	C/BE1	V07	NC	A18
AD9	U09	A_CAD20	A16	C/BE2	U05	NC	B01
AD10	V09	A_CAD21	E14	C/BE3	P02	NC	B02
AD11	W09	A_CAD22	B15	CLOCK	A09	NC	B03
AD12	V08	A_CAD23	B14	DATA	B09	NC	B04
AD13	U08	A_CAD24	A14	DEVSEL	U06	NC	B05
AD14	R08	A_CAD25	C13	FRAME	R06	NC	B06
AD15	W07	A_CAD26	B13	GND	F07	NC	B07
AD16	W04	A_CAD27	C11	GND	F10	NC	B08
AD17	T02	A_CAD28	E11	GND	F13	NC	B17
AD18	T01	A_CAD29	F11	GND	G14	NC	B18
AD19	R03	A_CAD30	A10	GND	H06	NC	B19
AD20	P05	A_CAD31	C10	GND	K06	NC	C01
AD21	R02	A_CAUDIO	B12	GND	K14	NC	C02
AD22	R01	A_CBLOCK	H15	GND	M14	NC	C03
AD23	P03	A_CC/BE0	L17	GND	N06	NC	C04
AD24	N03	A_CC/BE1	H18	GND	P07	NC	C05
AD25	N02	A_CC/BE2	E18	GND	P09	NC	C06
AD26	N01	A_CC/BE3	E13	GND	R14	NC	C07
AD27	M05	A_CCD1	N15	GND	R17	NC	C08
AD28	M06	A_CCD2	B11	GND	U13	NC	C16
AD29	M03	A_CCLK	F18	GND	U14	NC	C17
AD30	M02	A_CCLKRUN	A11	GND	U18	NC	C18
AD31	M01	A_CDEVSEL	F19	GNT	L02	NC	C19
A_CAD0	P19	A CFRAME	E19	GRST	K05	NC	D01
A CAD1	N18	A_CGNT	G17	IDSEL	N05	NC	D02
A_CAD2	N17	A_CINT	E12	IRDY	V05	NC	D03
A_CAD3	M15	A_CIRDY	F17	LATCH	C09	NC	D17
A CAD4	N19	A_CPAR	H14	MFUNC0	G01	NC	D18
A_CAD5	M18	A_CPERR	G19	MFUNC1	H05	NC	E01
A CAD6	M17	A_CREQ	C14	MFUNC2	H02	NC NC	E02
A_CAD7	L19	A_CRST	C14	MFUNC3	H01	NC NC	E03
A_CAD8	L18	A_CSERR	C13	MFUNC4	J01	NC NC	E06
A_CAD9	L15	A_CSTOP	G18	MFUNC5	J02	NC NC	E07
A_CAD9 A_CAD10	K18	A_CSTSCHG	A12	MFUNC6	J03	NC NC	E08

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

SIGNAL NAME	TERMINAL NUMBER	SIGNAL NAME	TERMINAL NUMBER	SIGNAL NAME	TERMINAL NUMBER	SIGNAL NAME	TERMINAL NUMBER
NC	E09	NC	V01	PERR	R07	Vcc	F12
NC	F02	NC	V02	PRST	K03	Vcc	F14
NC	F03	NC	V03	REQ	L03	Vcc	J06
NC	F05	NC	V04	RI_OUT/PME	L05	Vcc	J14
NC	F08	NC	V13	RSVD	F01	Vcc	L06
NC	G05	NC	V14	RSVD	G06	Vcc	L14
NC	N14	NC	V15	RSVD	R12	Vcc	P06
NC	P18	NC	V16	SCL	G02	Vcc	P08
NC	R13	NC	V17	SDA	G03	Vcc	P10
NC	R18	NC	V18	SERR	W06	Vcc	P13
NC	R19	NC	V19	SPKROUT	H03	Vcc	P14
NC	T03	NC	W02	STOP	V06	Vcc	U15
NC	T17	NC	W03	SUSPEND	J05	V _{CC}	U19
NC	T18	NC	W13	TEST0	P12	VCCA	A15
NC	T19	NC	W14	TEST1	U12	VCCA	J19
NC	U01	NC	W15	TEST2	V12	VCCP	P01
NC	U02	NC	W16	TEST3	W12	VCCP	W08
NC	U03	NC	W17	TEST4	P17	VR_EN	K02
NC	U04	NC	W18	TRDY	W05	VR_PORT	K01
NC	U16	PAR	U07	Vcc	F06	VR_PORT	K19
NC	U17	PCLK	L01	VCC	F09	VR_PORT	P15

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

SIGNAL NAME	TERMINAL NUMBER	SIGNAL NAME	TERMINAL NUMBER	SIGNAL NAME	TERMINAL NUMBER	SIGNAL NAME	TERMINAL NUMBER
AD0	R11	A_A11	K15	A_RESET	C15	NC	A02
AD1	P11	A_A12	E18	A_USB_EN	E10	NC	A03
AD2	U11	A_A13	H14	A_VS1	A13	NC	A04
AD3	V11	A_A14	G19	A_VS2	B16	NC	A05
AD4	W11	A_A15	F17	A_WAIT	C12	NC	A06
AD5	R10	A_A16	F18	A_WE	G17	NC	A07
AD6	U10	A_A17	H19	A_WP(IOIS16)	A11	NC	A08
AD7	V10	A_A18	H17	C/BE0	W10	NC	A17
AD8	R09	A_A19	H15	C/BE1	V07	NC	A18
AD9	U09	A_A20	G18	C/BE2	U05	NC	B01
AD10	V09	A_A21	F19	C/BE3	P02	NC	B02
AD11	W09	A_A22	G15	CLOCK	A09	NC	B03
AD12	V08	A_A23	E19	DATA	B09	NC	B04
AD13	U08	A_A24	F15	DEVSEL	U06	NC	B05
AD14	R08	A_A25	D19	FRAME	R06	NC	B06
AD15	W07	A_BVD1(STSCHG/RI)	A12	GND	F07	NC	B07
AD16	W04	A_BVD2(SPKR)	B12	GND	F10	NC	B08
AD17	T02	A_CD1	N15	GND	F13	NC	B17
AD18	T01	A_CD2	B11	GND	G14	NC	B18
AD19	R03	A_CE1	L17	GND	H06	NC	B19
AD20	P05	A_CE2	K18	GND	K06	NC	C01
AD21	R02	A_D0	C11	GND	K14	NC	C02
AD22	R01	A_D1	F11	GND	M14	NC	C03
AD23	P03	A_D2	B10	GND	N06	NC	C04
AD24	N03	A_D3	P19	GND	P07	NC	C05
AD25	N02	A_D4	N18	GND	P09	NC	C06
AD26	N01	A_D5	M15	GND	R14	NC	C07
AD27	M05	A_D6	M18	GND	R17	NC	C08
AD28	M06	A_D7	L19	GND	U13	NC	C16
AD29	M03	A_D8	E11	GND	U14	NC	C17
AD30	M02	A_D9	A10	GND	U18	NC	C18
AD31	M01	A_D10	C10	GNT	L02	NC	C19
A_A0	B13	A_D11	N17	GRST	K05	NC	D01
A_A1	C13	A_D12	N19	IDSEL	N05	NC	D02
A_A2	A14	A_D13	M17	IRDY	V05	NC	D03
A_A3	B14	A_D14	M19	LATCH	C09	NC	D17
A_A4	B15	A_D15	L18	MFUNC0	G01	NC	D18
A_A5	E14	A_INPACK	C14	MFUNC1	H05	NC	E01
A_A6	A16	A_IORD	J18	MFUNC2	H02	NC	E02
A_A7	E17	A_IOWR	J17	MFUNC3	H01	NC	E03
A_A8	H18	A_OE	K17	MFUNC4	J01	NC	E06
A_A9	J15	A_READY(IREQ)	E12	MFUNC5	J02	NC	E07
A_A10	L15	A_REG	E13	MFUNC6	J03	NC	E08

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

SIGNAL NAME	TERMINAL NUMBER	SIGNAL NAME	TERMINAL NUMBER	SIGNAL NAME	TERMINAL NUMBER	SIGNAL NAME	TERMINAL NUMBER
NC	E09	NC	V01	PERR	R07	Vcc	F12
NC	F02	NC	V02	PRST	K03	Vcc	F14
NC	F03	NC	V03	REQ	L03	Vcc	J06
NC	F05	NC	V04	RI_OUT/PME	L05	Vcc	J14
NC	F08	NC	V13	RSVD	F01	Vcc	L06
NC	G05	NC	V14	RSVD	G06	Vcc	L14
NC	N14	NC	V15	RSVD	R12	Vcc	P06
NC	P18	NC	V16	SCL	G02	Vcc	P08
NC	R13	NC	V17	SDA	G03	Vcc	P10
NC	R18	NC	V18	SERR	W06	Vcc	P13
NC	R19	NC	V19	SPKROUT	H03	Vcc	P14
NC	T03	NC	W02	STOP	V06	V _{CC}	U15
NC	T17	NC	W03	SUSPEND	J05	Vcc	U19
NC	T18	NC	W13	TEST0	P12	VCCA	A15
NC	T19	NC	W14	TEST1	U12	VCCA	J19
NC	U01	NC	W15	TEST2	V12	VCCP	P01
NC	U02	NC	W16	TEST3	W12	VCCP	W08
NC	U03	NC	W17	TEST4	P17	VR_EN	K02
NC	U04	NC	W18	TRDY	W05	VR_PORT	K01
NC	U16	PAR	U07	V _{CC}	F06	VR_PORT	K19
NC	U17	PCLK	L01	Vcc	F09	VR_PORT	P15

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

Table 2-4. Power Supply Terminals

TE	TERMINAL		DECORPTION		
NAME	NUMBER	I/O	DESCRIPTION		
GND	F07, F10, F13, G14, H06, K06, K14, M14, N06, P07, P09, R14, R17, U13, U14, U18	_	Digital ground terminal		
Vcc	F06, F09, F12, F14, J06, J14, L06, L14, P06, P08, P10, P13, P14, U15, U19	_	3.3-V power supply terminal for I/O and internal voltage regulator		
VCCA	A15, J19	_	Clamp voltage for PC Card A interface. Matches card A signaling environment, 5 V or 3.3 V		
VCCP	P01, W08	_	Clamp voltage for PCI and miscellaneous I/O, 5 V or 3.3 V		
VR_EN	K02	I	Internal voltage regulator enable. Active low		
VR_PORT	K01, K19, P15	I/O	1.5-V output from the internal voltage regulator		

Table 2-5. PC Card Power Switch Terminals

TERM	IINAL	1/0	DESCRIPTION			
NAME	NUMBER	1/0	DESCRIPTION			
CLOCK	A09	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 an output by using bit 27 (P2CCLK) in the system control register (offset 80h, see Section 4.29).			
DATA	B09	0	Power switch data. DATA is used to communicate socket power control information serially to the power switch.			
LATCH	C09	0	Power switch latch. LATCH is asserted by the controller to indicate to the power switch that the data on the DATA line is valid.			

Table 2-6. PCI System Terminals

TERM	TERMINAL		DESCRIPTION
NAME	NUMBER	1/0	DESCRIPTION
GRST	K05	I	Global reset. When the global reset is asserted, the GRST signal causes the controller to place all output buffers in a high-impedance state and reset all internal registers. When GRST is asserted, the controller is completely in its default state. For systems that require wake-up from D3, GRST is normally asserted only during initial boot. PRST must 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, GRST must be tied to PRST. When the SUSPEND mode is enabled, the controller is protected from the 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	L01	-	PCI bus clock. PCLK provides timing for all transactions on the PCI bus. All PCI signals are sampled at the rising edge of PCLK.
PRST	K03	ı	PCI bus reset. When the PCI bus reset is asserted, PRST causes the controller to place all output buffers in a high-impedance state and reset some internal registers. When PRST is asserted, the controller is completely nonfunctional. After PRST is deasserted, the controller is in a default state. When SUSPEND and PRST are asserted, the controller is protected from PRST clearing the internal registers. All outputs are placed in a high-impedance state, but the contents of the registers are preserved.

Table 2-7. PCI Address and Data Terminals

TER	MINAL		
NAME	NUMBER	1/0	DESCRIPTION
AD31	M01		
AD30	M02		
AD29	M03		
AD28	M06		
AD27	M05		
AD26	N01		
AD25	N02		
AD24	N03		
AD23	P03		
AD22	R01		
AD21	R02		
AD20	P05		
AD19	R03		
AD18	T01		
AD17	T02		
AD16	W04	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
AD15	W07	1/0	destination information. During the data phase, AD31–AD0 contain data.
AD14	R08		german and a germa
AD13	U08		
AD12	V08		
AD11	W09		
AD10	V09		
AD9	U09		
AD8	R09		
AD7	V10		
AD6	U10		
AD5	R10		
AD4	W11		
AD3	V11		
AD2	U11		
AD1	P11		
AD0	R11		
C/BE3	P02		PCI-bus commands and byte enables. These signals are multiplexed on the same PCI terminals. During the
C/BE2	U05	I/O	address phase of a primary-bus PCI cycle, C/BE3–C/BE0 define the bus command. During the data phase, this 4-bit bus is used as byte enables. The byte enables determine which byte paths of the full 32-bit data bus carry
C/BE1	V07	,,,	meaningful data. C/BE0 applies to byte 0 (AD7–AD0), C/BE1 applies to byte 1 (AD15–AD8), C/BE2 applies to
C/BE0	W10		byte 2 (AD23–AD16), and C/BE3 applies to byte 3 (AD31–AD24).
PAR	U07	I/O	PCI-bus parity. In all PCI-bus read and write cycles, the controller calculates even parity across the AD31–AD0 and C/BE3–C/BE0 buses. As an initiator during PCI cycles, the controller outputs this parity indicator with a one-PCLK delay. As a target during PCI cycles, the controller compares its calculated parity to the parity indicator of the initiator. A compare error results in the assertion of a parity error (PERR).

Table 2–8. PCI Interface Control Terminals

TERM	/IINAL		
NAME	NUMBER	1/0	DESCRIPTION
DEVSEL	U06	I/O	PCI device select. The controller asserts DEVSEL to claim a PCI cycle as the target device. As a PCI initiator on the bus, the controller monitors DEVSEL until a target responds. If no target responds before timeout occurs, then the controller terminates the cycle with an initiator abort.
FRAME	R06	I/O	PCI cycle frame. FRAME is driven by the initiator of a bus cycle. FRAME is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When FRAME is deasserted, the PCI bus transaction is in the final data phase.
GNT	L02	I	PCI bus grant. GNT is driven by the PCI bus arbiter to grant the controller access to the PCI bus after the current data transaction has completed. GNT may or may not follow a PCI bus request, depending on the PCI bus parking algorithm.
IDSEL	N05	I	Initialization device select. IDSEL selects the controller during configuration space accesses. IDSEL can be connected to one of the upper 24 PCI address lines on the PCI bus.
ĪRDY	V05	I/O	PCI initiator ready. IRDY indicates the ability of the PCI bus initiator to complete the current data phase of the transaction. A data phase is completed on a rising edge of PCLK where both IRDY and TRDY are asserted. Until IRDY and TRDY are both sampled asserted, wait states are inserted.
PERR	R07	I/O	PCI parity error indicator. PERR is driven by a PCI controller to indicate that calculated parity does not match PAR when PERR is enabled through bit 6 of the command register (PCI offset 04h, see Section 4.4).
REQ	L03	0	PCI bus request. REQ is asserted by the controller to request access to the PCI bus as an initiator.
SERR	W06	0	PCI system error. SERR is an output that is pulsed from the controller when enabled through bit 8 of the command register (PCI offset 04h, see Section 4.4) indicating a system error has occurred. The controller need not be the target of the PCI cycle to assert this signal. When SERR is enabled in the command register, this signal also pulses, indicating that an address parity error has occurred on a CardBus interface.
STOP	V06	I/O	PCI cycle stop signal. STOP is driven by a PCI target to request the initiator to stop the current PCI bus transaction. STOP is used for target disconnects and is commonly asserted by target devices that do not support burst data transfers.
TRDY	W05	I/O	PCI target ready. TRDY indicates the ability of the primary bus target to complete the <u>current</u> data phase of the transaction. A data phase is completed on a rising edge of PCLK when both IRDY and TRDY are asserted. Until both IRDY and TRDY are asserted, wait states are inserted.

Table 2-9. Multifunction and Miscellaneous Terminals

TERMINAL					
NAME	NUMBER	1/0	DESCRIPTION		
A_USB_EN	E10	0	USB enable. This output terminal controls an external CBT switch when an USB card is inserted into the socket.		
MFUNC0	G01				
MFUNC1	H05				
MFUNC2	H02				
MFUNC3	H01	I/O	Multifunction terminals 0–6. See Section 4.35, <i>Multifunction Routing Status Register</i> , for configuration details.		
MFUNC4	J01		Trogrator, for corningulation dotails.		
MFUNC5	J02				
MFUNC6	J03	1			
NC	A02, A03, A04, A05, A06, A07, A08, A17, A18, B01, B02, B03, B04, B05, B06, B07, B08, B17, B18, B19, C01, C02, C03, C04, C05, C06, C07, C08, C16, C17, C18, C19, D01, D02, D03, D17, D18, E01, E02, E03, E06, E07, E08, E09, F02, F03, F05, F08, G05, N14, P18, R13, R18, R19, T03, T17, T18, T19, U01, U02, U03, U04, U16, U17, V01, V02, V03, V04, V13, V14, V15, V16, V17, V18, V19, W02, W03, W13, W14, W15, W16, W17, W18	_	Reserved. This terminal has no connection anywhere within the package.		
RSVD	F01, R12	l —	Reserved. This terminal must be tied to ground.		
RSVD	G06	l —	Reserved. This terminal must be connected to V _{CC} or GND.		
RI_OUT/ PME	L05	0	Ring indicate out and power management event output. This terminal provides an output for ring-indicate or PME signals.		
SCL	G02	I/O	Serial clock. At $\overline{\text{PRST}}$, the SCL signal is sampled to determine if a two-wire serial ROM is present. If the serial ROM is detected, then this terminal provides the serial clock signaling and is implemented as open-drain. For normal operation (a ROM is implemented in the design), this terminal must be pulled high to the ROM VDD with a 2.7-k Ω resistor. Otherwise, it must be pulled low to ground with a 220- Ω resistor.		
SDA	G03	I/O	Serial data. If the serial ROM is detected, then this terminal provides the serial data signaling and is implemented as open-drain. For normal operation (a ROM is implemented in the design), this terminal must be pulled high to the ROM $V_{\mbox{\scriptsize DD}}$ with a 2.7-k Ω resistor. Otherwise, it must be pulled low to ground with a 220- Ω resistor.		
SPKROUT	H03	0	Speaker output. SPKROUT is the output to the host system that can carry SPKR or CAUDIO through the controller from the PC Card interface. SPKROUT is driven as the exclusive-OR combination of card SPKR//CAUDIO inputs.		
SUSPEND	J05	I	Suspend. SUSPEND protects the internal registers from clearing when the GRST or PRST signal is asserted. See Section 3.8.5, Suspend Mode, for details.		
TEST0 TEST1 TEST2 TEST3	P12 U12 V12 W12	I/O	Terminals TEST0-TEST3 are used for factory test of the controller and must be connected to ground for normal operation.		
TEST4	P17	I/O	Terminal TEST4 is not for customer use. It must be pulled high with a 4.7-k Ω resistor.		

Table 2–10. 16-Bit PC Card Address and Data Terminals

TERM	IINAL	1/0	
NAME	NUMBER		DESCRIPTION
A_A25	D19		PC Card address. 16-bit PC Card address lines. A25 is the most significant bit.
A_A24	F15		
A_A23	E19		
A_A22	G15		
A_A21	F19		
A_A20	G18		
A_A19	H15		
A_A18	H17		
A_A17	H19		
A_A16	F18		
A_A15	F17		
A_A14	G19		
A_A13	H14	0	
A_A12	E18		
A_A11	K15		
A_A10	L15		
A_A9	J15		
A_A8	H18		
A_A7	E17		
A_A6	A16		
A_A5	E14		
A_A4	B15		
A_A3	B14		
A_A2	A14		
A_A1	C13		
A_A0	B13		
A_D15	L18		
A_D14	M19		PC Card data. 16-bit PC Card data lines. D15 is the most significant bit.
A_D13	M17		
A_D12	N19		
A_D11	N17		
A_D10	C10		
A_D9	A10		
A_D8	E11	I/O	
A_D7	L19 M18		
A_D6 A_D5	M15		
A_D5 A_D4	N18		
A_D4 A_D3	P19		
A_D3 A_D2	B10		
A_D2 A_D1	F11		
A_D1 A_D0	C11		
A_DU	CIT		

Table 2–11. 16-Bit PC Card Interface Control Terminals

TERMINAL			
NAME	NUMBER	1/0	DESCRIPTION
A_BVD1 (STSCHG/RI)	A12	ı	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 must 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.
A_BVD2 (SPKR)	B12	ı	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 must 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 controller 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.
A_CD1 A_CD2	N15 B11	I	Card detect 1 and card detect 2. $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ are internally connected to ground on the PC Card. When a PC Card is inserted into a socket, $\overline{\text{CD1}}$ and $\overline{\text{CD2}}$ are pulled low. For signal status, see Section 5.2, $ExCA$ Interface Status Register.
A_CE1 A_CE2	L17 K18	0	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.
A_INPACK	C14	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.
A_IORD	J18	0	I/O read. IORD is asserted by the controller 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 controller asserts IORD during DMA transfers from the PC Card to host memory.
A_IOWR	J17	0	I/O write. IOWR is driven low by the controller 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 controller asserts IOWR during transfers from host memory to the PC Card.

Table 2-11. 16-Bit PC Card Interface Control Terminals (Continued)

TERMINAL			
NAME	NUMBER	1/0	DESCRIPTION
Ā_ŌĒ	K17	0	Output enable. OE is driven low by the controller to enable 16-bit memory PC Card data output during host memory read cycles. DMA terminal count. OE is used as terminal count (TC) during DMA operations to a 16-bit PC Card that supports DMA. The controller asserts OE to indicate TC for a DMA write operation.
A_READY (IREQ)	E12	ı	Ready. The ready function is provided when the 16-bit PC Card and the host socket are configured for the memory-only interface. READY is driven low by 16-bit memory PC Cards to indicate that the memory card circuits are busy processing a previous write command. READY is driven high when the 16-bit memory PC Card is ready to accept a new data transfer command. Interrupt request. IREQ is asserted by a 16-bit I/O PC Card to indicate to the host that a controller on the 16-bit I/O PC Card requires service by the host software. IREQ is high (deasserted) when no interrupt is requested.
Ā_REG	E13	0	Attribute memory select. REG remains high for all common memory accesses. When REG is asserted, access is limited to attribute memory (OE or WE active) and to the I/O space (IORD or IOWR active). Attribute memory is a separately accessed section of card memory and is generally used to record card capacity and other configuration and attribute information. DMA acknowledge. REG is used as a DMA acknowledge (DACK) during DMA operations to a 16-bit PC Card that supports DMA. The controller asserts REG to indicate a DMA operation. REG is used in conjunction with the DMA read (IOWR) or DMA write (IORD) strobes to transfer data.
A_RESET	C15	0	PC Card reset. RESET forces a hard reset to a 16-bit PC Card.
A_VS1 A_VS2	A13 B16	I/O	Voltage sense 1 and voltage sense 2. $\overline{\text{VS1}}$ and $\overline{\text{VS2}}$, when used in conjunction with each other, determine the operating voltage of the PC Card.
A_WAIT	C12	I	Bus cycle wait. WAIT is driven by a 16-bit PC Card to extend the completion of the memory or I/O cycle in progress.
A_WE	G17	0	Write enable. WE is used to strobe memory write data into 16-bit memory PC Cards. WE is also used for memory PC Cards that employ programmable memory technologies. DMA terminal count. WE is used as a TC during DMA operations to a 16-bit PC Card that supports DMA. The controller asserts WE to indicate the TC for a DMA read operation.
A_WP (IOIS16)	A11	ı	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 (IOIS16) function. I/O is 16 bits. IOIS16 applies to 16-bit I/O PC Cards. 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.

Table 2–12. CardBus PC Card Interface System Terminals

SOCKET A TERMINAL			PEODERICAL
NAME	NUMBER	1/0	DESCRIPTION
A_CCLK	F18	0	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.
A_CCLKRUN	A11	I/O	CardBus clock run. CCLKRUN is used by a CardBus PC Card to request an increase in the CCLK frequency, and by the controller to indicate that the CCLK frequency is going to be decreased.
A_CRST	C15	0	CardBus reset. CRST brings CardBus PC Card-specific registers, sequencers, and signals to a known state. When CRST is asserted, all CardBus PC Card signals are placed in a high-impedance state, and the controller drives these signals to a valid logic level. Assertion can be asynchronous to CCLK, but deassertion must be synchronous to CCLK.

Table 2–13. CardBus PC Card Address and Data Terminals

TERMINAL				
NAME	NUMBER	I/O	DESCRIPTION	
A_CAD31	C10		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.	
A_CAD30	A10			
A_CAD29	F11			
A_CAD28	E11			
A_CAD27	C11			
A_CAD26	B13			
A_CAD25	C13			
A_CAD24	A14			
A_CAD23	B14			
A_CAD22	B15			
A_CAD21	E14			
A_CAD20	A16			
A_CAD19	D19			
A_CAD18	E17			
A_CAD17	F15			
A_CAD16	H19	I/O		
A_CAD15	J17	1/0		
A_CAD14	J15			
A_CAD13	J18			
A_CAD12	K15			
A_CAD11	K17			
A_CAD10	K18			
A_CAD9	L15			
A_CAD8	L18			
A_CAD7	L19			
A_CAD6	M17			
A_CAD5	M18			
A_CAD4	N19			
A_CAD3	M15			
A_CAD2	N17			
A_CAD1	N18			
A_CAD0	P19			
A_CC/BE3	E13		CardBus bus commands and byte enables. CC/BE3-CC/BE0 are multiplexed on the same CardBus	
A_CC/BE2	E18		terminals. During the address phase of a CardBus cycle, CC/BE3–CC/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	
A_CC/BE1	H18	I/O	full 32-bit data bus carry meaningful data. CC/BE0 applies to byte 0 (CAD7–CAD0), CC/BE1 applies to	
A_CC/BE0	L17		byte 1 (CAD15-CAD8), CC/BE2 applies to byte 2 (CAD23-CAD16), and CC/BE3 applies to byte 3 (CAD31-CAD24).	
A_CPAR	H14	I/O	CardBus parity. In all CardBus read and write cycles, the controller calculates even parity across the CAD and CC/BE buses. As an initiator during CardBus cycles, the controller outputs CPAR with a one-CCLK delay. As a target during CardBus cycles, the controller compares its calculated parity to the parity indicator of the initiator; a compare error results in a parity error assertion.	

Table 2-14. CardBus PC Card Interface Control Terminals

TERMINAL			
NAME	NUMBER	I/O	DESCRIPTION
A_CAUDIO	B12	I	CardBus audio. CAUDIO is a digital input signal from a PC Card to the system speaker. The controller supports the binary audio mode and outputs a binary signal from the card to SPKROUT.
A_CBLOCK	H15	I/O	CardBus lock. CBLOCK is used to gain exclusive access to a target.
A_CCD1 A_CCD2	N15 B11	1	CardBus detect 1 and CardBus detect 2. CCD1 and CCD2 are used in conjunction with CVS1 and CVS2 to identify card insertion and interrogate cards to determine the operating voltage and card type.
A_CDEVSEL	F19	I/O	CardBus device select. The controller asserts CDEVSEL to claim a CardBus cycle as the target device. As a CardBus initiator on the bus, the controller monitors CDEVSEL until a target responds. If no target responds before timeout occurs, then the controller terminates the cycle with an initiator abort.
A_CFRAME	E19	I/O	CardBus cycle frame. CFRAME is driven by the initiator of a CardBus bus cycle. CFRAME is asserted to indicate that a bus transaction is beginning, and data transfers continue while this signal is asserted. When CFRAME is deasserted, the CardBus bus transaction is in the final data phase.
A_CGNT	G17	0	CardBus bus grant. CGNT is driven by the controller to grant a CardBus PC Card access to the CardBus bus after the current data transaction has been completed.
A_CINT	E12	I	CardBus interrupt. CINT is asserted low by a CardBus PC Card to request interrupt servicing from the host.
A_CIRDY	F17	I/O	CardBus initiator ready. CIRDY indicates the ability of the CardBus initiator to complete the current data phase of the transaction. A data phase is completed on a rising edge of CCLK when both CIRDY and CTRDY are asserted. Until CIRDY and CTRDY are both sampled asserted, wait states are inserted.
A_CPERR	G19	I/O	CardBus parity error. CPERR reports parity errors during CardBus transactions, except during special cycles. It is driven low by a target two clocks following the data cycle during which a parity error is detected.
A_CREQ	C14	I	CardBus request. CREQ indicates to the arbiter that the CardBus PC Card desires use of the CardBus bus as an initiator.
A_CSERR	C12	I	CardBus system error. CSERR reports address parity errors and other system errors that could lead to catastrophic results. CSERR is driven by the card synchronous to CCLK, but deasserted by a weak pullup; deassertion may take several CCLK periods. The controller can report CSERR to the system by assertion of SERR on the PCI interface.
A_CSTOP	G18	I/O	CardBus stop. CSTOP is driven by a CardBus target to request the initiator to stop the current CardBus transaction. CSTOP is used for target disconnects, and is commonly asserted by target devices that do not support burst data transfers.
A_CSTSCHG	A12	I	CardBus status change. CSTSCHG alerts the system to a change in the card status, and is used as a wake-up mechanism.
A_CTRDY	G15	I/O	CardBus target ready. CTRDY indicates the ability of the CardBus target to complete the current data phase of the transaction. A data phase is completed on a rising edge of CCLK, when both CIRDY and CTRDY are asserted; until this time, wait states are inserted.
A_CVS1 A_CVS2	A13 B16	I/O	CardBus voltage sense 1 and CardBus voltage sense 2. CVS1 and CVS2 are used in conjunction with CCD1 and CCD2 to identify card insertion and interrogate cards to determine the operating voltage and card type.

3 Feature/Protocol Descriptions

The following sections give an overview of the PCI1515 controller. Figure 3–1 shows the connections to the PCI1515 controller. 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.

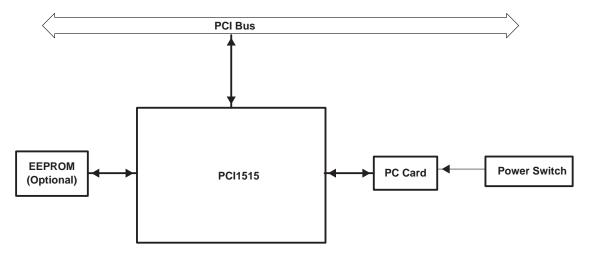


Figure 3-1. PCI1515 System Block Diagram

3.1 Power Supply Sequencing

The PCI1515 controller 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 1.5 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. Power core 1.5 V.
- 2. Apply the I/O voltage.
- 3. Apply the clamp voltage.

The power-down sequence is:

- 1. Remove the clamp voltage.
- 2. Remove the I/O voltage.
- 3. Remove power from the core.

NOTE: If the voltage regulator is enabled, then steps 2 and 3 of the power-up sequence and steps 1 and 2 of the power-down sequence all occur simultaneously.

3.2 I/O Characteristics

The PCI1515 controller meets the ac specifications of the *PC Card Standard* (release 8.1) and the *PCI Local Bus Specification*. Figure 3–2 shows a 3-state bidirectional buffer. Section 7.2, *Recommended Operating Conditions*, provides the electrical characteristics of the inputs and outputs.

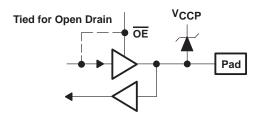


Figure 3-2. 3-State Bidirectional Buffer

3.3 Clamping Voltages

The clamping voltages are set to match whatever external environment the PCI1515 controller is interfaced with: 3.3 V or 5 V. The I/O sites can be pulled through a clamping diode to a voltage rail that protects the core from external signals. The core power supply is 1.5 V and is independent of the clamping voltages. For example, PCI signaling can be either 3.3 V or 5 V, and the PCI1515 controller 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.

3.4 Peripheral Component Interconnect (PCI) Interface

The PCI1515 controller is fully compliant with the *PCI Local Bus Specification*. The PCI1515 controller 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 PCI1515 controller provides the \overline{INTA} interrupt signal.

3.4.1 Device Resets

During the power-up sequence, \overline{GRST} and \overline{PRST} must be asserted. \overline{GRST} is deasserted a minimum of 2 ms after V_{CC} is stable. \overline{PRST} can be deasserted a minimum of 100 μ s after PCLK is stable or any time thereafter.

3.4.2 PCI Bus Lock (LOCK)

The bus-locking protocol defined in the *PCI Local Bus Specification* is not highly recommended, but is provided on the PCI1515 controller as an additional compatibility feature. The PCI LOCK signal can be routed to the MFUNC4 terminal by setting the appropriate values in bits 19–16 of the multifunction routing status register. See Section 4.35, *Multifunction Routing Status Register*, for details. Note that the use of LOCK is only supported by PCI-to-CardBus bridges in the downstream direction (away from the processor).

PCI LOCK indicates an atomic operation that may require multiple transactions to complete. When 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 assure control of LOCK; control of 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 LOCK. Note that the CardBus signal for this protocol is CBLOCK to avoid confusion with the bus clock.

An agent may need to do an exclusive operation because a critical access to memory might be broken into several transactions, but the master wants exclusive rights to a region of memory. The granularity of the lock is defined by PCI to be 16 bytes, aligned. The LOCK protocol defined by 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{LOCK} protocol. In this scenario, the arbiter does not grant the bus to any other agent (other than the \overline{LOCK} master) while \overline{LOCK} is asserted. A complete bus lock may have a significant impact on the performance of the video. The arbiter that supports complete bus \overline{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 PCI1515 controller supports all LOCK protocols 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{LOCK} .

3.4.3 Serial EEPROM I²C Bus

The PCI1515 controller offers many choices for modes of operation, and these choices are selected by programming several configuration registers. For system board applications, these registers are normally programmed through the BIOS routine. For add-in card and docking-station/port-replicator applications, the PCI1515 controller provides a two-wire inter-integrated circuit (IIC or I²C) serial bus for use with an external serial EEPROM.

The PCI1515 controller is always the bus master, and the EEPROM is always the slave. Either device can drive the bus low, but neither device drives the bus high. The high level is achieved through the use of pullup resistors on the SCL and SDA signal lines. The PCI1515 controller is always the source of the clock signal, SCL.

System designers who wish to load register values with a serial EEPROM must use pullup resistors on the SCL and SDA terminals. If the PCI1515 controller detects a logic-high level on the SCL terminal at the end of GRST, then it initiates incremental reads from the external EEPROM. Any size serial EEPROM up to the I²C limit of 16 Kbits can be used, but only the first 96 bytes (from offset 00h to offset 5Fh) are required to configure the PCI1515 controller. Figure 3–3 shows a serial EEPROM application.

In addition to loading configuration data from an EEPROM, the PCI1515 I²C bus can be used to read and write from other I²C serial devices. A system designer can control the I²C bus, using the PCI1515 controller as bus master, by reading and writing PCI configuration registers. Setting bit 3 (SBDETECT) in the serial bus control/status register (PCI offset B3h, see Section 4.49) causes the PCI1515 controller to route the SDA and SCL signals to the SDA and SCL terminals, respectively. The read/write data, slave address, and byte addresses are manipulated by accessing the serial bus data, serial bus index, and serial bus slave address registers (PCI offsets B0h, B1h, and B2h; see Sections 4.46, 4.47, and 4.48, respectively).

EEPROM interface status information is communicated through the serial bus control and status register (PCI offset B3h, see Section 4.49). Bit 3 (SBDETECT) in this register indicates whether or not the PCI1515 serial ROM circuitry detects the pullup resistor on SCL. Any undefined condition, such as a missing acknowledge, results in bit 0 (ROM_ERR) being set. Bit 4 (ROMBUSY) is set while the subsystem ID register is loading (serial ROM interface is busy).

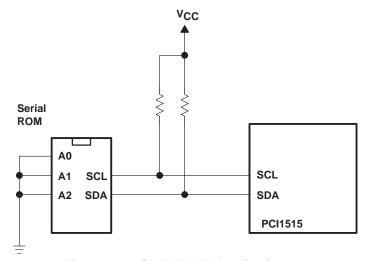


Figure 3-3. Serial ROM Application

3.4.4 Function 0 (CardBus) Subsystem Identification

The subsystem vendor ID register (PCI offset 40h, see Section 4.26) and subsystem ID register (PCI offset 42h, see Section 4.27) make up a doubleword of PCI configuration space for function 0. This doubleword register is used for

system and option card (mobile dock) identification purposes and is required by some operating systems. Implementation of this unique identifier register is a *PC 99/PC 2001* requirement.

The PCI1515 controller offers two mechanisms to load a read-only value into the subsystem registers. The first mechanism relies upon the system BIOS providing the subsystem ID value. The default access mode to the subsystem registers is read-only, but can be made read/write by clearing bit 5 (SUBSYSRW) in the system control register (PCI offset 80h, see Section 4.29). Once this bit is cleared, the BIOS can write a subsystem identification value into the registers at PCI offset 40h. The BIOS must set the SUBSYSRW bit such that the subsystem vendor ID register and subsystem ID register are 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 PCI1515 controller loads the data from the serial EEPROM after a reset of the primary bus. Note that the SUSPEND input gates the PCI reset from the entire PCI1515 core, including the serial-bus state machine (see Section 3.8.5, Suspend Mode, for details on using SUSPEND).

The PCI1515 controller provides a two-line serial-bus host controller that can interface to a serial EEPROM. See Section 3.6, *Serial EEPROM Interface*, for details on the two-wire serial-bus controller and applications.

3.5 PC Card Applications

The PCI1515 controller supports all the PC Card features and applications as described below.

- Card insertion/removal and recognition per the PC Card Standard (release 8.1)
- Speaker and audio applications
- LED socket activity indicators
- PC Card controller programming model
- CardBus socket registers

3.5.1 PC Card Insertion/Removal and Recognition

The *PC Card Standard* (release 8.1) addresses the card-detection and recognition process through an interrogation procedure that the socket must initiate on card insertion into a cold, nonpowered socket. Through this interrogation, card voltage requirements and interface (16-bit versus CardBus) are determined.

The scheme uses the card-detect and voltage-sense signals. The configuration of these four terminals identifies the card type and voltage requirements of the PC Card interface.

3.5.2 Low Voltage CardBus Card Detection

The card detection logic of the PCI1515 controller includes the detection of Cardbus cards with $V_{CC} = 3.3 \text{ V}$ and $V_{PP} = 1.8 \text{ V}$. The reporting of the 1.8-V CardBus card ($V_{CC} = 3.3 \text{ V}$, $V_{PP} = 1.8 \text{ V}$) is reported through the socket present state register as follows based on bit 10 (12V_SW_SEL) in the general control register (PCI offset 86h, see Section 4.30):

- If the 12V_SW_SEL bit is 0 (TPS2228 is used), then the 1.8-V CardBus card causes the 3VCARD bit in the socket present state register to be set.
- If the 12V_SW_SEL bit is 1 (TPS2226A is used), then the 1.8-V CardBus card causes the XVCARD bit in the socket present state register to be set.

3.5.3 Card Detection

The PCI1515 controller is capable of detecting USB custom cards as defined by the *PC Card Standard*. The detection of these devices is made possible through circuitry included in the PCI1515 controller and the adapters used to interface these devices with the PC Card/CardBus socket. No additional hardware requirements are placed on the system designer in order to support these devices.

The *PC Card Standard* addresses the card detection and recognition process through an interrogation procedure that the socket must initiate upon card insertion into a cold, unpowered socket. Through this interrogation, card voltage requirements and interface type (16-bit vs. CardBus) are determined. The scheme uses the CD1, CD2, VS1, and VS2 signals (CCD1, CCD2, CVS1, CVS2 for CardBus). A PC Card designer connects these four terminals in a certain configuration to indicate the type of card and its supply voltage requirements. The encoding scheme for this, defined in the *PC Card Standard*, is shown in Table 3–1.

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

CD2//CCD2	CD1//CCD1	VS2//CVS2	VS1//CVS1	Key	Interface	v _{cc}	V _{PP} /V _{CORE}
Ground	Ground	Open	Open	5 V	16-bit PC Card	5 V	Per CIS (V _{PP})
Ground	Ground	Open	Ground	5 V	16-bit PC Card	5 V and 3.3 V	Per CIS (V _{PP})
Ground	Ground	Ground	Ground	5 V	16-bit PC Card	5 V, 3.3 V, and X.X V	Per CIS (Vpp)
Ground	Ground	Open	Ground	LV	16-bit PC Card	3.3 V	Per CIS (V _{PP})
Ground	Connect to CVS1	Open	Connect to CCD1	LV	CardBus PC Card	3.3 V	Per CIS (V _{PP})
Ground	Ground	Ground	Ground	LV	16-bit PC Card	3.3 V and X.X V	Per CIS (V _{PP})
Connect to CVS2	Ground	Connect to CCD2	Ground	LV	CardBus PC Card	3.3 V and X.X V	Per CIS (V _{PP})
Connect to CVS1	Ground	Ground	Connect to CCD2	LV	CardBus PC Card	3.3 V, X.X V, and Y.Y V	Per CIS (Vpp)
Ground	Ground	Ground	Open	LV	16-bit PC Card	X.X V	Per CIS (V _{PP})
Connect to CVS2	Ground	Connect to CCD2	Open	LV	CardBus PC Card	3.3 V	1.8 V (V _{CORE})
Ground	Connect to CVS2	Connect to CCD1	Open	LV	CardBus PC Card	X.X V and Y.Y V	Per CIS (V _{PP})
Connect to CVS1	Ground	Open	Connect to CCD2	LV	CardBus PC Card	Y.Y V	Per CIS (V _{PP})
Ground	Connect to CVS1	Ground	Connect to CCD1	LV	Custom Card	Per query t	erminals
Ground	Connect to CVS2	Connect to CCD1	Ground		Reserved	t	Reserved

3.5.4 Power Switch Interface

The power switch interface of the PCI1515 controller is a 3-pin serial interface. This 3-pin interface is implemented such that the PCI1515 controller can connect to the TPS2228, TPS2226A, TPS2224A, TPS2223A, and TPS2220A power switches. Bit 10 (12V_SW_SEL) in the general control register (PCI offset 86h, see Section 4.30) selects the power switch that is implemented. The PCI1515 controller defaults to use the control logic for the TPS2228 power switch. See Table 3–2 through Table 3–5 for the power switch control logic. The TPS2224A, TPS2223A, and TPS2220A power switches have similar power control logic as the TPS2226A power switch. Refer to SLVS428A for details.

Table 3-2. TPS2228 Control Logic—xVPP/VCORE

AVPP/VC0	ORE CONT	ROL SIGN	ALS	OUTPUT	BVPP/VCORE CONTROL SIGNALS				OUTPUT
D8(SHDN)	D0	D1	D9	V_AVPP/VCORE	D8(SHDN)	D4	D5	D10	V_BVPP/VCORE
1	0	0	Х	0 V	1	0	0	Х	0 V
1	0	1	0	3.3 V	1	0	1	0	3.3 V
1	0	1	1	5 V	1	0	1	1	5 V
1	1	0	Х	Hi-Z	1	1	0	Х	Hi-Z
1	1	1	0	Hi-Z	1	1	1	0	Hi-Z
1	1	1	1	1.8 V	1	1	1	1	1.8 V
0	Х	Х	Х	Hi-Z	0	Х	Х	Х	Hi-Z

Table 3-3. TPS2228 Control Logic—xVCC

AVCC	AVCC CONTROL SIGNALS			BVCC CC	NTROL SIGN	OUTPUT	
D8(SHDN)	D3	D2	V_AVCC	D8(SHDN)	D6	D7	V_BVCC
1	0	0	0 V	1	0	0	0 V
1	0	1	3.3 V	1	0	1	3.3 V
1	1	0	5 V	1	1	0	5 V
1	1	1	0 V	1	1	1	0 V
0	Х	Х	Hi-Z	0	Х	Х	Hi-Z

Table 3-4. TPS2226A Control Logic—xVPP

AVPF	CONTROL	SIGNALS		OUTPUT						
D8(SHDN)	D0	D1	D9	V_AVPP	D8(SHDN)	D4	D5	D10	V_BVPP	
1	0	0	Х	0 V	1	0	0	Х	0 V	
1	0	1	0	3.3 V	1	0	1	0	3.3 V	
1	0	1	1	5 V	1	0	1	1	5 V	
1	1	0	Х	12 V	1	1	0	Х	12 V	
1	1	1	Х	Hi-Z	1	1	1	Х	Hi-Z	
0	Х	Х	Х	Hi-Z	0	Х	Х	Х	Hi-Z	

Table 3-5. TPS2226A Control Logic—xVCC

AVCC	CONTROL SIG	NALS	OUTPUT	BVCC CC	BVCC CONTROL SIGNALS			
D8(SHDN)	D3	D2	V_AVCC	D8(SHDN)	D6	D7	V_BVCC	
1	0	0	0 V	1	0	0	0 V	
1	0	1	3.3 V	1	0	1	3.3 V	
1	1	0	5 V	1	1	0	5 V	
1	1	1	0 V	1	1	1	0 V	
0	X	Χ	Hi-Z	0	Х	Χ	Hi-Z	

3.5.5 Internal Ring Oscillator

The internal ring oscillator provides an internal clock source for the PCI1515 controller so that neither the PCI clock nor an external clock is required in order for the PCI1515 controller to power down a socket or interrogate a PC Card. This internal oscillator, operating nominally at 16 kHz, is always enabled.

3.5.6 Integrated Pullup Resistors for PC Card Interface

The *PC Card Standard* requires pullup resistors on various terminals to support both CardBus and 16-bit PC Card configurations. The PCI1515 controller has integrated all of these pullup resistors and requires no additional external components. The I/O buffer on the BVD1(STSCHG)/CSTSCHG terminal has the capability to switch to an internal pullup resistor when a 16-bit PC Card is inserted, or switch to an internal pulldown resistor when a CardBus card is inserted. This prevents inadvertent CSTSCHG events.

3.5.7 SPKROUT and CAUDPWM Usage

The SPKROUT terminal carries the digital audio signal from the PC Card to the system. When a 16-bit PC Card is configured for I/O mode, the BVD2 terminal becomes the SPKR input terminal from the card. This terminal, in CardBus applications, is referred to as CAUDIO. SPKR passes a TTL-level binary audio signal to the PCI1515 controller. The CardBus CAUDIO signal also can pass a single-amplitude binary waveform as well as a PWM signal. The binary audio signal from the PC Card socket is enabled by bit 1 (SPKROUTEN) of the card control register (PCI offset 91h, see Section 4.37).

Older controllers support CAUDIO in binary or PWM mode, but use the same output terminal (SPKROUT). Some audio chips may not support both modes on one terminal and may have a separate terminal for binary and PWM. The PCI1515 implementation includes a signal for PWM, CAUDPWM, which can be routed to an MFUNC terminal. Bit 2 (AUD2MUX), located in the card control register, is programmed to route a CardBus CAUDIO PWM terminal to CAUDPWM. See Section 4.35, *Multifunction Routing Register*, for details on configuring the MFUNC terminals.

Figure 3-4 illustrates the SPKROUT connection.

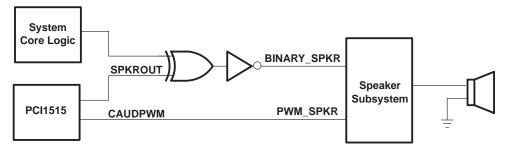


Figure 3-4. SPKROUT Connection to Speaker Driver

3.5.8 LED Socket Activity Indicators

The socket activity LED is provided to indicate when a PC Card is being accessed. The LEDA1 signal can be routed to the multifunction terminals. When configured for LED output, this terminal outputs an active high signal to indicate socket activity. LEDA1 indicates socket A (card A) activity. The LED_SKT output also indicates socket activity for socket A for compatibility with two socket controllers. See Section 4.35, *Multifunction Routing Status Register*, for details on configuring the multifunction terminals.

The active-high LED signal is driven for 64 ms. When the LED is not being driven high, it is driven to a low state. Either of the two circuits shown in Figure 3–5 can be implemented to provide LED signaling, and the board designer must implement the circuit that best fits the application.

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

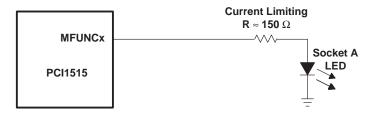


Figure 3-5. Sample LED Circuit

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 SUSPEND signal is asserted, when the PCI clock is to be stopped during the clock run protocol, or when in the D2 or D1 power state.

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

3.5.9 CardBus Socket Registers

The PCI1515 controller contains all registers for compatibility with the *PCI Local Bus Specification* and the *PC Card Standard*. These registers, which exist as the CardBus socket registers, are listed in Table 3–6.

Table 3-6. CardBus Socket Registers

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

3.6 Serial EEPROM Interface

The PCI1515 controller has a dedicated serial bus interface that can be used with an EEPROM to load certain registers in the PCI1515 controller. The EEPROM is detected by a pullup resistor on the SCL terminal. See Table 3–8 for the EEPROM loading map.

3.6.1 Serial-Bus Interface Implementation

The PCI1515 controller drives SCL at nearly 100 kHz during data transfers, which is the maximum specified frequency for standard mode I²C. The serial EEPROM must be located at address A0h.

Some serial device applications may include PC Card power 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 Accessing Serial-Bus Devices Through Software

The PCI1515 controller 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 3–7 lists the registers used to program a serial-bus device through software.

Table 3–7. PCI1515 Registers Used to Program Serial-Bus Devices

PCI OFFSET	REGISTER NAME	DESCRIPTION
B0h Serial-bus data Contains the data byte to send of		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.
		Write transactions to this register initiate a serial-bus transaction. The slave device address and the R/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.

3.6.3 Serial-Bus Interface Protocol

The SCL and SDA signals are bidirectional, open-drain signals and require pullup resistors as shown in Figure 3–3. The PCI1515 controller, which supports up to 100-Kb/s data-transfer rate, is compatible with standard mode I²C using 7-bit addressing.

All data transfers are initiated by the serial bus master. The beginning of a data transfer is indicated by a start condition, which is signaled when the SDA line transitions to the low state while SCL is in the high state, as shown in Figure 3–6. The end of a requested data transfer is indicated by a stop condition, which is signaled by a low-to-high transition of SDA while SCL is in the high state, as shown in Figure 3–6. 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.

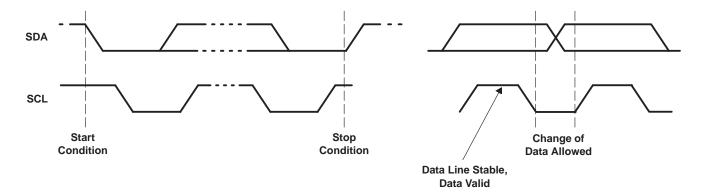


Figure 3-6. Serial-Bus Start/Stop Conditions and Bit Transfers

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 3–7 illustrates the acknowledge protocol.

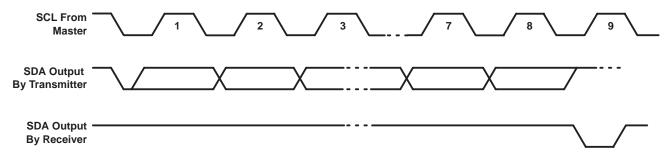


Figure 3-7. Serial-Bus Protocol Acknowledge

The PCI1515 controller is a serial bus master; all other devices connected to the serial bus external to the PCI1515 controller are slave devices. As the bus master, the PCI1515 controller 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 PCI1515 controller 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 3.6.4, *Serial-Bus EEPROM Application*, for details on how the PCI1515 controller automatically loads the subsystem identification and other register defaults through a serial-bus EEPROM.

Figure 3–8 illustrates a byte write. The PCI1515 controller issues a start condition and sends the 7-bit slave device address and the command bit zero. A 0 in the R/W command bit indicates that the data transfer is a write. The slave device acknowledges if it recognizes the address. If no acknowledgment is received by the PCI1515 controller, then an appropriate status bit is set in the serial-bus control/status register (PCI offset B3h, see Section 4.49). The word address byte is then sent by the PCI1515 controller, and another slave acknowledgment is expected. Then the PCI1515 controller delivers the data byte MSB first and expects a final acknowledgment before issuing the stop condition.

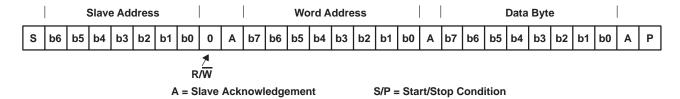


Figure 3-8. Serial-Bus Protocol—Byte Write

Figure 3–9 illustrates a byte read. The read protocol is very similar to the write protocol, except the R/W command bit must be set to 1 to indicate a read-data transfer. In addition, the PCI1515 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 PCI1515 master.

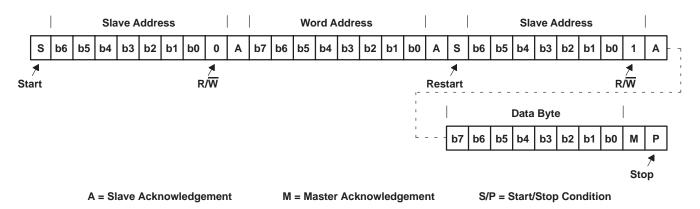


Figure 3-9. Serial-Bus Protocol—Byte Read

Figure 3–10 illustrates EEPROM interface doubleword data collection protocol.

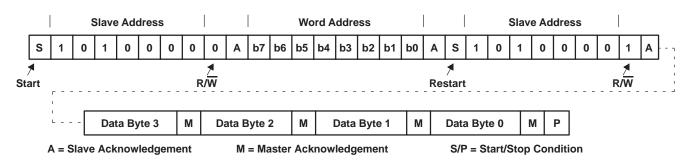


Figure 3-10. EEPROM Interface Doubleword Data Collection

3.6.4 Serial-Bus EEPROM Application

When the PCI bus is reset and the serial-bus interface is detected, the PCI1515 controller attempts to read the subsystem identification and other register defaults from a serial EEPROM.

This format must be followed for the PCI1515 controller to load initializations from a serial EEPROM. All bit fields must be considered when programming the EEPROM.

The serial EEPROM is addressed at slave address 1010 000b by the PCI1515 controller. All hardware address bits for the EEPROM must be tied to the appropriate level to achieve this address. The serial EEPROM chip in the sample application (Figure 3–10) 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.

Table 3-8. EEPROM Loading Map

SERIAL ROM OFFSET		BYTE DESCRIPTION							
00h		CardBus function indicator (00h)							
01h		Number of bytes (20h)							
		PCI 04h, command register, function 0, bits 8, 6–5, 2–0							
02h	[7] Command register, bit 8								
			Cl 04h, command r	egister, function					
03h	[7] Command register, bit 8	[6] Command register, bit 6	[5] Command register, bit 5	[4:3] RSVD	[2] Command register, bit 2	[1] Command register, bit 1	[0] Command register, bit 0		
04h			PCI 40h, sub	system vend	or ID, byte 0				
05h			PCI 41h, sub	osystem vend	or ID, byte 1				
06h			PCI 42h,	subsystem II	D, byte 0				
07h			PCI 43h,	subsystem II), byte 1				
08h		PCI 44h, PC C	ard 16-bit I/F legac	y mode base	address register, b	yte 0, bits 7–1			
09h		PCI 45h, F	PC Card 16-bit I/F I	egacy mode b	oase address regis	ter, byte 1			
0Ah		PCI 46h, PC Card 16-bit I/F legacy mode base address register, byte 2							
0Bh		PCI 47h, PC Card 16-bit I/F legacy mode base address register, byte 3							
0Ch		PCI 80h, system control, function 0, byte 0, bits 6-0							
0Dh		PCI 80h, system control, function 1, byte 0, bit 2							
0Eh		PCI 81h, system control, byte 1							
0Fh		R	eserved load all 0s	(PCI 82h, sys	stem control, byte 2	2)			
10h			PCI 83h,	system contro	ol, byte 3				
11h			PCI 8Ch,	MFUNC routing	ng, byte 0				
12h			PCI 8Dh,	MFUNC routing	ng, byte 1				
13h			PCI 8Eh,	MFUNC routir	ng, byte 2				
14h			PCI 8Fh,	MFUNC routir	ng, byte 3				
15h			PCI 90h	, retry status,	bits 7, 6				
16h			PCI 91	h, card contro	ol, bit 7				
17h			PCI 92h, de	vice control, b	its 6, 5, 3–0				
18h				diagnostic, bi					
19h	PCI A2h, pc	ower-management	capabilities, function	on 0, bit 15 (bi	t 7 of EEPROM off	fset 16h correspon	ds to bit 15)		
1Ah	PCI A2h, pc	ower-management	capabilities, function	on 1, bit 15 (bi	t 7 of EEPROM off	fset 16h correspon	ds to bit 15)		
1Bh	CB Sock	CB Socket + 0Ch, function 0 socket force event, bit 27 (bit 3 of EEPROM offset 17h corresponds to bit 27)							
1Ch	CB Sock	et + 0Ch, function	I socket force ever	nt, bit 27 (bit 3	of EEPROM offse	t 18h corresponds	to bit 27)		
1Dh		E	xCA 00h, ExCA id	entification an	nd revision, bits $7-0$	0			
1Eh			PCI 86h, gene	eral control, by	/te 0, bits 7-0				
1Fh			PCI 87h, genera	I control, byte	1, bits 7, 6, 4–0				
20h			PCI 89h, G	PE enable, bit	ts 7, 6, 4–0				
21h			PCI 8Bh, gene	ral-purpose o	utput, bits 4-0				
22h			End-o	f-list indicator	(80h)				

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 PCI1515 controller. The PCI1515 controller provides several interrupt signaling schemes to accommodate the needs of a variety of platforms. The different mechanisms for dealing with interrupts in this controller 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 PCI1515 controller is, therefore, backward compatible with existing interrupt control register definitions, and new registers have been defined where required.

The PCI1515 controller 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 PCI1515 controller, PC Card interrupts are classified either as card status change (CSC) or as functional interrupts.

The method by which any type of PCI1515 interrupt is communicated to the host interrupt controller varies from system to system. The PCI1515 controller offers system designers the choice of using parallel PCI interrupt signaling, parallel ISA-type IRQ interrupt signaling, or the IRQSER serialized ISA and/or PCI interrupt protocol. It is possible to use the parallel PCI interrupts in combination with either parallel IRQs or serialized IRQs, as detailed in the sections that follow. All interrupt signaling is provided through the seven multifunction terminals, MFUNC0–MFUNC6.

3.7.1 PC Card Functional and Card Status Change Interrupts

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

Card status change (CSC)-type interrupts are defined as events at the PC Card interface that are detected by the PCI1515 controller and may warrant notification of host card and socket services software for service. CSC events include both card insertion and removal from the PC Card socket, 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	
40 1.11	Battery conditions (BVD1, BVD2)	ExCA offset 05h/805h bits 1 and 0	ExCA offset 04h/804h bits 1 and 0	
16-bit memory	Wait states (READY)	ExCA offset 05h/805h bit 2	ExCA offset 04h/804h bit 2	
16-bit I/O	Change in card status (STSCHG)	ExCA offset 05h/805h bit 0	ExCA offset 04h/804h bit 0	
16-bit I/O	Interrupt request (IREQ)	Always enabled	PCI configuration offset 91h bit 0	
All 16-bit PC Cards/ Smart Card adapters	Power cycle complete	ExCA offset 05h/805h bit 3	ExCA offset 04h/804h bit 3	
	Change in card status (CSTSCHG)	Socket mask bit 0	Socket event bit 0	
CardBus	Interrupt request (CINT)	Always enabled	PCI configuration offset 91h bit 0	
Calubus	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	
	Battery conditions	000	BVD1(STSCHG)//CSTSCHG	A transition on BVD1 indicates a change in the PC Card battery conditions.	
16-bit memory	(BVD1, BVD2)	CSC	BVD2(SPKR)//CAUDIO	A transition on BVD2 indicates a change in the PC Card battery conditions.	
memory	Wait states (READY)	CSC	READY(IREQ)//CINT	A transition on READY indicates a change in the ability of the memory PC Card to accept or provide data.	
16-bit I/O	16-bit I/O Change in card status (STSCHG) CS		BVD1(STSCHG)//CSTSCHG	The assertion of STSCHG indicates a status change on the PC Card.	
16-bit I/O	Interrupt request (IREQ) Functional		READY(IREQ)//CINT	The assertion of IREQ indicates an interrupt request from the PC Card.	
O and D and	Change in card status (CSTSCHG)	CSC	BVD1(STSCHG)//CSTSCHG	The assertion of CSTSCHG indicates a status change on the PC Card.	
CardBus	Interrupt request (CINT)	Functional	READY(IREQ)//CINT	The assertion of CINT indicates an interrupt request from the PC Card.	
All PC Cards/ Smart Card	Card insertion or removal	CSC	CD1//CCD1, CD2//CCD2	A transition on either CD1//CCD1 or CD2//CCD2 indicates an insertion or removal of a 16-bit or CardBus PC Card.	
adapters	Power cycle complete	CSC	N/A	An interrupt is generated when a PC Card power-up cycle has completed.	

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

The 1997 PC Card Standard describes the power-up sequence that must be followed by the PCI1515 controller 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 PCI1515 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 PCI1515 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 PCI1515 controller. By individually masking the interrupt sources listed, software can control those events that cause a PCI1515 interrupt. Host software has some control over the system interrupt the PCI1515 controller asserts by programming the appropriate routing registers. The PCI1515 controller 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 PCI1515 controller, the interrupt service routine must determine which of the events listed in Table 3–9 caused the interrupt. Internal registers in the PCI1515 controller 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 PCI1515 controller 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 must never be a card interrupt that does not require service after proper initialization.

Table 3–9 lists the various methods of clearing the interrupt flag bits. The flag bits in the ExCA registers (16-bit PC Card-related interrupt flags) can be cleared using two different methods. One method is an explicit write of 1 to the flag bit to clear and the other is by reading the flag bit register. The selection of flag bit clearing methods is made by bit 2 (IFCMODE) in the ExCA global control register (ExCA offset 1Eh/81Eh, see Section 5.20), and defaults to the flag-cleared-on-read method.

The CardBus-related interrupt flags can be cleared by an explicit write of 1 to the interrupt flag in the socket event register (see Section 6.1). Although some of the functionality is shared between the CardBus registers and the ExCA registers, software must 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 PCI1515 controller can be routed to obtain a subset of the ISA IRQs. The IRQ choices provide ultimate flexibility in PC Card host interruptions. To use the parallel ISA-type IRQ interrupt signaling, software must program the device control register (PCI offset 92h, see Section 4.38), to select the parallel IRQ signaling scheme. See Section 4.35, *Multifunction Routing Status Register*, for details on configuring the multifunction terminals.

A system using parallel IRQs requires (at a minimum) one PCI terminal, INTA, to signal CSC events. This requirement is dictated by certain card and socket-services software. The INTA requirement calls for routing the MFUNC0 terminal for INTA signaling. 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, IRQ9, IRQ10, and IRQ15. The multifunction routing status register must be programmed to a value of 0A9F 5432h. This value routes the MFUNC0 terminal to INTA signaling and routes the remaining terminals as illustrated in Figure 3–11. Not shown is that INTA must also be routed to the programmable interrupt controller (PIC), or to some circuitry that provides parallel PCI interrupts to the host.

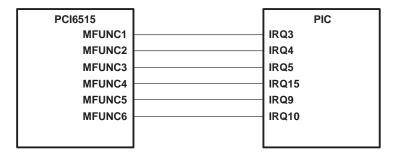


Figure 3-11. IRQ Implementation

Power-on software is responsible for programming the multifunction routing status register to reflect the IRQ configuration of a system implementing the PCI1515 controller. The multifunction routing status register is a global register that is shared between the four PCI1515 functions. See Section 4.35, *Multifunction Routing Status Register*, for details on configuring the multifunction terminals.

The parallel ISA-type IRQ signaling from the MFUNC6–MFUNC0 terminals is compatible with the input signal requirements of the 8259 PIC. The parallel IRQ option is provided for system designs that require legacy ISA IRQs. Design constraints may demand more MFUNC6–MFUNC0 IRQ terminals than the PCI1515 controller makes available.

3.7.4 Using Parallel PCI Interrupts

Parallel PCI interrupts are available when exclusively in parallel PCI interrupt/parallel ISA IRQ signaling mode, and when only IRQs are serialized with the IRQSER protocol. The INTA interrupt signal is routed to the MFUNC0 terminal.

3.7.5 Using Serialized IRQSER Interrupts

The serialized interrupt protocol implemented in the PCI1515 controller 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 INTA, INTB, INTC, and INTD. For details on the IRQSER protocol, refer to the document *Serialized IRQ Support for PCI Systems*.

3.7.6 SMI Support in the PCI1515 Controller

The PCI1515 controller provides a mechanism for interrupting the system when power changes have been made to the PC Card socket interface. The interrupt mechanism is designed to fit into a system maintenance interrupt (SMI) scheme. SMI interrupts are generated by the PCI1515 controller, when enabled, after a write cycle to the socket control register (CB offset 10h, see Section 6.5) of the CardBus register set, or the ExCA power control register (ExCA offset 02h/802h, see Section 5.3) causes a power cycle change sequence to be sent on the power switch interface.

The SMI control is programmed through three bits in the system control register (PCI offset 80h, see Section 4.29). These bits are SMIROUTE (bit 26), SMISTATUS (bit 25), and SMIENB (bit 24). Table 3–11 describes the SMI control bits function.

BIT NAME

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.

Table 3-11. SMI Control

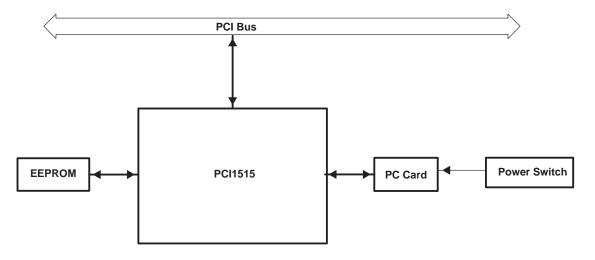
The CSC interrupt can be either level or edge mode, depending upon the CSCMODE bit in the ExCA global control register (ExCA offset 1Eh/81Eh, see Section 5.20).

If IRQ2 is selected by SMIROUTE, then the IRQSER signaling protocol supports SMI signaling in the IRQ2 IRQ/Data slot. In a parallel ISA IRQ system, the support for an active low IRQ2 is provided only if IRQ2 is routed to either MFUNC3 or MFUNC6 through the multifunction routing status register (PCI offset 8Ch, see Section 4.35).

3.8 Power Management Overview

In addition to the low-power CMOS technology process used for the PCI1515 controller, various features are designed into the controller to allow implementation of popular power-saving techniques. These features and techniques are as follows:

- Clock run protocol
- Cardbus PC Card power management
- 16-bit PC Card power management
- Suspend mode
- Ring indicate
- PCI power management
- Cardbus bridge power management
- ACPI support



[†] The system connection to $\overline{\mathsf{GRST}}$ is implementation-specific. $\overline{\mathsf{GRST}}$ must be asserted on initial power up of the PCI1515 controller. $\overline{\mathsf{PRST}}$ must be asserted for subsequent warm resets.

Figure 3-12. System Diagram Implementing CardBus Device Class Power Management

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

The PCI1515 controller requires 1.5-V core voltage. The core power can be supplied by the PCI1515 controller itself using the internal LDO-VR. The core power can alternatively be supplied by an external power supply through the VR_PORT terminal. Table 3–12 lists the requirements for both the internal core power supply and the external core power supply.

VR EN SUPPLY VCC VR_PORT NOTE Internal 1.5-V LDO-VR is enabled. A 1.0-μF bypass capacitor is required on the VR_PORT Internal 3.3 V 1.5-V output terminal for decoupling. This output is not for external use. Vcc Internal 1.5-V LDO-VR is disabled. An external 1.5-V power supply, of minimum 50-mA External 3.3 V 1.5-V input capacity, is required. A 0.1-μF bypass capacitor on the VR_PORT terminal is required.

Table 3–12. Requirements for Internal/External 1.5-V Core Power Supply

3.8.2 CardBus (Function 0) Clock Run Protocol

The PCI CLKRUN feature is the primary method of power management on the PCI interface of the PCI1515 controller. CLKRUN signaling is provided through the MFUNC6 terminal. Since some chip sets do not implement CLKRUN, this is not always available to the system designer, and alternate power-saving features are provided. For details on the CLKRUN protocol see the PCI Mobile Design Guide.

The PCI1515 controller does not permit the central resource to stop the PCI clock under any of the following conditions:

- Bit 1 (KEEPCLK) in the system control register (PCI offset 80h, see Section 4.29) is set.
- The 16-bit PC Card resource manager is busy.
- The PCI1515 CardBus master state machine is busy. A cycle may be in progress on CardBus.
- The PCI1515 master is busy. There may be posted data from CardBus to PCI in the PCI1515 controller.
- Interrupts are pending.
- The CardBus CCLK for the socket has not been stopped by the PCI1515 CCLKRUN manager.
- PC Card interrogation is in progress.

The PCI1515 controller restarts the PCI clock using the CLKRUN protocol under any of the following conditions:

- A 16-bit PC Card IREQ or a CardBus CINT has been asserted by either card.
- A CardBus CBWAKE (CSTSCHG) or 16-bit PC Card STSCHG/RI event occurs in the socket.
- A CardBus attempts to start the CCLK using CCLKRUN.
- A CardBus card arbitrates for the CardBus bus using CREQ.
- Bit 1 (KEEPCLK) in the system control register (PCI offset 80h, see Section 4.29) is set.
- Data is in any of the FIFOs (receive or transmit).
- The master state machine is busy.
- There are pending interrupts.

3.8.3 CardBus PC Card Power Management

The PCI1515 controller implements its own card power-management engine that can turn off the CCLK to a socket when there is no activity to the CardBus PC Card. The PCI clock-run protocol is followed on the CardBus CCLKRUN interface to control this clock management.

3.8.4 16-Bit PC Card Power Management

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

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

3.8.5 Suspend Mode

The SUSPEND signal, provided for backward compatibility, gates the PRST (PCI reset) signal and the GRST (global reset) signal from the PCI1515 controller. Besides gating PRST and GRST, SUSPEND also gates PCLK inside the PCI1515 controller in order to minimize power consumption.

It should also be noted that asynchronous signals, such as card status change interrupts and RI_OUT, can be passed to the host system without a PCI clock. However, if card status change interrupts are routed over the serial interrupt stream, then the PCI clock must be restarted in order to pass the interrupt, because neither the internal oscillator nor an external clock is routed to the serial-interrupt state machine. Figure 3–13 is a signal diagram of the suspend function.

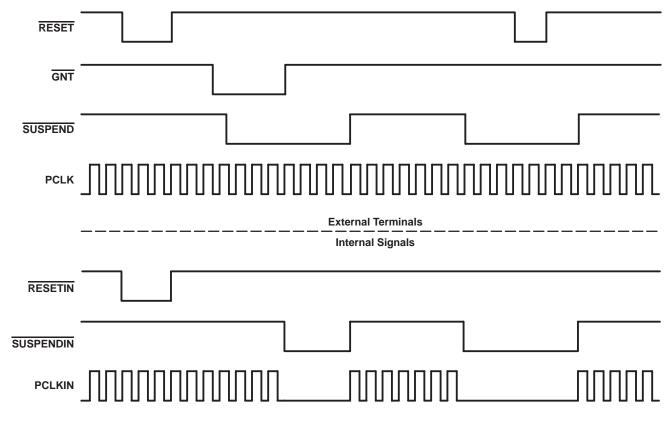


Figure 3-13. Signal Diagram of Suspend Function

3.8.6 Requirements for Suspend Mode

The suspend mode prevents the clearing of all register contents on the assertion of reset (PRST or GRST) which would require the reconfiguration of the PCI1515 controller by software. Asserting the SUSPEND signal places the PCI outputs of the controller in a high-impedance state and gates the PCLK signal internally to the controller unless a PCI transaction is currently in process (GNT is asserted). It is important that the PCI bus not be parked on the PCI1515 controller when SUSPEND is asserted because the outputs are in a high-impedance state.

The GPIOs, MFUNC signals, and RI_OUT signal are all active during SUSPEND, unless they are disabled in the appropriate PCI1515 registers.

3.8.7 Ring Indicate

The RI_OUT output is an important feature in power management, allowing a system to go into a suspended mode and wake-up on modem rings and other card events. TI-designed flexibility permits this signal to fit wide platform requirements. RI_OUT on the PCI1515 controller can be asserted under any of the following conditions:

- A 16-bit PC Card modem in a powered socket asserts \overline{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 3–14 shows various enable bits for the PCI1515 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.

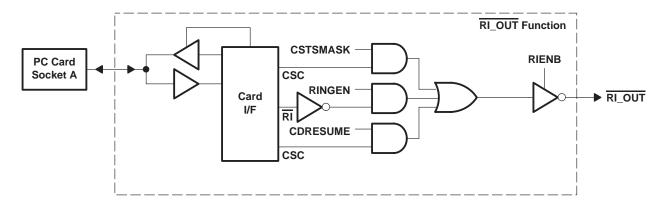


Figure 3–14. RI_OUT Functional Diagram

RI from the 16-bit PC Card interface is masked by bit 7 (RINGEN) in the ExCA interrupt and general control register (ExCA offset 03h/803h, see Section 5.4). This is only applicable when a 16-bit card is powered in the socket.

The CBWAKE signaling to $\overline{\text{RI_OUT}}$ is enabled through the same mask as the CSC event for CSTSCHG. The mask bit (bit 0, CSTSMASK) is programmed through the socket mask register (CB offset 04h, see Section 6.2) in the CardBus socket registers.

RI_OUT can be routed through any of three different pins, RI_OUT/PME, MFUNC2, or MFUNC4. The RI_OUT function is enabled by setting bit 7 (RIENB) in the card control register (PCI offset 91h, see Section 4.37). The PME function is enabled by setting bit 8 (PME_ENABLE) in the power-management control/status register (PCI offset A4h, see Section 4.43). When bit 0 (RIMUX) in the system control register (PCI offset 80h, see Section 4.29) is set to 0, both the RI_OUT function and the PME function are routed to the RI_OUT/PME terminal. If both functions are enabled and RIMUX is set to 0, then the RI_OUT/PME terminal becomes RI_OUT only and PME assertions are never seen. Therefore, in a system using both the RI_OUT function and the PME function, RIMUX must be set to 1 and RI_OUT must be routed to either MFUNC2 or MFUNC4.

3.8.8 PCI Power Management (Function 0)

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

The seven power-management states of PCI functions are:

- D0-uninitialized Before controller configuration, controller not fully functional
- D0-active Fully functional state
- D1 Low-power state
- D2 Low-power state
- D3_{hot} Low-power state. Transition state before D3_{cold}
- D3_{cold} PME signal-generation capable. Main power is removed and VAUX is available.
- D3_{off} No power and completely nonfunctional
- NOTE 1: In the D0-uninitialized state, the PCI1515 controller does not generate $\overline{\text{PME}}$ and/or interrupts. When bits 0 (IO_EN) and 1 (MEM_EN) of the command register (PCI offset 04h, see Section 4.4) are both set, the PCI1515 controller switches the state to D0-active. Transition from D3_{COId} to the D0-uninitialized state happens at the deassertion of $\overline{\text{PRST}}$. The assertion of $\overline{\text{GRST}}$ forces the controller to the D0-uninitialized state immediately.
- NOTE 2: The PWR_STATE bits (bits 1–0) of the power-management control/status register (PCI offset A4h, see Section 4.43) only code for four power states, D0, D1, D2, and D3hot. The differences between the three D3 states is invisible to the software because the controller is not accessible in the D3cold or D3off state.

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

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

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

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

The capabilities pointer provides access to the first item in the linked list of capabilities. For the PCI1515 controller, a CardBus bridge with PCI configuration space header type 2, the capabilities pointer is mapped to an offset of 14h. The first byte of each capability register block is required to be a unique ID of that capability. PCI power management has been assigned an ID of 01h. The next byte is a pointer to the next pointer item in the list of capabilities. If there are no more items in the list, then the next item pointer must be set to 0. The registers following the next item pointer are specific to the capability of the function. The PCI power-management capability implements the register block outlined in Table 3–13.

Table 3–13. Power-Management Registers

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

The power-management capabilities register (PCI offset A2h, see Section 4.42) provides information on the capabilities of the function related to power management. The power-management control/status register (PCI offset A4h, see Section 4.43) enables control of power-management states and enables/monitors power-management events. The data register is an optional register that can provide dynamic data.

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

3.8.9 CardBus Bridge Power Management

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

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

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

The Texas Instruments PCI1515 controller addresses these D3 wake-up issues in the following manner:

- Two resets are provided to handle preservation of PME context bits:
 - Global reset (GRST) is used only on the initial boot up of the system after power up. It places the PCI1515 controller in its default state and requires BIOS to configure the controller before becoming fully functional.
 - PCI reset (PRST) has dual functionality based on whether PME is enabled or not. If PME is enabled, then PME context is preserved. If PME is not enabled, then PRST acts the same as a normal PCI reset. Please see the master list of PME context bits in Section 3.8.11.

Power source in D3_{cold} if wake-up support is required from this state. Since V_{CC} is removed in D3_{cold}, an auxiliary power source must be supplied to the PCI1515 V_{CC} terminals. Consult the PCI14xx Implementation Guide for D3 Wake-Up or the PCI Power Management Interface Specification for PCI to CardBus Bridges for further information.

3.8.10 ACPI Support

The Advanced Configuration and Power Interface (ACPI) Specification provides a mechanism that allows unique pieces of hardware to be described to the ACPI driver. The PCI1515 controller offers a generic interface that is compliant with ACPI design rules.

Two doublewords of general-purpose ACPI programming bits reside in PCI1515 PCI configuration space at offset 88h. The programming model is broken into status and control functions. In compliance with ACPI, the top level event status and enable bits reside in the general-purpose event status register (PCI offset 88h, see Section 4.31) and general-purpose event enable register (PCI offset 89h, see Section 4.32). The status and enable bits are implemented as defined by ACPI and illustrated in Figure 3–15.

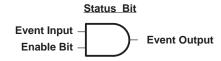


Figure 3-15. Block Diagram of a Status/Enable Cell

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

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

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

PME context bit means that the bit is cleared only by the assertion of GRST when the PME enable bit, bit 8 of the power management control/status register (PCI offset A4h, see Section 4.43) is set. If PME is not enabled, then these bits are cleared when either PRST or GRST is asserted.

The PME context bits (function 0) are:

- Bridge control register (PCI offset 3Eh, see Section 4.25): bit 6
- System control register (PCI offset 80h, see Section 4.29): bits 10–8
- Power management control/status register (PCI offset A4h, see Section 4.43): bit 15
- ExCA power control register (ExCA 802h, see Section 5.3): bits 7, 5 (82365SL mode only), 4, 3, 1, 0
- ExCA interrupt and general control (ExCA 803h, see Section 5.4): bits 6, 5
- ExCA card status-change register (ExCA 804h, see Section 5.5): bits 3–0
- ExCA card status-change interrupt configuration register (ExCA 805h, see Section 5.6): bits 3–0
- ExCA card detect and general control register (ExCA 816h, see Section 5.19): bits 7, 6
- Socket event register (CardBus offset 00h, see Section 6.1): bits 3–0
- Socket mask register (CardBus offset 04h, see Section 6.2): bits 3–0
- Socket present state register (CardBus offset 08h, see Section 6.3): bits 13-7, 5-1
- Socket control register (CardBus offset 10h, see Section 6.5): bits 6–4, 2–0

Global reset-only bits, as the name implies, are cleared only by \overline{GRST} . These bits are never cleared by \overline{PRST} , regardless of the setting of the \overline{PME} enable bit. The \overline{GRST} signal is gated only by the $\overline{SUSPEND}$ signal. This means that assertion of $\overline{SUSPEND}$ blocks the \overline{GRST} signal internally, thus preserving all register contents. Figure 3–12 is a diagram showing the application of \overline{GRST} and \overline{PRST} .

The global reset-only bits (function 0) are:

- Status register (PCI offset 06h, see Section 4.5): bits 15–11, 8
- Secondary status register (PCI offset 16h, see Section 4.14): bits 15–11, 8
- Subsystem vendor ID register (PCI offset 40h, see Section 4.26): bits 15–0
- Subsystem ID register (PCI offset 42h, see Section 4.27): bits 15–0
- PC Card 16-bit I/F legacy-mode base-address register (PCI offset 44h, see Section 4.28): bits 31–0
- System control register (PCI offset 80h, see Section 4.29): bits 31–24, 22–13, 11, 6–0
- General control register (PCI offset 86h, see Section 4.30): bits 13–10, 7, 5–3, 1, 0
- General-purpose event status register (PCI offset 88h, see Section 4.31): bits 7, 6, 4–0
- General-purpose event enable register (PCI offset 89h, see Section 4.32): bits 7, 6, 4–0
- General-purpose output register (PCI offset 8Bh, see Section 4.34): bits 4–0
- Multifunction routing register (PCI offset 8Ch, see Section 4.35): bits 31–0
- Retry status register (PCI offset 90h, see Section 4.36): bits 7–5, 3, 1
- Card control register (PCI offset 91h, see Section 4.37): bits 7, 2–0
- Device control register (PCI offset 92h, see Section 4.38): bits 7–5, 3–0
- Diagnostic register (PCI offset 93h, see Section 4.39): bits 7–0
- Power management capabilities register (PCI offset A2h, see Section 4.42): bit 15
- Power management CSR register (PCI offset A4h, see Section 4.43): bits 15, 8
- Serial bus data register (PCI offset B0h, see Section 4.46): bits 7–0
- Serial bus index register (PCI offset B1h, see Section 4.47): bits 7–0
- Serial bus slave address register (PCI offset B2h, see Section 4.48); bits 7–0
- Serial bus control/status register (PCI offset B3h, see Section 4.49): bits 7, 3–0
- ExCA identification and revision register (ExCA 800h, see Section 5.1): bits 7–0
- ExCA global control register (ExCA 81Eh, see Section 5.20): bits 2–0
- CardBus socket power management register (CardBus 20h, see Section 6.6): bits 25, 24

4 PC Card Controller Programming Model

This chapter describes the PCI1515 PCI configuration registers that make up the 256-byte PCI configuration header for each PCI1515 function.

Any bit followed by a \dagger is not cleared by the assertion of \overline{PRST} (see CardBus Bridge Power Management, Section 3.8.9, for more details) if \overline{PME} is enabled (PCI offset A4h, bit 8). In this case, these bits are cleared only by \overline{GRST} . If \overline{PME} is not enabled, then these bits are cleared by \overline{GRST} or \overline{PRST} . These bits are sometimes referred to as PME context bits and are implemented to allow \overline{PME} context to be preserved during the transition from D3hot or D3cold to D0.

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

A bit description table, typically included when the register contains bits of more than one type or purpose, indicates bit field names, a detailed field description, and field access tags which appear in the *type* column. Table 4–1 describes the field access tags.

ACCESS TAG	S TAG NAME MEANING						
R	Read	Field can be read by software.					
W	Write	Field can be written by software to any value.					
S	Set	Field can be set by a write of 1. Writes of 0 have no effect.					
С	Clear	Field can be cleared by a write of 1. Writes of 0 have no effect.					
U	Update	Field can be autonomously updated by the PCI1515 controller.					

Table 4-1. Bit Field Access Tag Descriptions

4.1 PCI Configuration Register Map (Function 0)

The PCI1515 controller is a single function PC Card controller (PCI function 0). The configuration header, compliant with the *PCI Local Bus Specification* as a CardBus bridge header, is *PC99/PC2001* compliant as well. Table 4–2 illustrates the PCI configuration register map, which includes both the predefined portion of the configuration space and the user-definable registers.

	Table 4-2. Function 0	PCI Configuration Regis	ster wap						
	REGISTE	R NAME		OFFSET					
Dev	ice ID	Vend	or ID	00h					
Sta	tus ‡	Comr	mand	04h					
	Class code		Revision ID	08h					
BIST	Header type	Latency timer	Cache line size	0Ch					
	CardBus socket registers/E	xCA base address register		10h					
Seconda	ry status ‡	Reserved	Capability pointer	14h					
CardBus latency timer	Subordinate bus number	CardBus bus number	PCI bus number	18h					
	CardBus memor	y base register 0		1Ch					
	CardBus memor	ry limit register 0		20h					
	CardBus memor	y base register 1		24h					
	CardBus memory limit register 1								

Table 4-2. Function 0 PCI Configuration Register Map

[‡] One or more bits in this register are cleared only by the assertion of GRST.

Table 4–2. Function 0 PCI Configuration Register Map (Continued)

	REGISTE	R NAME		OFFSET
	CardBus I/O ba	ase register 0		2Ch
	CardBus I/O li	mit register 0		30h
	CardBus I/O ba	ase register 1		34h
	CardBus I/O li	mit register 1		38h
Bridge	control †	Interrupt pin	Interrupt line	3Ch
Subsys	tem ID ‡	Subsystem	vendor ID ‡	40h
	PC Card 16-bit I/F legacy	y-mode base-address ‡		44h
	Rese	rved		48h-7Ch
	System co	ontrol †‡		80h
General	control ‡	Rese	rved	84h
General-purpose output ‡	General-purpose input	General-purpose event enable ‡	General-purpose event status ‡	88h
	Multifunction ro	outing status ‡		8Ch
Diagnostic ‡	Device control ‡	Card control ‡	Retry status ‡	90h
	Rese	rved		94h-9Ch
Power managen	nent capabilities ‡	Next item pointer	Capability ID	A0h
Power management data (Reserved)	Power management control/status bridge support extensions	Power managemen	nt control/status †‡	A4h
	Rese	rved		A8h-ACh
Serial bus control/status ‡	Serial bus slave address ‡	Serial bus index ‡	Serial bus data ‡	B0h
	Rese	rved		B4h-FCh

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

4.2 Vendor ID Register

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

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

Register: **Vendor ID**Offset: 00h (Function 0)

Type: Read-only Default: 104Ch

[‡] One or more bits in this register are cleared only by the assertion of GRST.

4.3 Device ID Register Function 0

This read-only register contains the device ID assigned by TI to the PCI1515 CardBus controller functions (PCI function 0).

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		Device ID—Smart Card enabled														
Туре	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Default	1	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0

Register: **Device ID**

Offset: 02h (Function 0)

Type: Read-only Default: 8036h

4.4 Command Register

The PCI command register provides control over the PCI1515 interface to the PCI bus. All bit functions adhere to the definitions in the *PCI Local Bus Specification* (see Table 4–3).

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

Register: Command

Offset: 04h

Type: Read-only, Read/Write

Default: 0000h

Table 4–3. Command Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–11	RSVD	R	Reserved. Bits 15–11 return 0s when read.
10	INT_DISABLE	RW	INTx disable. When set to 1, this bit disables the function from asserting interrupts on the INTx signals. 0 = INTx assertion is enabled (default) 1 = INTx assertion is disabled
9	FBB_EN	R	Fast back-to-back enable. The PCI1515 controller does not generate fast back-to-back transactions; therefore, this bit is read-only. This bit returns a 0 when read.
8	SERR_EN	RW	System error (SERR) enable. This bit controls the enable for the SERR driver on the PCI interface. SERR can be asserted after detecting an address parity error on the PCI bus. Both this bit and bit 6 must be set for the PCI1515 controller to report address parity errors. 0 = Disables the SERR output driver (default) 1 = Enables the SERR output driver
7	RSVD	R	Reserved. Bit 7 returns 0 when read.
6	PERR_EN	RW	Parity error response enable. This bit controls the PCI1515 response to parity errors through the PERR signal. Data parity errors are indicated by asserting PERR, while address parity errors are indicated by asserting SERR. 0 = PCI1515 controller ignores detected parity errors (default). 1 = PCI1515 controller responds to detected parity errors.
5	VGA_EN	RW	VGA palette snoop. When set to 1, palette snooping is enabled (i.e., the PCI1515 controller does not respond to palette register writes and snoops the data). When the bit is 0, the PCI1515 controller treats all palette accesses like all other accesses.
4	MWI_EN	R	Memory write-and-invalidate enable. This bit controls whether a PCI initiator device can generate memory write-and-invalidate commands. The PCI1515 controller does not support memory write-and-invalidate commands, it uses memory write commands instead; therefore, this bit is hardwired to 0. This bit returns 0 when read. Writes to this bit have no effect.
3	SPECIAL	R	Special cycles. This bit controls whether or not a PCI device ignores PCI special cycles. The PCI1515 controller does not respond to special cycle operations; therefore, this bit is hardwired to 0. This bit returns 0 when read. Writes to this bit have no effect.
2	MAST_EN	RW	Bus master control. This bit controls whether or not the PCI1515 controller can act as a PCI bus initiator (master). The PCI1515 controller can take control of the PCI bus only when this bit is set. 0 = Disables the PCI1515 ability to generate PCI bus accesses (default) 1 = Enables the PCI1515 ability to generate PCI bus accesses
1	MEM_EN	RW	Memory space enable. This bit controls whether or not the PCI1515 controller can claim cycles in PCI memory space. 0 = Disables the PCI1515 response to memory space accesses (default) 1 = Enables the PCI1515 response to memory space accesses
0	IO_EN	RW	I/O space control. This bit controls whether or not the PCI1515 controller can claim cycles in PCI I/O space. 0 = Disables the PCI1515 controller from responding to I/O space accesses (default) 1 = Enables the PCI1515 controller to respond to I/O space accesses

4.5 Status Register

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

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

Register: Status

Offset: 06h (Function 0)
Type: Read-only, Read/Write

Default: 0210h

Table 4-4. Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15 ‡	PAR_ERR	RW	Detected parity error. This bit is set when a parity error is detected, either an address or data parity error. Write a 1 to clear this bit.
14 ‡	SYS_ERR	RW	Signaled system error. This bit is set when SERR is enabled and the PCI1515 controller signaled a system error to the host. Write a 1 to clear this bit.
13 ‡	MABORT	RW	Received master abort. This bit is set when a cycle initiated by the PCI1515 controller on the PCI bus has been terminated by a master abort. Write a 1 to clear this bit.
12‡	TABT_REC	RW	Received target abort. This bit is set when a cycle initiated by the PCI1515 controller on the PCI bus was terminated by a target abort. Write a 1 to clear this bit.
11 ‡	TABT_SIG	RW	Signaled target abort. This bit is set by the PCI1515 controller when it terminates a transaction on the PCI bus with a target abort. Write a 1 to clear this bit.
10–9	PCI_SPEED	R	DEVSEL timing. These bits encode the timing of DEVSEL and are hardwired to 01b indicating that the PCI1515 controller asserts this signal at a medium speed on nonconfiguration cycle accesses.
8‡	DATAPAR	RW	Data parity error detected. Write a 1 to clear this bit. 0 = The conditions for setting this bit have not been met. 1 = A data parity error occurred and the following conditions were met: a. PERR was asserted by any PCI device including the PCI1515 controller. b. The PCI1515 controller was the bus master during the data parity error. c. The parity error response bit is set in the command register.
7	FBB_CAP	R	Fast back-to-back capable. The PCI1515 controller cannot accept fast back-to-back transactions; thus, this bit is hardwired to 0.
6	UDF	R	UDF supported. The PCI1515 controller does not support user-definable features; therefore, this bit is hardwired to 0.
5	66MHZ	R	66-MHz capable. The PCI1515 controller operates at a maximum PCLK frequency of 33 MHz; therefore, this bit is hardwired to 0.
4	CAPLIST	R	Capabilities list. This bit returns 1 when read. This bit indicates that capabilities in addition to standard PCI capabilities are implemented. The linked list of PCI power-management capabilities is implemented in this function.
3	INT_STATUS	RU	Interrupt status. This bit reflects the interrupt status of the function. Only when bit 10 (INT_DISABLE) in the command register (PCI offset 04h, see Section 4.4) is a 0 and this bit is a 1, is the function's INTx signal asserted. Setting the INT_DISABLE bit to a 1 has no effect on the state of this bit.
2-0	RSVD	R	Reserved. These bits return 0s when read.

[‡] One or more bits in this register are cleared only by the assertion of GRST.

4.6 Revision ID Register

The revision ID register indicates the silicon revision of the PCI1515 controller.

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

Register: Revision ID
Offset: 08h (Function 0)
Type: Read-only

Default: 00h

4.7 Class Code Register

The class code register recognizes PCI1515 function 0 as a bridge device (06h) and a CardBus bridge device (07h), with a 00h programming interface.

Bit	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		PCI class code																						
	Base class Subclass												F	Progra	ammir	ng inte	erface)						
Туре	R	R	R	R	R	R	R	R	R	RRRRRRR				R	R	R	R	R	R	R	R			
Default	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

Register: PCI class code
Offset: 09h (Function 0)
Type: Read-only
Default: 06 0700h

4.8 Cache Line Size Register

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

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

Register: Cache line size
Offset: OCh (Function 0)

Type: Read/Write

Default: 00h

4.9 Latency Timer Register

The latency timer register specifies the latency timer for the PCI1515 controller, in units of PCI clock cycles. When the PCI1515 controller is a PCI bus initiator and asserts FRAME, the latency timer begins counting from zero. If the latency timer expires before the PCI1515 transaction has terminated, then the PCI1515 controller terminates the transaction when its GNT is deasserted.

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

Register: Latency timer

Offset: 0Dh

Type: Read/Write

Default: 00h

4.10 Header Type Register

The header type register returns 82h when read, indicating that the PCI1515 function 0 configuration spaces adhere to the CardBus bridge PCI header. The CardBus bridge PCI header ranges from PCI registers 00h–7Fh, and 80h–FFh is user-definable extension registers.

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

Register: **Header type**Offset: 0Eh (Function 0)

Type: Read-only

Default: 82h

4.11 BIST Register

Because the PCI1515 controller 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								
Туре	R	R	R	R	R	R				
Default	0	0		0	0	0	0	0		

Register: BIST

Offset: 0Fh (Function 0)

Type: Read-only Default: 00h

4.12 CardBus Socket Registers/ExCA Base Address Register

This register is programmed with a base address referencing the CardBus socket registers and the memory-mapped ExCA register set. Bits 31–12 are read/write, and allow the base address to be located anywhere in the 32-bit PCI memory address space on a 4-Kbyte boundary. Bits 11–0 are read-only, returning 0s when read. When software writes all 1s to this register, the value read back is FFFF F000h, indicating that at least 4K bytes of memory address space are required. The CardBus registers start at offset 000h, and the memory-mapped ExCA registers begin at offset 800h.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name						CardB	us socke	et registe	ers/ExCA	h base a	ddress					
Туре	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						CardB	us socke	et registe	ers/ExCA	base a	ddress					
Туре	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: CardBus socket registers/ExCA base address

Offset: 10h

Type: Read-only, Read/Write

Default: 0000 0000h

4.13 Capability Pointer Register

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

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

Register: Capability pointer

Offset: 14h
Type: Read-only
Default: A0h

4.14 Secondary Status Register

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							S	econdary	/ status							
Туре	RC	RC	RC	RC	RC	R	R	RC	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Register: Secondary status

Offset: 16h

Type: Read-only, Read/Clear

Default: 0200h

Table 4-5. Secondary Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
15 ‡	CBPARITY	RC	Detected parity error. This bit is set when a CardBus parity error is detected, either an address or data parity error. Write a 1 to clear this bit.
14 ‡	CBSERR	RC	Signaled system error. This bit is set when CSERR is signaled by a CardBus card. The PCI1515 controller does not assert the CSERR signal. Write a 1 to clear this bit.
13 ‡	CBMABORT	RC	Received master abort. This bit is set when a cycle initiated by the PCI1515 controller on the CardBus bus is terminated by a master abort. Write a 1 to clear this bit.
12 ‡	REC_CBTA	RC	Received target abort. This bit is set when a cycle initiated by the PCI1515 controller on the CardBus bus is terminated by a target abort. Write a 1 to clear this bit.
11 ‡	SIG_CBTA	RC	Signaled target abort. This bit is set by the PCI1515 controller when it terminates a transaction on the CardBus bus with a target abort. Write a 1 to clear this bit.
10–9	CB_SPEED	R	CDEVSEL timing. These bits encode the timing of CDEVSEL and are hardwired to 01b indicating that the PCI1515 controller asserts this signal at a medium speed.
8‡	CB_DPAR	RC	CardBus data parity error detected. Write a 1 to clear this bit. 0 = The conditions for setting this bit have not been met. 1 = A data parity error occurred and the following conditions were met: a. CPERR was asserted on the CardBus interface. b. The PCI1515 controller was the bus master during the data parity error. c. The parity error response enable bit (bit 0) is set in the bridge control register (PCI offset 3Eh, see Section 4.25).
7	CBFBB_CAP	R	Fast back-to-back capable. The PCI1515 controller cannot accept fast back-to-back transactions; therefore, this bit is hardwired to 0.
6	CB_UDF	R	User-definable feature support. The PCI1515 controller does not support user-definable features; therefore, this bit is hardwired to 0.
5	CB66MHZ	R	66-MHz capable. The PCI1515 CardBus interface operates at a maximum CCLK frequency of 33 MHz; therefore, this bit is hardwired to 0.
4-0	RSVD	R	These bits return 0s when read.

[‡] One or more bits in this register are cleared only by the assertion of GRST.

4.15 PCI Bus Number Register

The PCI bus number register is programmed by the host system to indicate the bus number of the PCI bus to which the PCI1515 controller is connected. The PCI1515 controller 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												
Туре	RW	RW	RW	RW	RW	RW	RW	RW					
Default	0	0	0	0	0	0	0	0					

Register: **PCI bus number**Offset: 18h (Function 0)
Type: Read/Write

Default: 00h

4.16 CardBus Bus Number Register

The CardBus bus number register is programmed by the host system to indicate the bus number of the CardBus bus to which the PCI1515 controller is connected. The PCI1515 controller 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.

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

Register: CardBus bus number

Offset: 19h Type: Read/Write

Default: 00h

4.17 Subordinate Bus Number Register

The subordinate bus number register is programmed by the host system to indicate the highest numbered bus below the CardBus bus. The PCI1515 controller 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.

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

Register: Subordinate bus number

Offset: 1Ah

Type: Read/Write

Default: 00h

4.18 CardBus Latency Timer Register

The CardBus latency timer register is programmed by the host system to specify the latency timer for the PCI1515 CardBus interface, in units of CCLK cycles. When the PCI1515 controller is a CardBus initiator and asserts CFRAME, the CardBus latency timer begins counting. If the latency timer expires before the PCI1515 transaction has terminated, then the PCI1515 controller terminates the transaction at the end of the next data phase. A recommended minimum value for this register of 20h allows most transactions to be completed.

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

Register: CardBus latency timer

Offset: 1Bh (Function 0) Type: Read/Write

Default: 00h

4.19 CardBus Memory Base Registers 0, 1

These registers indicate the lower address of a PCI memory address range. They are used by the PCI1515 controller to determine when to forward a memory transaction to the CardBus bus, and likewise, when to forward a CardBus cycle to PCI. Bits 31–12 of these registers are read/write and allow the memory base to be located anywhere in the 32-bit PCI memory space on 4-Kbyte boundaries. Bits 11–0 are read-only and always return 0s. Writes to these bits have no effect. Bits 8 and 9 of the bridge control register (PCI offset 3Eh, see Section 4.25) specify whether memory windows 0 and 1 are prefetchable or nonprefetchable. The memory base register or the memory limit register must be nonzero in order for the PCI1515 controller to claim any memory transactions through CardBus memory windows (i.e., these windows by default are not enabled to pass the first 4 Kbytes of memory to CardBus).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							Memo	ory base	register	s 0, 1						
Туре	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW						
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							Memo	ory base	register	s 0, 1						
Туре	RW	RW	RW	RW	R	R	R	R	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: Memory base registers 0, 1

Offset: 1Ch. 24h

Type: Read-only, Read/Write

Default: 0000 0000h

4.20 CardBus Memory Limit Registers 0, 1

These registers indicate the upper address of a PCI memory address range. They are used by the PCI1515 controller to determine when to forward a memory transaction to the CardBus bus, and likewise, when to forward a CardBus cycle to PCI. Bits 31–12 of these registers are read/write and allow the memory base to be located anywhere in the 32-bit PCI memory space on 4-Kbyte boundaries. Bits 11–0 are read-only and always return 0s. Writes to these bits have no effect. Bits 8 and 9 of the bridge control register (PCI offset 3Eh, see Section 4.25) specify whether memory windows 0 and 1 are prefetchable or nonprefetchable. The memory base register or the memory limit register must be nonzero in order for the PCI1515 controller to claim any memory transactions through CardBus memory windows (i.e., these windows by default are not enabled to pass the first 4 Kbytes of memory to CardBus).

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

Register: Memory limit registers 0, 1

Offset: 20h, 28h

Type: Read-only, Read/Write

Default: 0000 0000h

4.21 CardBus I/O Base Registers 0, 1

These registers indicate the lower address of a PCI I/O address range. They are used by the PCI1515 controller to determine when to forward an I/O transaction to the CardBus bus, and likewise, when to forward a CardBus cycle to the PCI bus. The lower 16 bits of this register locate the bottom of the I/O window within a 64-Kbyte page. The upper 16 bits (31–16) are all 0s, which locates this 64-Kbyte page in the first page of the 32-bit PCI I/O address space. Bits 31–2 are read/write and always return 0s forcing I/O windows to be aligned on a natural doubleword boundary in the first 64-Kbyte page of PCI I/O address space. Bits 1–0 are read-only, returning 00 or 01 when read, depending on the value of bit 11 (IO_BASE_SEL) in the general control register (PCI offset 86h, see Section 4.30). These I/O windows are enabled when either the I/O base register or the I/O limit register is nonzero. The I/O windows by default are not enabled to pass the first doubleword of I/O to CardBus.

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

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							I/O	base re	gisters C), 1						
Туре	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW						
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							I/O	base re	gisters C), 1						
Туре	RW	RW	RW	RW	RW	RW	RW	RW	R	R						
Default	0	0	Λ	٥	Λ	0	0	0	0	0	0	0	0	0	0	Х

Register: I/O base registers 0, 1

Offset: 2Ch, 34h

Type: Read-only, Read/Write

Default: 0000 000Xh

4.22 CardBus I/O Limit Registers 0, 1

These registers indicate the upper address of a PCI I/O address range. They are used by the PCI1515 controller to determine when to forward an I/O transaction to the CardBus bus, and likewise, when to forward a CardBus cycle to PCI. The lower 16 bits of this register locate the top of the I/O window within a 64-Kbyte page, and the upper 16 bits are a page register which locates this 64-Kbyte page in 32-bit PCI I/O address space. Bits 15–2 are read/write and allow the I/O limit address to be located anywhere in the 64-Kbyte page (indicated by bits 31–16 of the appropriate I/O base register) on doubleword boundaries.

Bits 31–16 are read-only and always return 0s when read. The page is set in the I/O base register. Bits 15–2 are read/write and bits 1–0 are read-only, returning 00 or 01 when read, depending on the value of bit 12 (IO_LIMIT_SEL) in the general control register (PCI offset 86h, see Section 4.30). Writes to read-only bits have no effect.

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

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

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							I/C	limit re	gisters 0	, 1						
Туре	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/C	limit re	gisters 0	, 1						
Туре	RW	RW	RW	RW	RW	RW	RW	RW	R	R						
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Х

Register: I/O limit registers 0, 1

Offset: 30h, 38h

Type: Read-only, Read/Write

Default: 0000 000Xh

4.23 Interrupt Line Register

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

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

Register: Interrupt line

Offset: 3Ch

Type: Read/Write

Default: FFh

4.24 Interrupt Pin Register

The value read from this register is function dependent. The value for function 0 is fixed to 01h ($\overline{\text{INTA}}$).

PCI function 0

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

Interrupt pin 3Dh

Register: Offset:

Type: Default: Read-only 01h (function 0)

4.25 Bridge Control Register

The bridge control register provides control over various PCI1515 bridging functions. 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														
Туре	R	R	R	R	R	RW	RW	RW	RW	RW	RW	R	RW	RW	RW	RW
Default	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0

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

Default: 0340h

Table 4-6. Bridge Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–11	RSVD	R	These bits return 0s when read.
10	POSTEN	RW	Write posting enable. Enables write posting to and from the CardBus socket. Write posting enables the posting of write data on burst cycles. Operating with write posting disabled impairs performance on burst cycles. Note that burst write data can be posted, but various write transactions may not.
9	PREFETCH1	RW	Memory window 1 type. This bit specifies whether or not memory window 1 is prefetchable. This bit is encoded as: 0 = Memory window 1 is nonprefetchable. 1 = Memory window 1 is prefetchable (default).
8	PREFETCH0	RW	Memory window 0 type. This bit specifies whether or not memory window 0 is prefetchable. This bit is encoded as: 0 = Memory window 0 is nonprefetchable. 1 = Memory window 0 is prefetchable (default).
7	INTR	RW	PCI interrupt – IREQ routing enable. This bit is used to select whether PC Card functional interrupts are routed to PCI interrupts or to the IRQ specified in the ExCA registers. 0 = Functional interrupts are routed to PCI interrupts (default). 1 = Functional interrupts are routed by ExCA registers.
6†	CRST	RW	CardBus reset. When this bit is set, the CRST signal is asserted on the CardBus interface. The CRST signal can also be asserted by passing a PRST assertion to CardBus. 0 = CRST is deasserted. 1 = CRST is asserted (default). This bit is not cleared by the assertion of PRST. It is only cleared by the assertion of GRST.
5	MABTMODE	RW	Master abort mode. This bit controls how the PCI1515 controller responds to a master abort when the PCI1515 controller is an initiator on the CardBus interface. 0 = Master aborts not reported (default). 1 = Signal target abort on PCI and signal SERR, if enabled.
4	RSVD	R	This bit returns 0 when read.
3	VGAEN	RW	VGA enable. This bit affects how the PCI1515 controller responds to VGA addresses. When this bit is set, accesses to VGA addresses are forwarded.
2	ISAEN	RW	ISA mode enable. This bit affects how the PCI1515 controller passes I/O cycles within the 64-Kbyte ISA range. When this bit is set, the PCI1515 controller does not forward the last 768 bytes of each 1K I/O range to CardBus.
1	CSERREN	RW	CSERR enable. This bit controls the response of the PCI1515 controller to CSERR signals on the CardBus bus. 0 = CSERR is not forwarded to PCI SERR (default) 1 = CSERR is forwarded to PCI SERR.
0	CPERREN	RW	CardBus parity error response enable. This bit controls the response of the PCI1515 to CardBus parity errors. 0 = CardBus parity errors are ignored (default). 1 = CardBus parity errors are reported using CPERR.

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

4.26 Subsystem Vendor ID Register

The subsystem vendor ID register, used for system and option card identification purposes, may be required for certain operating systems. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register (PCI offset 80h, See Section 4.29). When bit 5 is 0, this register is read-write; when bit 5 is 1, this register is read-only. The default mode is read-only. All bits in this register are reset by GRST only.

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

Register: Subsystem vendor ID

Offset: 40h (Function 0)

Type: Read-only, (Read/Write when bit 5 in the system control register is 0)

Default: 0000h

4.27 Subsystem ID Register

The subsystem ID register, used for system and option card identification purposes, may be required for certain operating systems. This register is read-only or read/write, depending on the setting of bit 5 (SUBSYSRW) in the system control register (PCI offset 80h, see Section 4.29). When bit 5 is 0, this register is read/write; when bit 5 is 1, this register is read-only. The default mode is read-only. All bits in this register are reset by GRST only.

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

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

Register: Subsystem ID
Offset: 42h (Function 0)

Type: Read-only, (Read/Write when bit 5 in the system control register is 0)

Default: 0000h

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

The PCI1515 controller 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. See the ExCA register set description in Section 5 for register offsets. All bits in this register are reset by GRST only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name						PC Ca	rd 16-bit	I/F lega	cy-mode	base-a	ddress					
Туре	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name						PC Ca	rd 16-bit	I/F lega	cy-mode	base-a	ddress					
Туре	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

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

Offset: 44h (Function 0)
Type: Read-only, Read/Write

Default: 0000 0001h

4.29 System Control Register

System-level initializations are performed through programming this doubleword register. 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							
Туре	RW	R	RW	RW	RW	R	R	R	R							
Default	0	0	0	0	1	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							
Туре	RW	RW	R	R	R	R	R	R	R	RW	RW	RW	RW	R	RW	RW
Default	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0

Register: System control
Offset: 80h (Function 0)
Type: Read-only, Read/Write

Default: 0844 9060h

Table 4-7. System Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
31-30‡	SER_STEP	RW	Serial input stepping. In serial PCI interrupt mode, these bits are used to configure the serial stream PCI interrupt frames, and can be used to accomplish an even distribution of interrupts signaled on the four PCI interrupt slots. 00 = INTA/INTB/INTC/INTD signal in INTA/INTB/INTC/INTD slots (default) 01 = INTA/INTB/INTC/INTD signal in INTB/INTC/INTD/INTA slots 10 = INTA/INTB/INTC/INTD signal in INTC/INTD/INTA/INTB slots 11 = INTA/INTB/INTC/INTD signal in INTC/INTA/INTB/INTC slots
29–28	RSVD	RW	Reserved. These bits are reserved, reads or writes have no effect on functionality.
27 ‡	PSCCLK	RW	P2C power switch clock. The PCI1515 CLOCK signal clocks the serial interface power switch and the internal state machine. The default state for this bit is 0, requiring an external clock source provided to the CLOCK terminal. Bit 27 can be set to 1, allowing the internal oscillator to provide the clock signal. 0 = CLOCK is provided externally, input to the PCI1515 controller. 1 = CLOCK is generated by the internal oscillator and driven by the PCI1515 controller. (default)
26 ‡	SMIROUTE	RW	SMI interrupt routing. This bit selects whether IRQ2 or CSC is signaled when a write occurs to power a PC Card socket. 0 = PC Card power change interrupts are routed to IRQ2 (default). 1 = A CSC interrupt is generated on PC Card power changes.
25 ‡	SMISTATUS	RW	SMI interrupt status. This bit is set when a write occurs to set the socket power, and the SMIENB bit is set. Writing a 1 to this bit clears the status. 0 = SMI interrupt is signaled. 1 = SMI interrupt is not signaled.
24 ‡	SMIENB	RW	SMI interrupt mode enable. When this bit is set, the SMI interrupt signaling generates an interrupt when a write to the socket power control occurs. This bit defaults to 0 (disabled). 0 = SMI interrupt mode is disabled (default). 1 = SMI interrupt mode is enabled.
23	RSVD	R	Reserved
22‡	CBRSVD	RW	CardBus reserved terminals signaling. When this bit is set, the RSVD CardBus terminals are driven low when a CardBus card has been inserted. When this bit is low, these signals are placed in a high-impedance state. 0 = Place the CardBus RSVD terminals in a high-impedance state. 1 = Drive the CardBus RSVD terminals low (default).

 $[\]ddagger$ One or more bits in this register are cleared only by the assertion of $\overline{\text{GRST}}$.

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

BIT	SIGNAL	TYPE	FUNCTION
21 ‡	VCCPROT	RW	V _{CC} protection enable. 0 = V _{CC} protection is enabled for 16-bit cards (default). 1 = V _{CC} protection is disabled for 16-bit cards.
20–16 ‡	RSVD	RW	These bits are reserved. Do not change the value of these bits.
15‡	MRBURSTDN	RW	Memory read burst enable downstream. When this bit is set, the PCI1515 controller allows memory read transactions to burst downstream. 0 = MRBURSTDN downstream is disabled. 1 = MRBURSTDN downstream is enabled (default).
14‡	MRBURSTUP	RW	Memory read burst enable upstream. When this bit is set, the PCI1515 controller allows memory read transactions to burst upstream. 0 = MRBURSTUP upstream is disabled (default). 1 = MRBURSTUP upstream is enabled.
13‡	SOCACTIVE	R	Socket activity status. When set, this bit indicates access has been performed to or from a PC Card. Reading this bit causes it to be cleared. 0 = No socket activity (default) 1 = Socket activity
12	RSVD	R	Reserved. This bit returns 1 when read.
11 ‡	PWRSTREAM	R	Power-stream-in-progress status bit. When set, this bit indicates that a power stream to the power switch is in progress and a powering change has been requested. When this bit is cleared, it indicates that the power stream is complete. 0 = Power stream is complete, delay has expired (default). 1 = Power stream is in progress.
10 †	DELAYUP	R	Power-up delay-in-progress status bit. When set, this bit indicates that a power-up stream has been sent to the power switch, and proper power may not yet be stable. This bit is cleared when the power-up delay has expired. 0 = Power-up delay has expired (default). 1 = Power-up stream sent to switch. Power might not be stable.
9†	DELAYDOWN	R	Power-down delay-in-progress status bit. When set, this bit indicates that a power-down stream has been sent to the power switch, and proper power may not yet be stable. This bit is cleared when the power-down delay has expired. 0 = Power-down delay has expired (default). 1 = Power-down stream sent to switch. Power might not be stable.
8†	INTERROGATE	R	Interrogation in progress. When set, this bit indicates an interrogation is in progress, and clears when the interrogation completes. 0 = Interrogation not in progress (default) 1 = Interrogation in progress
7	RSVD	R	Reserved. This bit returns 0 when read.
6‡	PWRSAVINGS	RW	Power savings mode enable. When this bit is set, the PCI1515 controller consumes less power with no performance loss. 0 = Power savings mode disabled 1 = Power savings mode enabled (default)
5‡	SUBSYSRW	RW	Subsystem ID and subsystem vendor ID, ExCA ID and revision register read/write enable. 0 = Registers are read/write. 1 = Registers are read-only (default).
4‡	CB_DPAR	RW	CardBus data parity SERR signaling enable. 0 = CardBus data parity not signaled on PCI SERR signal (default) 1 = CardBus data parity signaled on PCI SERR signal
3 ‡	RSVD	R	Reserved. This bit returns 0 when read.

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

‡ One or more bits in this register are cleared only by the assertion of GRST.

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

BIT	SIGNAL	TYPE	FUNCTION
2‡	EXCAPOWER	R	ExCA power control bit. 0 = Enables 3.3 V (default) 1 = Enables 5 V
1 ‡	KEEPCLK	RW	Keep clock. When this bit is set, the PCI1515 controller follows the CLKRUN protocol to maintain the system PCLK and the CCLK (CardBus clock). 0 = Allow system PCLK and CCLK clocks to stop (default) 1 = Never allow system PCLK or CCLK clock to stop
			Note that the functionality of this bit has changed relative to that of the PCI12XX family of TI CardBus controllers. In these CardBus controllers, setting this bit only maintains the PCI clock, not the CCLK. In the PCI1515 controller, setting this bit maintains both the PCI clock and the CCLK.
0‡	RIMUX	RW	PME/RI_OUT select bit. When this bit is 1, the PME signal is routed to the PME/RI_OUT terminal (R03). When this bit is 0 and bit 7 (RIENB) of the card control register is 1, the RI_OUT signal is routed to the PME/RI_OUT terminal. If this bit is 0 and bit 7 (RIENB) of the card control register is 0, then the output is placed in a high-impedance state. This terminal is encoded as: 0 = RI_OUT signal is routed to the PME/RI_OUT terminal if bit 7 of the card control register is 1. (default) 1 = PME signal is routed to the PME/RI_OUT terminal of the PCI1515 controller.
			NOTE: If this bit (bit 0) is 0 and bit 7 of the card control register (PCI offset 91h, see Section 4.37) is 0, then the output on the $\overline{\text{PME}/\text{RI}_{-}\text{OUT}}$ terminal is placed in a high-impedance state.

[‡]One or more bits in this register are cleared only by the assertion of GRST.

4.30 General Control Register

The general control register provides top level PCI arbitration control. It also provides control over miscellaneous new functionality. See Table 4–8 for a complete description of the register contents.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		General control														
Туре	RW	RWU	RW													
Default	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1

Register: General control

Offset: 86h

Type: Read/Write, Read-only

Table 4-8. General Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
15–13	RSVD	RW	Reserved, these bits have no effect on device operation.
12 ‡	IO_LIMIT_SEL	RW	When this bit is set, bit 0 in the I/O limit registers (PCI offsets 30h and 38h) is set. 0 = Bit 0 in the I/O limit registers is 0 (default) 1 = Bit 0 in the I/O limit registers is 1
11 ‡	IO_BASE_SEL	RW	When this bit is set, bit 0 in the I/O base registers (PCI offsets 2Ch and 34h) is set. 0 = Bit 0 in the I/O base registers is 0 (default) 1 = Bit 0 in the I/O base registers is 1
10 ‡	12V_SW_SEL	RW	Power switch select. This bit selects which power switch is implemented in the system. 0 = A 1.8-V capable power switch (TPS2228) is used (default) 1 = A 12-V capable power switch (TPS2226) is used
9–0	RSVD	RW	Reserved, these bits have no effect on device operation.

[‡] One or more bits in this register are cleared only by the assertion of GRST.

4.31 General-Purpose Event Status Register

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

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

Register: General-purpose event status

Offset: 88h

Type: Read/Clear/Update, Read-only

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

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

[‡]One or more bits in this register are cleared only by the assertion of GRST.

4.32 General-Purpose Event Enable Register

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

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

Register: General-purpose event enable

Offset: 89h

Type: Read-only, Read/Write

Default: 00h

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

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

[‡]One or more bits in this register are cleared only by the assertion of GRST.

4.33 General-Purpose Input Register

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

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

Register: General-purpose input

Offset: 8Ah

Type: Read/Update, Read-only

Default: XXh

Table 4-11. General-Purpose Input Register Description

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

4.34 General-Purpose Output Register

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

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

Register: General-purpose output

Offset: 8Bh

Type: Read-only, Read/Write

Table 4–12. General-Purpose Output Register Description

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

[‡]One or more bits in this register are cleared only by the assertion of GRST.

4.35 Multifunction Routing Status Register

The multifunction routing status register is used to configure the MFUNC6–MFUNC0 terminals. These terminals may be configured for various functions. This register is intended to be programmed once at power-on initialization. The default value for this register can also be loaded through a serial EEPROM. See Table 4–13 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							Multit	unction	routing s	status						
Туре	R	R RW RW RW R RW RW RW R RW RW RW RW RW 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		_	_	_	_	_	Multif	unction	routing s	status			_	_		
Туре	R	RW	RW	RW	R	RW	RW	RW	R	RW	RW	RW	R	RW	RW	RW
Default	0	0	0	4	Ο	0	0	0	0	0	_	0	0	0	0	0

Register: Multifunction routing status

Offset: 8Ch

Type: Read/Write, Read-only

Default: 0000 1000h

Table 4-13. Multifunction Routing Status Register Description

BIT	SIGNAL	TYPE			FUNCTION	
31-28 ‡	RSVD	R	Bits 31-28 return 0s when	read.		
27-24 ‡	MFUNC6	RW	as follows: 0000 = RSVD 0001 = CLKRUN 0010 = IRQ2	ofiguration. These b 0100 = IRQ4 0101 = IRQ5 0110 = IRQ6 0111 = IRQ7	its control the internal sig 1000 = IRQ8 1001 = IRQ9 1010 = IRQ10 1011 = IRQ11	gnal mapped to the MFUNC6 terminal 1100 = IRQ12 1101 = IRQ13 1110 = IRQ14 1111 = IRQ15
23-20‡	MFUNC5	RW	as follows: 0000 = GPI4 0001 = GPO4 0010 = RSVD	0100 = RSVD 0101 = IRQ5 0101 = IRQ5 0110 = RSVD 0111 = RSVD	its control the internal sig 1000 = CAUDPWM 1001 = IRQ9 1010 = RSVD 1011 = RSVD	gnal mapped to the MFUNC5 terminal 1100 = LEDA1 1101 = LED_SKT 1110 = GPE 1111 = IRQ15
19–16‡	MFUNC4	RW	as follows: 0000 = GPI3 0001 = GPO3 0010 = LOCK PCI	ofiguration. These b 0100 = IRQ4 0101 = RSVD 0110 = RSVD 0111 = RSVD	its control the internal sig 1000 = CAUDPWM 1001 = IRQ9 1010 = RSVD 1011 = RSVD	gnal mapped to the MFUNC4 terminal 1100 = RI_OUT 1101 = LED_SKT 1110 = GPE 1111 = IRQ15
15-12‡	MFUNC3	RW	as follows: 0000 = RSVD 0001 = IRQSER 0010 = IRQ2	0100 = IRQ4 0101 = IRQ5 0101 = IRQ6 0111 = IRQ6 0111 = IRQ7	its control the internal sig 1000 = IRQ8 1001 = IRQ9 1010 = IRQ10 1011 = IRQ11	gnal mapped to the MFUNC3 terminal 1100 = IRQ12 1101 = IRQ13 1110 = IRQ14 1111 = IRQ15
11-8 ‡	MFUNC2	RW	as follows: 0000 = GPI2 0001 = GPO2 0010 = RSVD	0100 = IRQ4 0101 = IRQ5 0101 = RSVD 0111 = RSVD	its control the internal sig 1000 = CAUDPWM 1001 = RSVD 1010 = IRQ10 1011 = RSVD	gnal mapped to the MFUNC2 terminal 1100 = RI_OUT 1101 = TEST_MUX 1110 = GPE 1111 = IRQ7

[‡] One or more bits in this register are cleared only by the assertion of GRST.

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

BIT	SIGNAL	TYPE	FUNCTION						
7–4‡	MFUNC1	RW	Multifunction terminal 1 configuration. These bits control the internal signal mapped to the MFUNC1 terminal as follows: $0000 = \text{GPI1} \qquad 0100 = \text{RSVD} \qquad 1000 = \text{CAUDPWM} \qquad 1100 = \text{LEDA1} \\ 0001 = \text{GPO1} \qquad 0101 = \text{IRQ5} \qquad 1001 = \text{IRQ9} \qquad 1101 = \frac{\text{RSVD}}{\text{PRSVD}} \\ 0010 = \text{RSVD} \qquad 0110 = \text{RSVD} \qquad 1010 = \text{IRQ10} \qquad 1110 = \frac{\text{GPE}}{\text{GPE}} \\ 0011 = \text{IRQ3} \qquad 0111 = \text{RSVD} \qquad 1011 = \text{IRQ11} \qquad 1111 = \text{IRQ15}$						
3-0‡	MFUNC0	RW	Multifunction terminal 0 configuration. These bits control the internal signal mapped to the MFUNC0 terminal as follows: $0000 = \text{GPI0} \qquad 0100 = \text{IRQ4} \qquad 1000 = \text{CAUDPWM} \qquad 1100 = \text{LEDA1} \\ 0001 = \frac{\text{GPO}}{\text{O}}0 \qquad 0101 = \text{IRQ5} \qquad 1001 = \text{IRQ9} \qquad 1101 = \frac{\text{RSVD}}{\text{D}}0 \\ 0010 = \overline{\text{INTA}} \qquad 0110 = \text{RSVD} \qquad 1010 = \overline{\text{IRQ10}} \qquad 1110 = \overline{\text{GPE}} \\ 0011 = \overline{\text{IRQ3}} \qquad 0111 = \overline{\text{RSVD}} \qquad 1011 = \overline{\text{IRQ11}} \qquad 1111 = \overline{\text{IRQ15}}$						

[‡]One or more bits in this register are cleared only by the assertion of GRST.

4.36 Retry Status Register

The contents of the retry status register enable the retry time-out counters and display the retry expiration status. The flags are set when the PCI1515 controller, as a master, receives a retry and does not retry the request within 2¹⁵ clock cycles. The flags are cleared by writing a 1 to the bit. See Table 4–14 for a complete description of the register contents.

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

Register: Retry status
Offset: 90h (Function 0)

Type: Read-only, Read/Write, Read/Clear

Table 4-14. Retry Status Register Description

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

[‡] One or more bits in this register are cleared only by the assertion of GRST.

4.37 Card Control Register

The card control register is provided for PCI1130 compatibility. The $\overline{\text{RI_OUT}}$ signal is enabled through this register. See Table 4–15 for a complete description of the register contents.

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

Register: Card control

Offset: 91h

Type: Read-only, Read/Write

Default: 00h

Table 4-15. Card Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7 ‡	RIENB	RW	Ring indicate enable. When this bit is 1, the RI_OUT output is enabled. This bit defaults to 0.
6–3	RSVD	RW	These bits are reserved. Do not change the value of these bits.
2.+			CardBus audio-to-MFUNC. When this bit is set, the CAUDIO CardBus signal must be routed through an MFUNC terminal.
2 ‡ AUD2MUX	RW	0 = CAUDIO set to CAUDPWM on MFUNC terminal (default) 1 = CAUDIO is not routed.	
1 ‡	1 ± SPKROUTEN R		When bit 1 is set, the SPKR terminal from the PC Card is enabled and is routed to tthe SPKROUT terminal. The SPKROUT terminal drives data only when the SPKROUTEN bit is set. This bit is encoded as:
'+	SIRROUTEN	EN RW	0 = <u>SPKR</u> to SPKROUT not enabled (default) 1 = SPKR to SPKROUT enabled
0.+	IFC	DW	Interrupt flag. This bit is the interrupt flag for 16-bit I/O PC Cards and for CardBus cards. This bit is set when a functional interrupt is signaled from a PC Card interface. Write back a 1 to clear this bit.
0 ‡ IFG		RW	0 = No PC Card functional interrupt detected (default) 1 = PC Card functional interrupt detected

[‡]One or more bits in this register are cleared only by the assertion of GRST.

4.38 Device Control Register

The device control register is provided for PCI1130 compatibility. The interrupt mode select is programmed through this register. The socket-capable force bits are also programmed through this register. See Table 4–16 for a complete description of the register contents.

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

Register: **Device control**Offset: 92h (Function 0)
Type: Read-only, Read/Write

Default: 66h

Table 4-16. Device Control Register Description

BIT	SIGNAL	TYPE	FUNCTION
7‡	SKTPWR_LOCK	RW	Socket power lock bit. When this bit is set to 1, software cannot power down the PC Card socket while in D3. It may be necessary to lock socket power in order to support wake on LAN or RING if the operating system is programmed to power down a socket when the CardBus controller is placed in the D3 state.
6‡	3VCAPABLE	RW	3-V socket capable force bit. 0 = Not 3-V capable 1 = 3-V capable (default)
5 ‡	IO16R2	RW	Diagnostic bit. This bit defaults to 1.
4	RSVD	R	Reserved. This bit returns 0 when read. A write has no effect.
3 ‡	TEST	RW	TI test bit. Write only 0 to this bit.
2–1 ‡	INTMODE	RW	Interrupt mode. These bits select the interrupt signaling mode. The interrupt mode bits are encoded: 00 = Parallel PCI interrupts only 01 = Reserved 10 = IRQ serialized interrupts and parallel PCI interrupts INTA, INTB, INTC, and INTD 11 = IRQ and PCI serialized interrupts (default)
0 ‡	RSVD	RW	Reserved. Bit 0 is reserved for test purposes. Only a 0 must be written to this bit.

[‡]One or more bits in this register are cleared only by the assertion of GRST.

4.39 Diagnostic Register

The diagnostic register is provided for internal TI test purposes. It is a read/write register, but only 0s must be written to it. See Table 4–17 for a complete description of the register contents.

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

Register: **Diagnostic**Offset: 93h (Function 0)
Type: Read/Write

Default: 60h

Table 4-17. Diagnostic Register Description

BIT	SIGNAL	TYPE	FUNCTION
7‡	TRUE_VAL	RW	This bit defaults to 0. This bit is encoded as: 0 = Reads true values in PCI vendor ID and PCI device ID registers (default) 1 = Returns all 1s to reads from the PCI vendor ID and PCI device ID registers
6‡	RSVD	R	Reserved. This bit is read-only and returns 1 when read.
5‡	CSC	RW	CSC interrupt routing control 0 = CSC interrupts routed to PCI if ExCA 803 bit 4 = 1 1 = CSC interrupts routed to PCI if ExCA 805 bits 7–4 = 0000b (default). In this case, the setting of ExCA 803 bit 4 is a don't care.
4 ‡	DIAG4	RW	Diagnostic RETRY_DIS. Delayed transaction disable.
3 ‡	DIAG3	RW	Diagnostic RETRY_EXT. Extends the latency from 16 to 64.
2 ‡	DIAG2	RW	Diagnostic DISCARD_TIM_SEL_CB. Set = 2 ¹⁰ , reset = 2 ¹⁵ .
1 ‡	DIAG1	RW	Diagnostic DISCARD_TIM_SEL_PCI. Set = 2 ¹⁰ , reset = 2 ¹⁵ .
0 ‡	RSVD	RW	These bits are reserved. Do not change the value of these bits.

[‡] One or more bits in this register are cleared only by the assertion of GRST.

4.40 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				Capab	ility ID			
Туре	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	1

Register: Capability ID

Offset: A0h Type: Read-only Default: 01h

4.41 Next Item Pointer Register

The contents of this register indicate the next item in the linked list of the PCI power management capabilities. Because the PCI1515 functions only include one capabilities item, this register returns 0s when read.

Bit	7	6	5	4	3	2	1	0
Name				Next iten	n pointer			
Туре	R	R	R	R	R	R	R	R
Default	0	0	0	0	0	0	0	0

Register: **Next item pointer**

Offset: A1h
Type: Read-only
Default: 00h

4.42 Power Management Capabilities Register

The power management capabilities register contains information on the capabilities of the PC Card function related to power management. The PCI1515 CardBus bridge function supports the D0, D1, D2, and D3 power states. The default register value is FE12h for operation in accordance with *PCI Bus Power Management Interface Specification* revision 1.1. See Table 4–18 for a complete description of the register contents.

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

Register: Power management capabilities

Offset: A2h (Function 0)
Type: Read-only, Read/Write

Default: FE12h

Table 4–18. Power Management Capabilities Register Description

BIT	SIGNAL	TYPE	FUNCTION
			This 5-bit field indicates the power states from which the PCI1515 controller functions can assert PME. A 0 for any bit indicates that the function cannot assert the PME signal while in that power state. These 5 bits return 11111b when read. Each of these bits is described below:
15‡	PME support	RW	Bit 15 – defaults to a 1 indicating the \overline{PME} signal can be asserted from the D3 _{COld} state. This bit is read/write because wake-up support from D3 _{COld} is contingent on the system providing an auxiliary power source to the V _{CC} terminals. If the system designer chooses not to provide an auxiliary power source to the V _{CC} terminals for D3 _{COld} wake-up support, then BIOS must write a 0 to this bit.
14–11			Bit 14 – contains the value 1 to indicate that the PME signal can be asserted from the D3hot state. Bit 13 – contains the value 1 to indicate that the PME signal can be asserted from the D2 state. Bit 12 – contains the value 1 to indicate that the PME signal can be asserted from the D1 state. Bit 11 – contains the value 1 to indicate that the PME signal can be asserted from the D0 state.
10	D2_Support	R	This bit returns a 1 when read, indicating that the function supports the D2 device power state.
9	D1_Support	R	This bit returns a 1 when read, indicating that the function supports the D1 device power state.
8–6	RSVD	R	Reserved. These bits return 000b when read.
5	DSI	R	Device-specific initialization. This bit returns 0 when read.
4	AUX PWR	R	Auxiliary power source. This bit is meaningful only if bit 15 (D3 _{Cold} supporting PME) is set. When this bit is set, it indicates that support for PME in D3 _{Cold} requires auxiliary power supplied by the system by way of a proprietary delivery vehicle.
4	AUX_PWR	K	A 0 (zero) in this bit field indicates that the function supplies its own auxiliary power source.
			If the function does not support PME while in the D3 _{cold} state (bit 15=0), then this field must always return 0.
3	PMECLK	R	When this bit is 1, it indicates that the function relies on the presence of the PCI clock for <u>PME</u> operation. When this bit is 0, it indicates that no PCI clock is required for the function to generate <u>PME</u> .
			Functions that do not support PME generation in any state must return 0 for this field.
2-0	Version	R	These 3 bits return 010b when read, indicating that there are 4 bytes of general-purpose power management (PM) registers as described in draft revision 1.1 of the <i>PCI Bus Power Management Interface Specification</i> .

[‡]One or more bits in this register are cleared only by the assertion of GRST.

4.43 Power Management Control/Status Register

The power management control/status register determines and changes the current power state of the PCI1515 CardBus function. The contents of this register are not affected by the internally generated reset caused by the transition from the D3_{hot} to D0 state. See Table 4–19 for a complete description of the register contents.

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

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

Register: Power management control/status

Offset: A4h (Function 0)

Type: Read-only, Read/Write, Read/Write/Clear

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

BIT	SIGNAL	TYPE	FUNCTION
15 †	PMESTAT	RC	PME status. This bit is set when the CardBus function would normally assert the PME signal, independent of the state of the PME_EN bit. This bit is cleared by a writeback of 1, and this also clears the PME signal if PME was asserted by this function. Writing a 0 to this bit has no effect.
14–13	DATASCALE	R	This 2-bit field returns 0s when read. The CardBus function does not return any dynamic data.
12–9	DATASEL	R	Data select. This 4-bit field returns 0s when read. The CardBus function does not return any dynamic data.
8‡	PME_ENABLE	RW	This bit enables the function to assert <u>PME</u> . If this bit is cleared, then assertion of <u>PME</u> is disabled. This bit is not cleared by the assertion of <u>PRST</u> . It is only cleared by the assertion of <u>GRST</u> .
7–2	RSVD	R	Reserved. These bits return 0s when read.
1–0	PWRSTATE	RW	Power state. This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state. This field is encoded as: 00 = D0 01 = D1 10 = D2 11 = D3hot

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

[‡]One or more bits in this register are cleared only by the assertion of GRST.

4.44 Power Management Control/Status Bridge Support Extensions Register

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

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

Register: Power management control/status bridge support extensions

Offset: A6h (Function 0)
Type: Read-only

Default: C0h

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

BIT	SIGNAL	TYPE	FUNCTION
			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).
7	BPCC_EN	R	A 0 indicates that the bus power/clock control policies defined in the <i>PCI Bus Power Management Interface Specification</i> are disabled. When the bus power/clock control enable mechanism is disabled, the power state field (bits 1–0) of the power management control/status register (PCI offset A4h, see Section 4.43) cannot be used by the system software to control the power or the clock of the secondary bus. A 1 indicates that the bus power/clock control mechanism is enabled.
6	B2_B3	R	B2/B3 support for D3 _{hot} . The state of this bit determines the action that is to occur as a direct result of programming the function to D3 _{hot} . This bit is only meaningful if bit 7 (BPCC_EN) is a 1. This bit is encoded as: 0 = When the bridge is programmed to D3 _{hot} , its secondary bus has its power removed (B3). 1 = When the bridge function is programmed to D3 _{hot} , its secondary bus PCI clock is stopped (B2) (default).
5–0	RSVD	R	Reserved. These bits return 0s when read.

4.45 Power-Management Data Register

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

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

Register: Power-management data

Offset: A7h (Function 0)
Type: Read-only

4.46 Serial Bus Data Register

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

On byte reads, the byte address is programmed into the serial bus index register, the serial bus slave address register must be programmed with both the 7-bit slave address and the read/write indicator bit, and bit 5 (REQBUSY) in the serial bus control and status register (see Section 4.49) must be polled until clear. Then the contents of this register are valid read data from the serial bus interface. See Table 4–21 for a complete description of the register contents.

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

Register: Serial bus data
Offset: B0h (Function 0)
Type: Read/Write

Default: 00h

Table 4-21. Serial Bus Data Register Description

BIT	SIGNAL	TYPE	FUNCTION
7-0‡	SBDATA		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.

[‡] One or more bits in this register are cleared only by the assertion of GRST.

4.47 Serial Bus Index Register

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

On byte reads, the word address is programmed into this register, the serial bus slave address must be programmed with both the 7-bit slave address and the read/write indicator bit, and bit 5 (REQBUSY) in the serial bus control and status register (see Section 4.49) must be polled until clear. Then the contents of the serial bus data register are valid read data from the serial bus interface. See Table 4–22 for a complete description of the register contents.

Bit	7	6	5	4	3	2	1	0
Name				Serial bu	ıs index			
Туре	RW	RW	RW	RW	RW	RW	RW	RW
Default	0	0	0	0	0	0	0	0

Register: Serial bus index
Offset: B1h (Function 0)
Type: Read/Write

Table 4–22. Serial Bus Index Register Description

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

[‡] One or more bits in this register are cleared only by the assertion of GRST.

4.48 Serial Bus Slave Address Register

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

On byte reads, the byte address is programmed into the serial bus index register, this register must be programmed with both the 7-bit slave address and the read/write indicator bit, and bit 5 (REQBUSY) in the serial bus control and status register (see Section 4.49) must be polled until clear. Then the contents of the serial bus data register are valid read data from the serial bus interface. See Table 4–23 for a complete description of the register contents.

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

Register: Serial bus slave address

Offset: B2h (Function 0)
Type: Read/Write

Table 4-23. Serial Bus Slave Address Register Description

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

[‡] One or more bits in this register are cleared only by the assertion of GRST.

4.49 Serial Bus Control/Status Register

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

Bit	7	6	5	4	3	2	1	0
Name		Serial bus control/status						
Туре	RW	R	R	R	RW	RW	RC	RC
Default	0	0	0	0	0	0	0	0

Register: Serial bus control/status

Offset: B3h (Function 0)

Type: Read-only, Read/Write, Read/Clear

Table 4-24. Serial Bus Control/Status Register Description

BIT	SIGNAL	TYPE	FUNCTION
7‡	PROT_SEL	RW	Protocol select. When bit 7 is set, the send-byte protocol is used on write requests and the receive-byte protocol is used on read commands. The word address byte in the serial bus index register (see Section 4.47) is not output by the PCI1515 controller 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 4.48). Bit 5 must be polled on reads from the serial interface. After the byte read access has been completed, this bit is cleared and the read data is valid in the serial bus data register.
4	ROMBUSY	R	Serial EEPROM busy status. Bit 4 indicates the status of the PCI1515 serial EEPROM circuitry. Bit 4 is set during the loading of the subsystem ID and other default values from the serial bus EEPROM. 0 = Serial EEPROM circuitry is not busy 1 = Serial EEPROM circuitry is busy
3‡	SBDETECT	RW	Serial bus detect. When the serial bus interface is detected through a pullup resistor on the SCL terminal after reset, this bit is set to 1. 0 = Serial bus interface not detected 1 = Serial bus interface detected
2‡	SBTEST	RW	Serial bus test. When bit 2 is set, the serial bus clock frequency is increased for test purposes. 0 = Serial bus clock at normal operating frequency, ≈ 100 kHz (default) 1 = Serial bus clock frequency increased for test purposes
1 ‡	REQ_ERR	RC	Requested serial bus access error. Bit 1 indicates when a data error occurs on the serial interface during a requested cycle and may be set due to a missing acknowledge. Bit 1 is cleared by a writeback of 1. 0 = No error detected during user-requested byte read or write cycle 1 = Data error detected during user-requested byte read or write cycle
0‡	ROM_ERR	RC	EEPROM data error status. Bit 0 indicates when a data error occurs on the serial interface during the auto-load from the serial bus EEPROM and may be set due to a missing acknowledge. Bit 0 is also set on invalid EEPROM data formats. See Section 3.6.4, Serial Bus EEPROM Application, for details on EEPROM data format. Bit 0 is cleared by a writeback of 1. 0 = No error detected during autoload from serial bus EEPROM 1 = Data error detected during autoload from serial bus EEPROM

[‡]One or more bits in this register are cleared only by the assertion of GRST.

5 ExCA Compatibility Registers (Function 0)

The ExCA (exchangeable card architecture) registers implemented in the PCI1515 controller are register-compatible with the Intel 82365SL-DF PCMCIA controller. ExCA registers are identified by an offset value, which is compatible with the legacy I/O index/data scheme used on the Intel™ 82365 ISA controller. The ExCA registers are accessed through this scheme by writing the register offset value into the index register (I/O base), and reading or writing the data register (I/O base + 1). The I/O base address used in the index/data scheme is programmed in the PC Card 16-bit I/F legacy mode base address register. The offsets from this base address run contiguously from 00h to 3Fh for socket A. See Figure 5–1 for an ExCA I/O mapping illustration. Table 5–1 identifies each ExCA register and its respective ExCA offset.

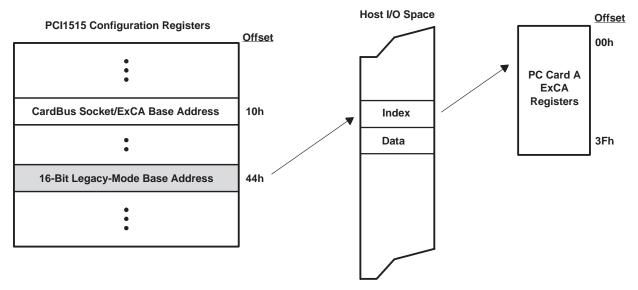
The PCI1515 controller also provides a memory-mapped alias of the ExCA registers by directly mapping them into PCI memory space. They are located through the CardBus socket registers/ExCA registers base address register (PCI register 10h) at memory offset 800h. See Figure 5–2 for an ExCA memory mapping illustration. Note that memory offsets are 800h–844h for function 0. This illustration also identifies the CardBus socket register mapping, which is mapped into the same 4K window at memory offset 0h.

The interrupt registers in the ExCA register set, as defined by the 82365SL specification, control such card functions as reset, type, interrupt routing, and interrupt enables. Special attention must be paid to the interrupt routing registers and the host interrupt signaling method selected for the PCI1515 controller to ensure that all possible PCI1515 interrupts can potentially be routed to the programmable interrupt controller. The ExCA registers that are critical to the interrupt signaling are at memory address ExCA offsets 803h and 805h.

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

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



Offset of desired register is placed in the index register and the data from that location is returned in the data register.

Host **PCI1515 Configuration Registers** Memory Space Offset Offset 10h (Function 0) 00h CardBus Socket A Registers CardBus Socket/ExCA Base Address 20h 800h **ExCA** 16-Bit Legacy-Mode Base Address 44h Registers Card A 844h

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

Offsets are from the CardBus socket/ExCA base address register's base address.

Figure 5-2. ExCA Register Access Through Memory

Table 5-1. ExCA Registers and Offsets

EXCA REGISTER NAME	PCI MEMORY ADDRESS OFFSET (HEX)	EXCA OFFSET (CARD A)
Identification and revision ‡	800	00
Interface status	801	01
Power control †	802†	02
Interrupt and general control †	803†	03
Card status change †	804†	04
Card status change interrupt configuration †	805†	05
Address window enable	806	06
I / O window control	807	07
I / O window 0 start-address low-byte	808	08
I / O window 0 start-address high-byte	809	09
I / O window 0 end-address low-byte	80A	0A
I / O window 0 end-address high-byte	80B	0B
I / O window 1 start-address low-byte	80C	0C
I / O window 1 start-address high-byte	80D	0D
I / O window 1 end-address low-byte	80E	0E
I / O window 1 end-address high-byte	80F	0F
Memory window 0 start-address low-byte	810	10
Memory window 0 start-address high-byte	811	11
Memory window 0 end-address low-byte	812	12
Memory window 0 end-address high-byte	813	13
Memory window 0 offset-address low-byte	814	14
Memory window 0 offset-address high-byte	815	15
Card detect and general control †	816	16
Reserved	817	17
Memory window 1 start-address low-byte	818	18
Memory window 1 start-address high-byte	819	19
Memory window 1 end-address low-byte	81A	1A
Memory window 1 end-address high-byte	81B	1B
Memory window 1 offset-address low-byte	81C	1C
Memory window 1 offset-address high-byte	81D	1D
Global control ‡	81E	1E
Reserved	81F	1F
Memory window 2 start-address low-byte	820	20
Memory window 2 start-address high-byte	821	21
Memory window 2 end-address low-byte	822	22
Memory window 2 end-address high-byte	823	23
Memory window 2 offset-address low-byte	824	24
Memory window 2 offset-address high-byte	825	25

[†] One or more bits in this register are cleared only by the assertion of GRST when PME is enabled. If PME is not enabled, then this bit is cleared by the assertion of PRST or GRST.

‡ One or more bits in this register are cleared only by the assertion of GRST.

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

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

5.1 ExCA Identification and Revision Register

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

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

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

Register: **ExCA identification and revision**

Offset: CardBus Socket Address + 800h: Card A ExCA Offset 00h

Type: Read/Write, Read-only

Default: 84h

Table 5–2. ExCA Identification and Revision Register Description

BIT	SIGNAL	TYPE	FUNCTION	
7–6‡	IFTYPE	R	Interface type. These bits, which are hardwired as 10b, identify the 16-bit PC Card support provided by PCI1515 controller. The PCI1515 controller supports both I/O and memory 16-bit PC Cards.	
5-4 ‡	RSVD	RW	These bits can be used for 82365SL emulation.	
3-0‡	365REV	RW	82365SL-DF revision. This field stores the Intel 82365SL-DF revision supported by the PCI1515 controller. Host software can read this field to determine compatibility to the 82365SL-DF register set. This field defaults to 0100b upon reset. Writing 0010b to this field places the controller in the 82356SL mode.	

[‡] One or more bits in this register are cleared only by the assertion of GRST.

5.2 ExCA Interface Status Register

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

Bit	7	6	5	4	3	2	1	0
Name	ExCA interface status							
Туре	R	R	R	R	R	R	R	R
Default	0	0	Х	Х	Х	Х	Х	Х

Register: **ExCA interface status**

Offset: CardBus Socket Address + 801h: Card A ExCA Offset 01h

Type: Read-only Default: 00XX XXXXb

Table 5-3. ExCA Interface Status Register Description

BIT	SIGNAL	TYPE	FUNCTION	
7	RSVD	R	This bit returns 0 when read. A write has no effect.	
6	CARDPWR	CARDPWR R	CARDPWR. Card power. This bit indicates the current power status of the PC Card socket. This bit reflects how the ExCA power control register has been programmed. The bit is encoded as: 0 = V _{CC} and V _{PP} to the socket are turned off (default).	
			1 = V _{CC} and V _{PP} to the socket are turned on.	
			This bit indicates the current status of the READY signal at the PC Card interface.	
5	READY	R	0 = PC Card is not ready for a data transfer. 1 = PC Card is ready for a data transfer.	
4	CARDWP	R	Card write protect. This bit indicates the current status of the WP signal at the PC Card interface. This signal reports to the PCI1515 controller whether or not the memory card is write protected. Further, write protection for an entire PCI1515 16-bit memory window is available by setting the appropriate bit in the ExCA memory window offset-address high-byte register.	
			0 = WP signal is 0. PC Card is R/W. 1 = WP signal is 1. PC Card is read-only.	
3	CDETECT2	R	Card detect 2. This bit indicates the status of the CD2 signal at the PC Card interface. Software can use this and CDETECT1 to determine if a PC Card is fully seated in the socket.	
3	CDETECT2		0 = <u>CD2</u> signal is 1. No PC Card inserted. 1 = <u>CD2</u> signal is 0. PC Card at least partially inserted.	
2	CDETECT4	R	Card detect 1. This bit indicates the status of the CD1 signal at the PC Card interface. Software can use this and CDETECT2 to determine if a PC Card is fully seated in the socket.	
2	CDETECT1	CDETECT	K	0 = CD1 signal is 1. No PC Card inserted. 1 = CD1 signal is 0. PC Card at least partially inserted.
			Battery voltage detect. When a 16-bit memory card is inserted, the field indicates the status of the battery voltage detect signals (BVD1, BVD2) at the PC Card interface, where bit 0 reflects the BVD1 status, and bit 1 reflects BVD2.	
1–0	BVDSTAT	R	00 = Battery is dead. 01 = Battery is dead. 10 = Battery is low; warning. 11 = Battery is good.	
			When a 16-bit I/O card is inserted, this field indicates the status of the SPKR (bit 1) signal and the STSCHG (bit 0) at the PC Card interface. In this case, the two bits in this field directly reflect the current state of these card outputs.	

5.3 ExCA Power Control Register

This register provides PC Card power control. Bit 7 of this register enables the 16-bit outputs on the socket interface, and can be used for power management in 16-bit PC Card applications. See Table 5–5 for a complete description of the register contents.

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

Register: **ExCA power control**

Offset: CardBus Socket Address + 802h: Card A ExCA Offset 02h

Type: Read-only, Read/Write

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

BIT	SIGNAL	TYPE	FUNCTION			
7	COE	RW	Card output enable. Bit 7 controls the state of all of the 16-bit outputs on the PCI1515 controller. This bit is encoded as: 0 = 16-bit PC Card outputs disabled (default) 1 = 16-bit PC Card outputs enabled			
6	RSVD	R	Reserved. Bit 6 returns 0 when read.			
5†	AUTOPWRSWEN	RW	Auto power switch enable. 0 = Automatic socket power switching based on card detects is disabled. 1 = Automatic socket power switching based on card detects is enabled.			
4	CAPWREN	RW	PC Card power enable. 0 = V _{CC} = No connection 1 = V _{CC} is enabled and controlled by bit 2 (EXCAPOWER) of the system control register (PCI offset 80h, see Section 4.29).			
3–2	RSVD	R	Reserved. Bits 3 and 2 return 0s when read.			
1–0	EXCAVPP	RW	PC Card Vpp power control. Bits 1 and 0 are used to request changes to card Vpp. The PCI1515 controller ignores this field unless V _{CC} to the socket is enabled. This field is encoded as: 00 = No connection (default) 10 = 12 V 01 = V _{CC} 11 = Reserved			

[†] One or more bits in this register are cleared only by the assertion of GRST when PME is enabled. If PME is not enabled, then this bit is cleared by the assertion of PRST or GRST.

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

BIT	SIGNAL	TYPE	FUNCTION			
7†	COE	RW	Card output enable. This bit controls the state of all of the 16-bit outputs on the PCI1515 controller. This bit is encoded as: 0 = 16-bit PC Card outputs are disabled (default). 1 = 16-bit PC Card outputs are enabled.			
6–5	RSVD	R	Reserved. These bits return 0s when read. Writes have no effect.			
4–3†	EXCAVCC	RW	V_{CC} . These bits are used to request changes to card V_{CC} . This field is encoded as: 00 = 0 V (default) 10 = 5 V 01 = 0 V reserved 11 = 3.3 V			
2	RSVD	R	This bit returns 0 when read. A write has no effect.			
1–0 †	EXCAVPP	RW	Vpp. These bits are used to request changes to card Vpp. The PCI1515 controller ignores this fiel V _{CC} to the socket is enabled (i.e., 5 Vdc or 3.3 Vdc). This field is encoded as: 00 = 0 V (default) 10 = 12 V 11 = 0 V reserved			

[†] This bit is cleared only by the assertion of GRST when PME is enabled. If PME is not enabled, then this bit is cleared by the assertion of PRST or GRST.

5.4 ExCA Interrupt and General Control Register

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

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

Register: ExCA interrupt and general control

Offset: CardBus Socket Address + 803h: Card A ExCA Offset 03h

Type: Read/Write

Default: 00h

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

BIT	SIGNAL	TYPE	FUNCTION
7	RINGEN	RW	Card ring indicate enable. Enables the ring indicate function of the BVD1/RI terminals. This bit is encoded as: 0 = Ring indicate disabled (default) 1 = Ring indicate enabled
6†	RESET	RW	Card reset. This bit controls the 16-bit PC Card RESET signal, and allows host software to force a card reset. This bit affects 16-bit cards only. This bit is encoded as: 0 = RESET signal asserted (default) 1 = RESET signal deasserted.
5†	CARDTYPE	RW	Card type. This bit indicates the PC Card type. This bit is encoded as: 0 = Memory PC Card is installed (default) 1 = I/O PC Card is installed
4	CSCROUTE	RW	PCI interrupt – CSC routing enable bit. This bit has meaning only if the CSC interrupt routing control bit (PCI offset 93h, bit 5) is 0. In this case, when this bit is set (high), the card status change interrupts are routed to PCI interrupts. When low, the card status change interrupts are routed using bits 7–4 in the ExCA card status-change interrupt configuration register (ExCA offset 805h, see Section 5.6). This bit is encoded as: 0 = CSC interrupts routed by ExCA registers (default) 1 = CSC interrupts routed to PCI interrupts If the CSC interrupt routing control bit (bit 5) of the diagnostic register (PCI offset 93h, see Section 4.39) is set to 1, this bit has no meaning, which is the default case.
3-0	INTSELECT	RW	Card interrupt select for I/O PC Card functional interrupts. These bits select the interrupt routing for I/O PC Card functional interrupts. This field is encoded as: 0000 = No IRQ selected (default). CSC interrupts are routed to PCI Interrupts. This bit setting is ORed with bit 4 (CSCROUTE) for backward compatibility. 0001 = IRQ1 enabled 0010 = SMI enabled 0011 = IRQ3 enabled 0100 = IRQ4 enabled 0110 = IRQ5 enabled 0111 = IRQ7 enabled 1000 = IRQ8 enabled 1001 = IRQ9 enabled 1001 = IRQ9 enabled 1101 = IRQ10 enabled 1101 = IRQ11 enabled 1101 = IRQ11 enabled 1101 = IRQ12 enabled 1101 = IRQ13 enabled 1111 = IRQ15 enabled

[†] This bit is cleared only by the assertion of GRST when PME is enabled. If PME is not enabled, then this bit is cleared by the assertion of PRST or GRST.

5.5 ExCA Card Status-Change Register

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

Bit	7	6	5	4	3	2	1	0		
Name		ExCA card status-change								
Туре	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

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

BIT	SIGNAL	TYPE	FUNCTION
7–4	RSVD	R	Reserved. Bits 7–4 return 0s when read.
3†	CDCHANGE	R	Card detect change. Bit 3 indicates whether a change on $\overline{\text{CD1}}$ or $\overline{\text{CD2}}$ occurred at the PC Card interface. This bit is encoded as: $0 = \text{No change detected on either } \overline{\text{CD1}} \text{ or } \overline{\text{CD2}}$ $1 = \text{Change detected on either } \overline{\text{CD1}} \text{ or } \overline{\text{CD2}}$
2†	READYCHANGE	R	Ready change. When a 16-bit memory is installed in the socket, bit 2 includes whether the source of a PCI1515 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 PCI1515 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 PCI1515 interrupt was due to a battery dead condition. This bit is encoded as: $0 = \frac{\text{STSCHG}}{\text{STSCHG}} \text{ deasserted (default)}$ $1 = \frac{\text{STSCHG}}{\text{STSCHG}} \text{ asserted}$ Ring indicate. When the PCI1515 is configured for ring indicate operation, bit 0 indicates the status of RI.

[†] These are PME context bits and can be cleared only by the assertion of GRST when PME is enabled. If PME is not enabled, then these bits are cleared by the assertion of PRST or GRST.

5.6 ExCA Card Status-Change Interrupt Configuration Register

This register controls interrupt routing for CSC interrupts, as well as masks/unmasks CSC interrupt sources. See Table 5–8 for a complete description of the register contents.

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

Register: ExCA card status-change interrupt configuration

Offset: CardBus Socket Address + 805h: Card A ExCA Offset 05h

Type: Read/Write

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

BIT	SIGNAL	TYPE	FUNCTION
			Interrupt select for card status change. These bits select the interrupt routing for card status-change interrupts. This field is encoded as:
7–4	CSCSELECT	RW	0000 = CSC interrupts routed to PCI interrupts if bit 5 of the diagnostic register (PCI offset 93h) is set to 1b. In this case bit 4 of ExCA 803 is a don't care. This is the default setting. 0000 = No ISA interrupt routing if bit 5 of the diagnostic register (PCI offset 93h) is set to 0b. In this case, CSC interrupts are routed to PCI interrupts by setting bit 4 of ExCA 803h to 1b. 0001 = IRQ1 enabled 0010 = SMI enabled 0011 = IRQ3 enabled 0100 = IRQ4 enabled 0110 = IRQ6 enabled 1011 = IRQ7 enabled 1000 = IRQ8 enabled 1001 = IRQ9 enabled 1011 = IRQ10 enabled 1011 = IRQ11 enabled 1101 = IRQ11 enabled 1101 = IRQ12 enabled 1101 = IRQ13 enabled 1111 = IRQ14 enabled
3†	CDEN	RW	Card detect enable. Enables interrupts on CD1 or CD2 changes. This bit is encoded as: 0 = Disables interrupts on CD1 or CD2 line changes (default) 1 = Enables interrupts on CD1 or CD2 line changes
2†	READYEN	RW	Ready enable. This bit enables/disables a low-to-high transition on the PC Card READY signal to generate a host interrupt. This interrupt source is considered a card status change. This bit is encoded as: 0 = Disables host interrupt generation (default) 1 = Enables host interrupt generation
1†	BATWARNEN	RW	Battery warning enable. This bit enables/disables a battery warning condition to generate a CSC interrupt. This bit is encoded as: 0 = Disables host interrupt generation (default) 1 = Enables host interrupt generation
0†	BATDEADEN	RW	Battery dead enable. This bit enables/disables a battery dead condition on a memory PC Card or assertion of the STSCHG I/O PC Card signal to generate a CSC interrupt. 0 = Disables host interrupt generation (default) 1 = Enables host interrupt generation

[†] This bit is cleared only by the assertion of GRST when PME is enabled. If PME is not enabled, then this bit is cleared by the assertion of PRST or GRST.

5.7 ExCA Address Window Enable Register

The ExCA address window enable register enables/disables the memory and I/O windows to the 16-bit PC Card. By default, all windows to the card are disabled. The PCI1515 controller 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								
Туре	RW	RW RW RW RW RW RW								
Default	0	0	0	0	0	0	0	0		

Register: ExCA address window enable

Type: Read-only, Read/Write

Offset: CardBus socket address + 806h; Card A ExCA offset 06h

Table 5-9. ExCA Address Window Enable Register Description

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

5.8 ExCA I/O Window Control Register

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

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

Register: ExCA I/O window control

Type: Read/Write

Offset: CardBus socket address + 807h: Card A ExCA offset 07h

Default: 00h

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

BIT	SIGNAL	TYPE	FUNCTION
7	WAITSTATE1	RW	I/O window 1 wait state. Bit 7 controls the I/O window 1 wait state for 16-bit I/O accesses. Bit 7 has no effect on 8-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 16-bit cycles have standard length (default). 1 = 16-bit cycles are extended by one equivalent ISA wait state.
6	ZEROWS1	RW	I/O window 1 zero wait state. Bit 6 controls the I/O window 1 wait state for 8-bit I/O accesses. Bit 6 has no effect on 16-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 8-bit cycles have standard length (default). 1 = 8-bit cycles are reduced to equivalent of three ISA cycles.
5	IOSIS16W1	RW	I/O window 1 IOIS16 source. Bit 5 controls the I/O window 1 automatic data-sizing feature that uses IOIS16 from the PC Card to determine the data width of the I/O data transfer. This bit is encoded as: 0 = Window data width determined by DATASIZE1, bit 4 (default). 1 = Window data width determined by IOIS16.
4	DATASIZE1	RW	I/O window 1 data size. Bit 4 controls the I/O window 1 data size. Bit 4 is ignored if bit 5 (IOSIS16W1) is set. This bit is encoded as: 0 = Window data width is 8 bits (default). 1 = Window data width is 16 bits.
3	WAITSTATE0	RW	I/O window 0 wait state. Bit 3 controls the I/O window 0 wait state for 16-bit I/O accesses. Bit 3 has no effect on 8-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 16-bit cycles have standard length (default). 1 = 16-bit cycles are extended by one equivalent ISA wait state.
2	ZEROWS0	RW	I/O window 0 zero wait state. Bit 2 controls the I/O window 0 wait state for 8-bit I/O accesses. Bit 2 has no effect on 16-bit accesses. This wait-state timing emulates the ISA wait state used by the Intel 82365SL-DF. This bit is encoded as: 0 = 8-bit cycles have standard length (default). 1 = 8-bit cycles are reduced to equivalent of three ISA cycles.
1	IOSIS16W0	RW	I/O window 0 IOIS16 source. Bit 1 controls the I/O window 0 automatic data sizing feature that uses IOIS16 from the PC Card to determine the data width of the I/O data transfer. This bit is encoded as: 0 = Window data width is determined by DATASIZE0, bit 0 (default). 1 = Window data width is determined by IOIS16.
0	DATASIZE0	RW	I/O window 0 data size. Bit 0 controls the I/O window 0 data size. Bit 0 is ignored if bit 1 (IOSIS16W0) is set. This bit is encoded as: 0 = Window data width is 8 bits (default). 1 = Window data width is 16 bits.

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

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

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

Register: ExCA I/O window 0 start-address low-byte

Offset: CardBus Socket Address + 808h: Card A ExCA Offset 08h

Register: ExCA I/O window 1 start-address low-byte

Offset: CardBus Socket Address + 80Ch: Card A ExCA Offset 0Ch

Type: Read/Write

Default: 00h

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

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

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

Register: ExCA I/O window 0 start-address high-byte

Offset: CardBus Socket Address + 809h: Card A ExCA Offset 09h

Register: ExCA I/O window 1 start-address high-byte

Offset: CardBus Socket Address + 80Dh: Card A ExCA Offset 0Dh

Type: Read/Write

Default: 00h

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

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

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

Register: ExCA I/O window 0 end-address low-byte

Offset: CardBus Socket Address + 80Ah: Card A ExCA Offset 0Ah

Register: ExCA I/O window 1 end-address low-byte

Offset: CardBus Socket Address + 80Eh: Card A ExCA Offset 0Eh

Type: Read/Write

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

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

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

Register: ExCA I/O window 0 end-address high-byte

Offset: CardBus Socket Address + 80Bh: Card A ExCA Offset 0Bh

Register: ExCA I/O window 1 end-address high-byte

Offset: CardBus Socket Address + 80Fh: Card A ExCA Offset 0Fh

Type: Read/Write

Default: 00h

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

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

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

Register: ExCA memory window 0 start-address low-byte

Offset: CardBus Socket Address + 810h: Card A ExCA Offset 10h

Register: ExCA memory window 1 start-address low-byte

Offset: CardBus Socket Address + 818h: Card A ExCA Offset 18h

Register: ExCA memory window 2 start-address low-byte

Offset: CardBus Socket Address + 820h: Card A ExCA Offset 20h

Register: ExCA memory window 3 start-address low-byte

Offset: CardBus Socket Address + 828h: Card A ExCA Offset 28h

Register: ExCA memory window 4 start-address low-byte

Offset: CardBus Socket Address + 830h: Card A ExCA Offset 30h

Type: Read/Write

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

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

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

Register: ExCA memory window 0 start-address high-byte

Offset: CardBus Socket Address + 811h: Card A ExCA Offset 11h

Register: ExCA memory window 1 start-address high-byte

Offset: CardBus Socket Address + 819h: Card A ExCA Offset 19h

Register: ExCA memory window 2 start-address high-byte

Offset: CardBus Socket Address + 821h: Card A ExCA Offset 21h

Register: ExCA memory window 3 start-address high-byte

Offset: CardBus Socket Address + 829h: Card A ExCA Offset 29h

Register: ExCA memory window 4 start-address high-byte

Offset: CardBus Socket Address + 831h: Card A ExCA Offset 31h

Type: Read/Write

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

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

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

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

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

Register: ExCA memory window 0 end-address low-byte

Offset: CardBus Socket Address + 812h: Card A ExCA Offset 12h

Register: ExCA memory window 1 end-address low-byte

Offset: CardBus Socket Address + 81Ah: Card A ExCA Offset 1Ah

Register: ExCA memory window 2 end-address low-byte

Offset: CardBus Socket Address + 822h: Card A ExCA Offset 22h

Register: ExCA memory window 3 end-address low-byte

Offset: CardBus Socket Address + 82Ah: Card A ExCA Offset 2Ah

Register: ExCA memory window 4 end-address low-byte

Offset: CardBus Socket Address + 832h: Card A ExCA Offset 32h

Type: Read/Write

Default: 00h

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

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

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

Register: ExCA memory window 0 end-address high-byte

Offset: CardBus Socket Address + 813h: Card A ExCA Offset 13h

Register: ExCA memory window 1 end-address high-byte

Offset: CardBus Socket Address + 81Bh: Card A ExCA Offset 1Bh

Register: ExCA memory window 2 end-address high-byte

Offset: CardBus Socket Address + 823h: Card A ExCA Offset 23h

Register: ExCA memory window 3 end-address high-byte

Offset: CardBus Socket Address + 82Bh: Card A ExCA Offset 2Bh

Register: ExCA Memory window 4 end-address high-byte

Offset: CardBus Socket Address + 833h: Card A ExCA Offset 33h

Type: Read/Write, Read-only

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

BIT	SIGNAL	TYPE	FUNCTION
7–6	MEMWS	RW	Wait state. These bits specify the number of equivalent ISA wait states to be added to 16-bit memory accesses. The number of wait states added is equal to the binary value of these 2 bits.
5–4	RSVD	R	Reserved. These bits return 0s when read. Writes have no effect.
3–0	ENDHN	RW	End-address high nibble. These bits represent the upper address bits A23–A20 of the memory window end address.

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

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

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

Register: ExCA memory window 0 offset-address low-byte

Offset: CardBus Socket Address + 814h: Card A ExCA Offset 14h

Register: ExCA memory window 1 offset-address low-byte

Offset: CardBus Socket Address + 81Ch: Card A ExCA Offset 1Ch

Register: ExCA memory window 2 offset-address low-byte

Offset: CardBus Socket Address + 824h: Card A ExCA Offset 24h

Register: ExCA memory window 3 offset-address low-byte

Offset: CardBus Socket Address + 82Ch: Card A ExCA Offset 2Ch

Register: ExCA memory window 4 offset-address low-byte

Offset: CardBus Socket Address + 834h: Card A ExCA Offset 34h

Type: Read/Write

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

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

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

Register: ExCA memory window 0 offset-address high-byte

Offset: CardBus Socket Address + 815h: Card A ExCA Offset 15h

Register: ExCA memory window 1 offset-address high-byte

Offset: CardBus Socket Address + 81Dh: Card A ExCA Offset 1Dh

Register: ExCA memory window 2 offset-address high-byte

Offset: CardBus Socket Address + 825h: Card A ExCA Offset 25h

Register: ExCA memory window 3 offset-address high-byte

Offset: CardBus Socket Address + 82Dh: Card A ExCA Offset 2Dh

Register: ExCA memory window 4 offset-address high-byte

Offset: CardBus Socket Address + 835h: Card A ExCA Offset 35h

Type: Read/Write

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

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

5.19 ExCA Card Detect and General Control Register

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

Bit	7	6	5	4	3	2	1	0			
Name	ExCA card detect and general control										
Туре	R	R R W RW R RW R									
Default	Х	Х	0	0	0	0	0	0			

Register: ExCA card detect and general control

Offset: CardBus Socket Address + 816h: Card A ExCA Offset 16h

Type: Read-only, Write-only, Read/Write

Default: XX00 0000b

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

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

[†] One or more bits in this register are cleared only by the assertion of GRST when PME is enabled. If PME is not enabled, then this bit is cleared by the assertion of PRST or GRST.

5.20 ExCA Global Control Register

This register controls the PC Card socket. The host interrupt mode bits in this register are retained for 82365SL-DF compatibility. See Table 5–15 for a complete description of the register contents.

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

Register: **ExCA global control**

Offset: CardBus Socket Address + 81Eh: Card A ExCA Offset 1Eh

Type: Read-only, Read/Write

Default: 00h

Table 5-15. ExCA Global Control Register Description

BIT	SIGNAL	TYPE	FUNCTION		
7–5	RSVD	R	These bits return 0s when read. Writes have no effect.		
4	INTMODEB	RW	Level/edge interrupt mode select, card B. This bit selects the signaling mode for the PCI1515 host interrupt for card B interrupts. This bit is encoded as:		
			0 = Host interrupt is edge mode (default). 1 = Host interrupt is level mode.		
3	INTMODEA	RW	Level/edge interrupt mode select, card A. This bit selects the signaling mode for the PCI1515 host interrupt for card A interrupts. This bit is encoded as:		
			0 = Host interrupt is edge-mode (default). 1 = Host interrupt is level-mode.		
2‡	IFCMODE	RW	Interrupt flag clear mode select. This bit selects the interrupt flag clear mechanism for the flags in the ExCA card status change register. This bit is encoded as:		
			0 = Interrupt flags cleared by read of CSC register (default) 1 = Interrupt flags cleared by explicit writeback of 1		
1‡	CSCMODE	RW	Card status change level/edge mode select. This bit selects the signaling mode for the PCI1515 host interrupt for card status changes. This bit is encoded as:		
			0 = Host interrupt is edge-mode (default). 1 = Host interrupt is level-mode.		
0‡	PWRDWN	RW	Power-down mode select. When this bit is set to 1, the PCI1515 controller is in power-down mode. In power-down mode the PCI1515 card outputs are placed in a high-impedance state until an active cycle is executed on the card interface. Following an active cycle the outputs are again placed in a high-impedance state. The PCI1515 controller still receives functional interrupts and/or card status change interrupts; however, an actual card access is required to wake up the interface. This bit is encoded as:		
			0 = Power-down mode disabled (default) 1 = Power-down mode enabled		

[‡] One or more bits in this register are cleared only by the assertion of GRST.

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

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

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

Register: ExCA I/O window 0 offset-address low-byte

Offset: CardBus Socket Address + 836h: Card A ExCA Offset 36h

Register: ExCA I/O window 1 offset-address low-byte

Offset: CardBus Socket Address + 838h: Card A ExCA Offset 38h

Type: Read/Write, Read-only

Default: 00h

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

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

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

Register: ExCA I/O window 0 offset-address high-byte

Offset: CardBus Socket Address + 837h: Card A ExCA Offset 37h

Register: ExCA I/O window 1 offset-address high-byte

Offset: CardBus Socket Address + 839h: Card A ExCA Offset 39h

Type: Read/Write

Default: 00h

5.23 ExCA Memory Windows 0-4 Page Registers

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

Bit	7	6	5	4	3	2	1	0						
Name	ExCA memory windows 0-4 page													
Туре	RW	RW	RW	RW	RW	RW	RW	R						
Default	0	0	0	0	0	0	0	0						

Register: ExCA memory windows 0-4 page

Offset: CardBus Socket Address + 840h, 841h, 842h, 843h, 844h

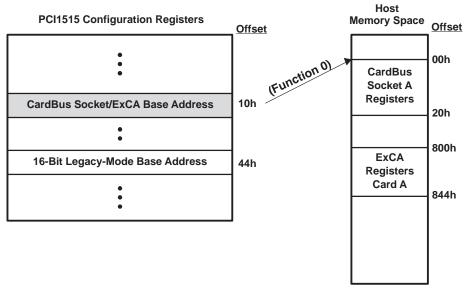
Type: Read/Write

Default: 00h

6 CardBus Socket Registers (Function 0)

The 1997 PC Card Standard requires a CardBus socket controller to provide five 32-bit registers that report and control socket-specific functions. The PCI1515 controller provides the CardBus socket/ExCA base address register (PCI offset 10h, see Section 4.12) to locate these CardBus socket registers in PCI memory address space. Table 6–1 gives the location of the socket registers in relation to the CardBus socket/ExCA base address.

In addition to the five required registers, the PCI1515 controller implements a register at offset 20h that provides power management control for the socket.



Offsets are from the CardBus socket/ExCA base address register's base address.

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

Table 6–1. CardBus Socket Registers

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

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

[‡]One or more bits in this register are cleared only by the assertion of GRST.

6.1 Socket Event Register

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

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								Socke	t event							
Туре	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								Socke	t event							
Туре	R	R	R	R	R	R	R	R	R	R	R	R	RWC	RWC	RWC	RWC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: Socket event

Offset: CardBus Socket Address + 00h Type: Read-only, Read/Write to Clear

Table 6-2. Socket Event Register Description

			· · · · · · · · · · · · · · · · · · ·
BIT	SIGNAL	TYPE	FUNCTION
31–4	RSVD	R	These bits return 0s when read.
3†	PWREVENT	RWC	Power cycle. This bit is set when the PCI1515 controller detects that the PWRCYCLE bit in the socket present state register (offset 08h, see Section 6.3) has changed. This bit is cleared by writing a 1.
2†	CD2EVENT	RWC	CCD2. This bit is set when the PCI1515 controller detects that the CDETECT2 field in the socket present state register (offset 08h, see Section 6.3) has changed. This bit is cleared by writing a 1.
1†	CD1EVENT	RWC	CCD1. This bit is set when the PCI1515 controller detects that the CDETECT1 field in the socket present state register (offset 08h, see Section 6.3) has changed. This bit is cleared by writing a 1.
0†	CSTSEVENT	RWC	CSTSCHG. This bit is set when the CARDSTS field in the socket present state register (offset 08h, see Section 6.3) has changed state. For CardBus cards, this bit is set on the rising edge of the CSTSCHG signal. For 16-bit PC Cards, this bit is set on both transitions of the CSTSCHG signal. This bit is reset by writing a 1.

[†] This bit is cleared only by the assertion of GRST when PME is enabled. If PME is not enabled, then this bit is cleared by the assertion of PRST or GRST.

6.2 Socket Mask Register

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

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								Socke	t mask							
Туре	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								Socke	t mask							
Туре	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: Socket mask

Offset: CardBus Socket Address + 04h

Type: Read-only, Read/Write

Table 6-3. Socket Mask Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–4	RSVD	R	These bits return 0s when read.
3†	DWDMACK	RW	Power cycle. This bit masks the PWRCYCLE bit in the socket present state register (offset 08h, see Section 6.3) from causing a status change interrupt.
31	PWRMASK	IXVV	0 = PWRCYCLE event does not cause a CSC interrupt (default). 1 = PWRCYCLE event causes a CSC interrupt.
			Card detect mask. These bits mask the CDETECT1 and CDETECT2 bits in the socket present state register (offset 08h, see Section 6.3) from causing a CSC interrupt.
2–1†	CDMASK	RW	00 = Insertion/removal does not cause a CSC interrupt (default). 01 = Reserved (undefined) 10 = Reserved (undefined) 11 = Insertion/removal causes a CSC interrupt.
ot	CSTSMASK	DW	CSTSCHG mask. This bit masks the CARDSTS field in the socket present state register (offset 08h, see Section 6.3) from causing a CSC interrupt.
01	CSTSWASK	RW	0 = CARDSTS event does not cause a CSC interrupt (default). 1 = CARDSTS event causes a CSC interrupt.

[†] This bit is cleared only by the assertion of GRST when PME is enabled. If PME is not enabled, then this bit is cleared by the assertion of PRST or GRST.

6.3 Socket Present State Register

This register reports information about the socket interface. Writes to the socket force event register (offset 0Ch, see Section 6.4), as well as general socket interface status, are reflected here. Information about PC Card V_{CC} support and card type is only updated at each insertion. Also note that the PCI1515 controller uses the $\overline{CCD1}$ and $\overline{CCD2}$ signals during card identification, and changes on these signals during this operation are not reflected in this register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Socket present state															
Туре	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							S	ocket pre	esent sta	te						
Туре	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	Х	0	0	0	Х	Х	Х

Register: Socket present state

Offset: CardBus Socket Address + 08h

Type: Read-only Default: 3000 00XXh

Table 6-4. Socket Present State Register Description

BIT	SIGNAL	TYPE	FUNCTION
31	YVSOCKET	R	YV socket. This bit indicates whether or not the socket can supply $V_{CC} = Y.Y V$ to PC Cards. The PCI1515 controller does not support Y.Y-V V_{CC} ; therefore, this bit is always reset unless overridden by the socket force event register (offset 0Ch, see Section 6.4). This bit defaults to 0.
30	XVSOCKET	R	XV socket. This bit indicates whether or not the socket can supply $V_{CC} = X.X V$ to PC Cards. The PCI1515 controller does not support X.X-V V_{CC} ; therefore, this bit is always reset unless overridden by the socket force event register (offset 0Ch, see Section 6.4). This bit defaults to 0.
29	3VSOCKET	R	3-V socket. This bit indicates whether or not the socket can supply $V_{CC} = 3.3$ Vdc to PC Cards. The PCI1515 controller does support 3.3-V V_{CC} ; therefore, this bit is always set unless overridden by the socket force event register (offset 0Ch, see Section 6.4).
28	5VSOCKET	R	5-V socket. This bit indicates whether or not the socket can supply V_{CC} = 5 Vdc to PC Cards. The PCI1515 controller does support 5-V V_{CC} ; therefore, this bit is always set unless overridden by bit 6 of the device control register (PCI offset 92h, see Section 4.38).
27–14	RSVD	R	These bits return 0s when read.
13 †	YVCARD	R	YV card. This bit indicates whether or not the PC Card inserted in the socket supports $V_{CC} = Y.Y Vdc$. This bit can be set by writing a 1 to the corresponding bit in the socket force event register (offset 0Ch, see Section 6.4).
12 †	XVCARD	R	XV card. This bit indicates whether or not the PC Card inserted in the socket supports $V_{CC} = X.X \text{ Vdc}$. This bit can be set by writing a 1 to the corresponding bit in the socket force event register (offset 0Ch, see Section 6.4).
11 †	3VCARD	R	3-V card. This bit indicates whether or not the PC Card inserted in the socket supports $V_{CC} = 3.3$ Vdc. This bit can be set by writing a 1 to the corresponding bit in the socket force event register (offset 0Ch, see Section 6.4).
10 †	5VCARD	R	5-V card. This bit indicates whether or not the PC Card inserted in the socket supports $V_{CC} = 5$ Vdc. This bit can be set by writing a 1 to the corresponding bit in the socket force event register (offset 0Ch, see Section 6.4).

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

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

BIT	SIGNAL	TYPE	FUNCTION
9†	BADVCCREQ	R	Bad V _{CC} request. This bit indicates that the host software has requested that the socket be powered at an invalid voltage. 0 = Normal operation (default) 1 = Invalid V _{CC} request by host software
8†	DATALOST	R	Data lost. This bit indicates that a PC Card removal event may have caused lost data because the cycle did not terminate properly or because write data still resides in the PCI1515 controller. 0 = Normal operation (default) 1 = Potential data loss due to card removal
7 †	NOTACARD	R	Not a card. This bit indicates that an unrecognizable PC Card has been inserted in the socket. This bit is not updated until a valid PC Card is inserted into the socket. 0 = Normal operation (default) 1 = Unrecognizable PC Card detected
6	IREQCINT	R	READY(IREQ)//CINT. This bit indicates the current status of the READY(IREQ)//CINT signal at the PC Card interface. 0 = READY(IREQ)//CINT is low. 1 = READY(IREQ)//CINT is high.
5†	CBCARD	R	CardBus card detected. This bit indicates that a CardBus PC Card is inserted in the socket. This bit is not updated until another card interrogation sequence occurs (card insertion).
4 †	16BITCARD	R	16-bit card detected. This bit indicates that a 16-bit PC Card is inserted in the socket. This bit is not updated until another card interrogation sequence occurs (card insertion).
3 †	PWRCYCLE	R	Power cycle. This bit indicates the status of each card powering request. This bit is encoded as: 0 = Socket is powered down (default). 1 = Socket is powered up.
2†	CDETECT2	R	CCD2. This bit reflects the current status of the CCD2 signal at the PC Card interface. Changes to this signal during card interrogation are not reflected here. 0 = CCD2 is low (PC Card may be present) 1 = CCD2 is high (PC Card not present)
1†	CDETECT1	R	CCD1. This bit reflects the current status of the CCD1 signal at the PC Card interface. Changes to this signal during card interrogation are not reflected here. 0 = CCD1 is low (PC Card may be present). 1 = CCD1 is high (PC Card not present).
0	CARDSTS	R	CSTSCHG. This bit reflects the current status of the CSTSCHG signal at the PC Card interface. 0 = CSTSCHG is low. 1 = CSTSCHG is high.

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

6.4 Socket Force Event Register

This register is used to force changes to the socket event register (offset 00h, see Section 6.1) and the socket present state register (offset 08h, see Section 6.3). The CVSTEST bit (bit 14) in this register must be written when forcing changes that require card interrogation. See Table 6–5 for a complete description of the register contents.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Socket force event															
Туре	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							S	Socket fo	rce ever	nt						
Туре	R	W	W	W	W	W	W	W	W	R	W	W	W	W	W	W
Default	Х	Х	Х	Х	Х	Х	Х	Х	Х	Х	Х	Х	Х	Х	Х	Х

Register: Socket force event

Offset: CardBus Socket Address + 0Ch

Type: Read-only, Write-only

Default: 0000 XXXXh

Table 6-5. Socket Force Event Register Description

BIT	SIGNAL	TYPE	FUNCTION
31–15	RSVD	R	Reserved. These bits return 0s when read.
14	CVSTEST	W	Card VS test. When this bit is set, the PCI1515 controller reinterrogates the PC Card, updates the socket present state register (offset 08h, see Section 6.3), and re-enables the socket power control.
13	FYVCARD	W	Force YV card. Writes to this bit cause the YVCARD bit in the socket present state register (offset 08h, see Section 6.3) to be written. When set, this bit disables the socket power control.
12	FXVCARD	W	Force XV card. Writes to this bit cause the XVCARD bit in the socket present state register (offset 08h, see Section 6.3) to be written. When set, this bit disables the socket power control.
11	F3VCARD	W	Force 3-V card. Writes to this bit cause the 3VCARD bit in the socket present state register (offset 08h, see Section 6.3) to be written. When set, this bit disables the socket power control.
10	F5VCARD	W	Force 5-V card. Writes to this bit cause the 5VCARD bit in the socket present state register (offset 08h, see Section 6.3) to be written. When set, this bit disables the socket power control.
9	FBADVCCREQ	W	Force BadVccReq. Changes to the BADVCCREQ bit in the socket present state register (offset 08h, see Section 6.3) can be made by writing this bit.
8	FDATALOST	W	Force data lost. Writes to this bit cause the DATALOST bit in the socket present state register (offset 08h, see Section 6.3) to be written.
7	FNOTACARD	W	Force not a card. Writes to this bit cause the NOTACARD bit in the socket present state register (offset 08h, see Section 6.3) to be written.
6	RSVD	R	This bit returns 0 when read.
5	FCBCARD	W	Force CardBus card. Writes to this bit cause the CBCARD bit in the socket present state register (offset 08h, see Section 6.3) to be written.
4	F16BITCARD	W	Force 16-bit card. Writes to this bit cause the 16BITCARD bit in the socket present state register (offset 08h, see Section 6.3) to be written.
3	FPWRCYCLE	W	Force power cycle. Writes to this bit cause the PWREVENT bit in the socket event register (offset 00h, see Section 6.1) to be written, and the PWRCYCLE bit in the socket present state register (offset 08h, see Section 6.3) is unaffected.
2	FCDETECT2	W	Force CCD2. Writes to this bit cause the CD2EVENT bit in the socket event register (offset 00h, see Section 6.1) to be written, and the CDETECT2 bit in the socket present state register (offset 08h, see Section 6.3) is unaffected.
1	FCDETECT1	W	Force CCD1. Writes to this bit cause the CD1EVENT bit in the socket event register (offset 00h, see Section 6.1) to be written, and the CDETECT1 bit in the socket present state register (offset 08h, see Section 6.3) is unaffected.
0	FCARDSTS	W	Force CSTSCHG. Writes to this bit cause the CSTSEVENT bit in the socket event register (offset 00h, see Section 6.1) to be written. The CARDSTS bit in the socket present state register (offset 08h, see Section 6.3) is unaffected.

6.5 Socket Control Register

This register provides control of the voltages applied to the socket V_{PP} and V_{CC} . The PCI1515 controller 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							
Туре	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							
Туре	R	R	R	R	R	R	RW	R	RW	RW	RW	RW	R	RW	RW	RW
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: Socket control

Offset: CardBus Socket Address + 10h

Type: Read-only, Read/Write

Table 6-6. Socket Control Register Description

BIT	SIGNAL	TYPE	FU	JNCTION
31–11	RSVD	R	These bits return 0s when read.	
10	RSVD	R	This bit returns 1 when read.	
9–8	RSVD	R	These bits return 0s when read.	
			to the CardBus card:	tate machine decides when to stop the CardBus clock
7	STOPCLK	RW	·	only attempt to stop/slow the CaredBus clock if the ne PCI CLKRUN protocol is preparing to stop/slow the
			·	only attempt to stop/slow the CaredBus clock if the ardless of the state of the PCI CLKRUN signal.
			V _{CC} control. These bits are used to request	card V _{CC} changes.
			000 = Request power off (default)	
6–4 †	VCCCTRL	RW	001 = Reserved	101 = Request V _{CC} = Y.Y V
			010 = Request V _{CC} = 5 V	110 = Reserved 111 = Reserved
	201/2		011 = Request V _{CC} = 3.3 V	TTT = Reserved
3	RSVD	R	This bit returns 0 when read.	
			Vpp control. These bits are used to request	card Vpp changes.
			000 = Request power off (default)	100 = Request Vpp = X.X V
2–0 †	VPPCTRL	RW	001 = Request Vpp = 12 V	101 = Request Vpp = Y.Y V
			010 = Request V _{PP} = 5 V	110 = Reserved
			011 = Request V _{PP} = 3.3 V	111 = Reserved

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

6.6 Socket Power Management Register

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

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							Socke	et power	manage	ement						
Туре	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							Socke	et power	manage	ement						
Туре	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register: Socket power management
Offset: CardBus Socket Address + 20h

Type: Read-only, Read/Write

Table 6-7. Socket Power Management Register Description

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

[‡]One or more bits in this register are cleared only by the assertion of GRST.

7 Electrical Characteristics

7.1 Absolute Maximum Ratings Over Operating Temperature Ranges[†]

Supply voltage range, VR_PORT	0.5 V to 1.836 V
V _{CC}	0.3 V to 4 V
V _{CCA}	–0.5 V to 5.5 V
V _{CCP}	–0.5 V to 5.5 V
Clamping voltage range, V _{CCP} and V _{CCCB}	0.5 V to 6 V
Input voltage range, V _I : PCI, CardBus, SC, miscellaneous	0.5 V to V _{CC} + 0.5 V
Output voltage range, VO: PCI, CardBus, SC, miscellaneous	
Input clamp current, I_{IK} ($V_I < 0$ or $V_I > V_{CC}$) (see Note 1)	
Output clamp current, I_{OK} ($V_O < 0$ or $V_O > V_{CC}$) (see Note 2)	
Operating free-air temperature, T _A	
Storage temperature range, T _{Stq}	–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 and miscellaneous terminals are measured with respect to V_{CCP} instead of V_{CC}. PC Card terminals are measured with respect to CardBus V_{CC}. The limit specified applies for a dc condition.
 - Applies for external output and bidirectional buffers. V_O > V_{CC} does not apply to fail-safe terminals. PCI terminals and miscellaneous terminals are measured with respect to V_{CCP} instead of V_{CC}. PC Card terminals are measured with respect to CardBus V_{CC}. The limit specified applies for a dc condition.

7.2 Recommended Operating Conditions (see Note 3)

		OPERATION	MIN	NOM	MAX	UNIT
VR_PORT	(see Table 2–4 for description)	1.5 V	1.35	1.5	1.65	V
Vcc		3.3 V	3	3.3	3.6	V
V _{CCP}		3.3 V	3	3.3	3.6	.,
	PCI and miscellaneous I/O clamp voltage	5 V	4.75	5	5.25	
V	DC Cond I/O algreen violations	3.3 V	3	3.3	3.6	
VCCA	PC Card I/O clamp voltage	5 V	4.75	5	5.25	

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

Recommended Operating Conditions (continued)

			OPERATION	MIN	NOM	MAX	UNIT	
		PCI [★]	3.3 V	0.5 VCCP		VCCP		
		PCI*	5 V	2		VCCP]	
			3.3 V CardBus 0.475 V _{CC(A/B)}			VCC(A/B)		
∨ _{IH} †		PC Card	3.3 V 16-bit	2			V	
	† Low-level input voltage Input voltage Output voltage Input transition time (t _r and t _f) Powerup reset time		5 V 16-bit	2.4		VCC(A/B)		
		Miscellaneous‡		2		VCC		
		TEST1:3	3.3 V 16-bit 2					
		PCI☆	3.3 V	0		0.3 V _{CCP}		
		PCI^	5 V	0		0.8		
+	Low-level input		3.3 V CardBus	0	(0.325 V _{CC(A/B)}] ,,	
V _{IL} †	voltage	PC Card	3.3 V 16-bit	0				
			5 V 16-bit	0		0.8		
		Miscellaneous‡		0		0.8		
		PCI [☆]		0		VCCP		
.,	lanut valta sa	PC Card		0		VCCCB	,,	
VI	input voitage	Miscellaneous‡		0		VCC	V	
		TEST1:3		0		VCC(A/B) VCC(A/B) VCC(A/B) VCC VCC VCC 0.3 VCCP 0.8 0.325 VCC(A/B) 0.8 0.8 VCCP VCCC VCC VCC 0.2 VCC		
		PCI [☆]		0		VCC		
∨o§	Output voltage	PC Card		0		VCC	V	
		Miscellaneous‡		0		Vcc]	
	Input transition time	PCI and PC Card		1		4		
t _t	$(t_f \text{ and } t_f)$	Miscellaneous‡		0		6	ns	
tpU	Powerup reset time	GRST input		2			ms	
TA	Operating ambient te	mperature range		0	25	70	°C	
TJ#	Virtual junction temper	erature		0	25	115	°C	

[†] Applies to external inputs and bidirectional buffers without hysteresis ‡ Miscellaneous terminals are: A9, B9, C9, G2, G3, J5, K5, P12, P17 (CLOCK, DATA, LATCH, SCL, SDA, SUSPEND, GRST, TEST0, TEST4

[§] Applies to external output buffers

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

[★]MFUNC(0:6) share the same specifications as the PCI terminals.

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

	PARAMETER	TERMINALS	OPERATION	TEST CONDITIONS	MIN	MAX	UNIT
		DO!¶	3.3 V	$I_{OH} = -0.5 \text{ mA}$	0.9 V _{CC}		
		PCIII	5 V	I _{OH} = -2 mA	2.4		
.,			3.3 V CardBus	$I_{OH} = -0.15 \text{ mA}$	0.9 V _{CC}		
VOH	High-level output voltage	PC Card	3.3 V 16-bit	$I_{OH} = -0.15 \text{ mA}$	2.4		V
			5 V 16-bit	$I_{OH} = -0.15 \text{ mA}$	2.8		
	High-level output voltage PC S S S S S S S S S	I _{OH} = -4 mA	V _{CC} -0.6				
		Pog	3.3 V	I _{OL} = 1.5 mA		0.1 V _{CC}	
		PCIII	5 V	I _{OL} = 6 mA		0.55	
.,	Lavo lavol autout valta va		3.3 V CardBus	I _{OL} = 0.7 mA		0.1 V _{CC}	.,
VOL	Low-level output voltage	PC Card	3.3 V 16-bit	I _{OL} = 0.7 mA	0.4		V
			5 V 16-bit	I _{OL} = 0.7 mA		0.55	
		Miscellaneous§	-	I _{OL} = 4 mA		0.5	
loz		Output terminals	3.6 V	$V_O = V_{CC}$ or GND		±20	μΑ
	High-impedance.		3.6 V	$V_I = V_{CC}$		-1	
lozL		Output terminals	5.25 V	VI = VCC		-1	μΑ
	High-impedance,		3.6 V	$V_I = V_{CC}^{\dagger}$		10	
lozh	•	Output terminals	5.25 V	$V_I = V_{CC}^{\dagger}$		25	μΑ
		Input terminals	3.6 V	V _I = GND		±20	_
IIL	Low-level input current	I/O terminals	3.6 V	V _I = GND	±20		μΑ
		PCI¶	3.6 V	$V_I = V_{CC}^{\ddagger}$		±20	
		Others	3.6 V	$V_I = V_{CC}^{\ddagger}$	±20		
			3.6 V			10	
'IH	High-level input current	Input terminals	5.25 V	$V_I = V_{CC}^{\ddagger}$		20	μΑ
			3.6 V	$V_1 = V_{CC}^{\ddagger}$		10	
		I/O terminals	5.25 V	$V_I = V_{CC}^{\dagger}$		25	

[†] For PCI and miscellaneous terminals, $V_I = V_{CCP}$. For PC Card terminals, $V_I = V_{CCA}$.

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

		•				
	PARAMETER	ALTERNATE SYMBOL	TEST CONDITIONS	MIN	MAX	UNIT
t _C	Cycle time, PCLK	t _{cyc}		30		ns
t _{w(H)}	Pulse duration (width), PCLK high	^t high		11		ns
t _W (L)	Pulse duration (width), PCLK low	t _{low}		11		ns
t _r , t _f	Slew rate, PCLK	Δν/Δt		1	4	V/ns
t _W	Pulse duration (width), GRST	t _{rst}		1		ms
t _{su}	Setup time, PCLK active at end of PRST	trst-clk		100		μs

[‡] For I/O terminals, input leakage (I_{IL} and I_{IH}) includes I_{OZ} leakage of the disabled output. § Miscellaneous terminals are: A9, B9, C9, G2, G3, J5, K5, P12, P17 (CLOCK, DATA, LATCH, SCL, SDA, SUSPEND, GRST, TEST0, TEST4

 $[\]P\operatorname{MFUNC}(0:6)$ share the same specifications as the PCI terminals.

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

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

	PARAMETER		ALTERNATE SYMBOL	TEST CONDITIONS	MIN	MAX	UNIT
	Desperation delegations Con Note 4	PCLK-to-shared signal valid delay time	^t val	C _L = 50 pF,		11	
^t pd	Propagation delay time, See Note 4	PCLK-to-shared signal invalid delay time	t _{inv}	See Note 4	2		ns
t _{en}	Enable time, high impedance-to-active	delay time from PCLK	ton		2		ns
tdis	t _{dis} Disable time, active-to-high impedance delay time from PCLK					28	ns
t _{su}	0.4 % 1.4 BOLK EL				7		ns
th	Hold time after PCLK high		t _h		0	·	ns

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

7.6 Reset Timing

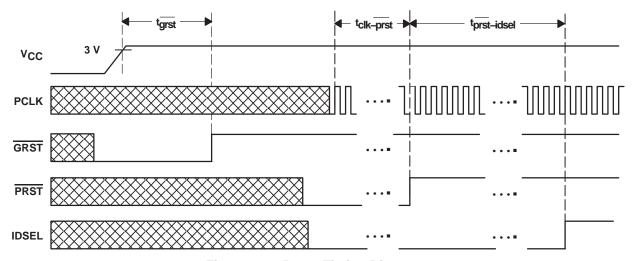


Figure 7–1. Reset Timing Diagram

	PARAMETER	MIN	MAX	UNIT
tgrst	$V_{CC} \ge 3.0 \text{ V to } \overline{\text{GRST}} \uparrow$	2		ms
t _{clk-prst}	PCLK ↑ to PRST ↑	100		μs
t _{prst-idsel}	PRST ↑ to IDSEL ↑	3		μs

NOTES: 5. GRST may be asynchronously deasserted, that is, it does not require a valid PCLK.

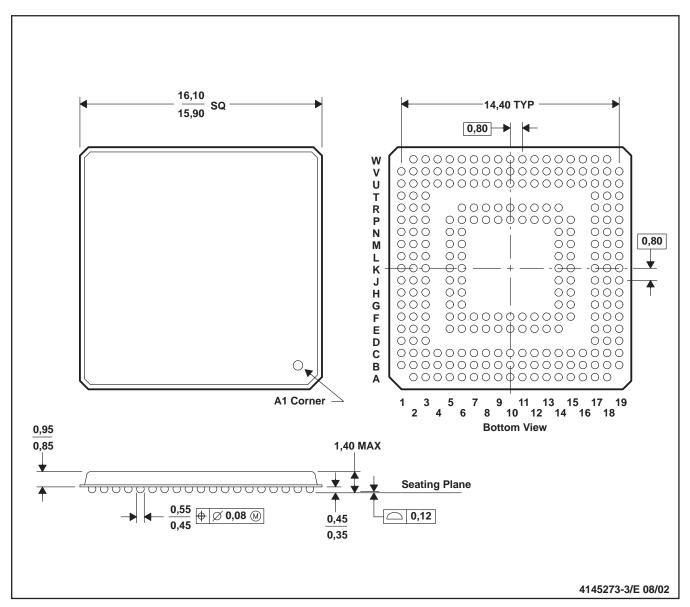
^{6.} There is no specific timing relationship of GRST to PRST. However, if GRST is deasserted after PRST then the PCLK to PRST ↑ and PRST ↑ to IDSEL ↑ apply to GRST.

8 Mechanical Information

The PCI1515 device is available in the 257-terminal MicroStar BGA™ package (GHK) or the 257-terminal lead (Pb atomic number 82) free MicroStar BGA™ package (ZHK). The following figure shows the mechanical dimensions for the GHK package. The GHK and ZHK packages are mechanically identical; therefore, only the GHK mechanical drawing is shown.

GHK (S-PBGA-N257)

PLASTIC BALL GRID ARRAY



NOTES: A. All linear dimensions are in millimeters.

B. This drawing is subject to change without notice

C. MicroStar BGA™ configuration

MicroStar BGA is a trademark of Texas Instruments.

Contact Us Buy About TI TI Worldwide my.TI

Products Applications Support Keyword

Part Number

TI Home > Semiconductors > Analog & Mixed-Signal > Interface > Connectivity > PCI > PCI CardBus Controllers >

PCI1515, Status: ACTIVE

Integrated 1 slot PC Card controller, pin compatible with PCI4515, PCI6515 & PCI7515

Add to my.TI

▼ Features	Samples	Technical Documents
▼ Quality & Pb-Free Data	▼ Pricing/Packaging	▼ Applications Notes
▼ Related Products	■ Inventory	Simulation Models
▼ Development Tools	Symbols/Footprints	Reference Designs

Datasheet



$\textbf{PCI1515 Single Socked CardBus Controller} \ (pci1515.pdf,\ 515\ KB)$

28 Jul 2004 Download

	PCI1515
Supply Voltage (Vcc) (Nom) (V)	3.3
1394 Ports	1
Pin/Package	257BGA
Approx. 1KU Price (US\$)	4.6
	Samples
	Inventory

Product Information

Features

- Save this to your personal library Add to my.TI
- PC Card Standard 8.1 compliant
- PCI Bus Power Management Interface Specification 1.1 compliant
- Advanced Configuration and Power Interface (ACPI) Specification 2.0 compliant
- PCI Local Bus Specification Revision 2.3 compliant
- PC 98/99 and PC2001 compliant
- Windows Logo Program 2.0 compliant
- PCI Bus Interface Specification for PCI-to-CardBus Bridges
- 1.5-V core logic and 3.3-V I/O cells with internal voltage regulator to generate 1.5-V core V_{CC}
- Universal PCI interfaces compatible with 3.3-V and 5-V PCI signaling environments
- Supports PC Card or CardBus with hot insertion and removal
- Supports 132-MBps burst transfers to maximize data throughput on both the PCI bus and the CardBus
- Supports serialized IRQ with PCI interrupts
- Programmable multifunction terminals
- Many interrupt modes supported
- Serial ROM interface for loading subsystem ID and subsystem vendor ID
- ExCA-compatible registers are mapped in memory or I/O space
- Intel 82365SL-DF register compatible
- Supports ring indicate, SUSPEND\, and PCI CLKRUN\ protocols and PCI bus Lock (LOCK)
- Provides VGA/palette memory and I/O, and subtractive decoding options, LED activity terminals
- Compliant with Intel Mobile Power Guideline 2000
- PCI power-management D0, D1, D2, and D3 power states
- · Advanced submicron, low-power CMOS technology

Other trademarks are the property of their respective owners.

Intel is a trademark of Intel Corporation.
TI and MicroStar BGA are trademarks of Texas Instruments.

• Advanced Submicron, low-power

Selection GuidesAnalog & Mixed-Signal: PCI

CardBus Controllers

Refine Your Selection

Support

- KnowledgeBase
- Contact Technical Support
- TI Cross Reference
- Training
- Part Marking Lookup

The PCI1515 controller is a single-function PCI controller compliant with PCI Local Bus Specification, Revision 2.3.

Function 0 provides a PC Card socket controller compliant with the PC Card Standard (Release 8.1). The PCI1515 controller provides features that make it the best choice for bridging between the PCI bus and PC Cards, and supports 16-bit, CardBus, or USB custom card interface PC Cards, powered at 5 V or 3.3 V, as required.

All card signals are internally buffered to allow hot insertion and removal without external buffering. The PCI1515 controller is register compatible with the Intel 82365SL-DF ExCA controller. The PCI1515 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 PCI1515 controller can be programmed to accept posted writes to improve bus utilization.

Pricing/Packaging/Samples							
		Price Packaging		Samples			
Device	Status	Temp (°C)	Budget Price (\$US) QTY	Package Type Pins	STD Pack QTY	Samples	
PCI1515GHK	ACTIVE	0 to 70	4.60 1KU	BGA (GHK) 257	90	Request Samples	

Inventory								
	TI Inventory Status		Reported Distributor Inventory			١		
PCI1515GHK		As of 10:23 AM	GMT,	16 Dec 2004		As of 10:23 AM G	MT, 16 Dec 2004	
	In Stock	In Progress QTY	Date	Lead Time	In Stock	Distributor: Region Company	y Purchase	
	1770*			Call**	90	Americas DigiKey	Buy Now	

View all Distributors

information.

Quality & Lead (Pb)-Free Data					
	Product Content MTE				
Device	Eco Plan*	Lead/Ball Finish	MSL Rating/Peak Reflow	Details	Details
PCI1515GHK	None	Call TI	Level-3-220C-168 HR	View	View

May not be currently available - please click on the Product Content Details "View" link in the table above for the latest availability information and additional product content details.

If the information you are requesting is not available online at this time, contact one of our Product Information Centers regarding the availability of this information.

* TI Recommends

Technical Documents	
Datasheets	Keep track of what's new Add to my.TI
PCI1515 Single Socked CardBus Controller (pci1515.pdf,515 KB)	
28 Jul 2004 Download	
Application Notes	
View Application Notes for PCI CardBus Controllers	
⚠ More Literature	
View More Literature for PCI CardBus Controllers	

Related Products			
⚠ Related Tools			
Development Tools	Part #	Company	Tool Type
Connectivity Developer Network Solutions	THIRD-PARTY-TOOLS	Texas Instruments	Third Party Development Environments

^{*} Our information is updated daily, so please check back with us soon if ** Lead time information is not available at this time. However, our this does not meet your needs. You may also contact your TI Authorized information is updated daily so please check back with us soon. Please Distributor, including those listed above, for real time stock information. contact your preferred TI Authorized Distributor for additional

Products | Applications | Support | Site Map

 $@\ Copyright\ 1995-2004\ Texas\ Instruments\ Incorporated.\ All\ rights\ reserved.\ Trademarks\ \mid\ Privacy\ Policy\ \mid\ Terms\ of\ Use$