PCI1510 Implementation Guide

Computer Connectivity Solutions

ABSTRACT

This document is provided to assist platform designers using the PCI1510 single-socket PC Card controller. Detailed information can be found in the PCI1510 data manual. However, this document provides design suggestions for the various options when designing in the PCI1510.

Contents

Figures

Document History

Revised by	Date	Document Name	Revision Comments
DGB	12/02/02	PCI1510 Implementation Guide – 0.90.doc	*Initial Draft
DGB	01/07/03	PCI1510 Implementation Guide – 1.00.doc	*Corrected typos

1 PCI1510 Typical System Implementation

The figure below represents a typical implementation of the PCI1510 PC Card Controller. The device serves as a bridge between a PCI Bus and a PC Card interface. The PCI1510 will operate only with the PCI Bus as a primary bus and the PC Card interface as the secondary bus. The PC Card interface operates with both CardBus (32-bit) and 16-bit PC Cards.

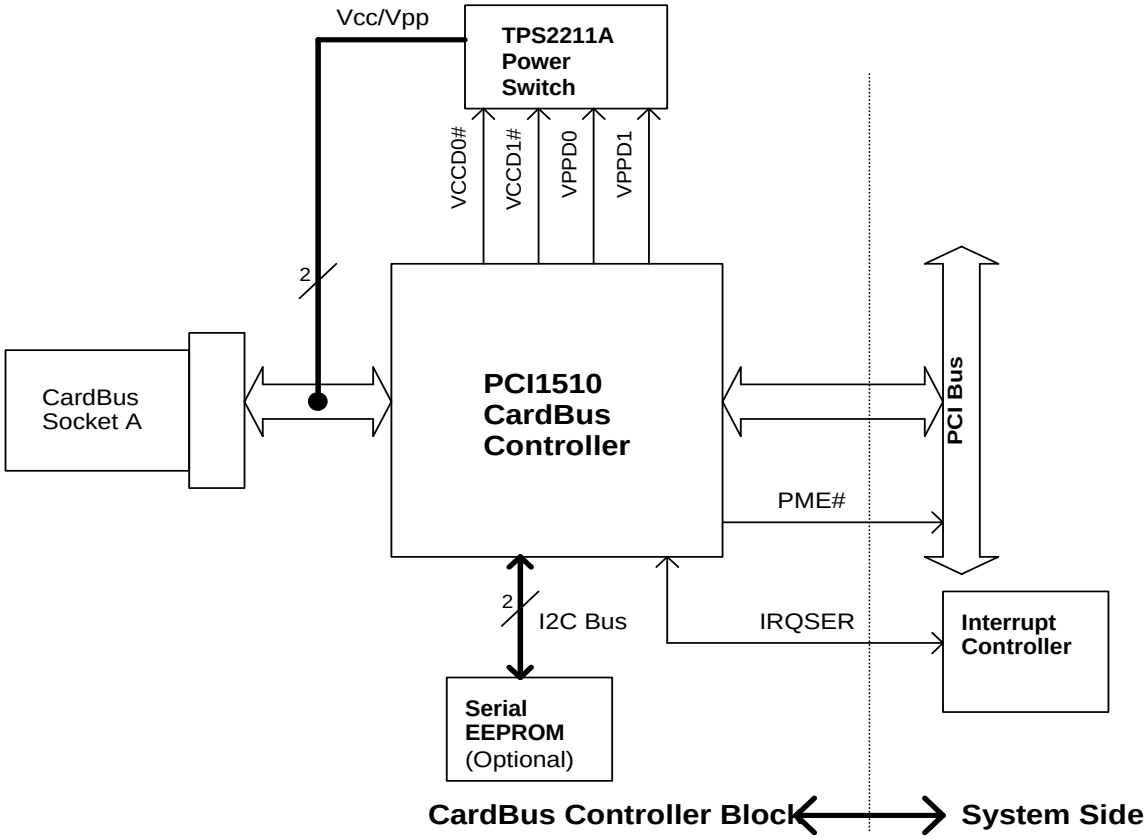

Figure 1. Typical System Implementation

A power switch is necessary in order to control power to the PC Card socket. The recommended power switch is the TPS2211A.

The EEPROM can be used to set various configuration registers but is not necessary if those registers are settable via software/BIOS for the system.

IRQSER is used to pass both PCI interrupts and ISA style legacy interrupts to the system. Only PCI interrupts are necessary in order for CardBus cards to operate correctly. Some 16-bit PC Cards require ISA style legacy interrupts in order to function properly in some systems.

2 Power Considerations

2.1 Internal Voltage Regulator

One of the major differences between the PCI1510 and previous Texas Instruments CardBus controllers is that the PCI1510 uses an internal voltage regulator to power the core logic at 2.5V. This allows for a more than 50% reduction in power consumption over previous controllers. The voltage regulator is enabled using the VR_EN# pin. If VR_EN# is high, the voltage regulator is disabled and VRPORT serves as a 2.5V external input to power the core. If VR_EN# is low, the voltage regulator is enabled and VRPORT serves as a 2.5V output. This 2.5V output cannot be used to power other devices and is only available externally in order to provide a 1µF bypass capacitor. VRPORT must have a 1µF bypass capacitor to ground in order for proper operation if the voltage regulator is enabled.

2.2 Clamping Rails

The PCI1510 has 2 clamping rails: VCCP and VCCCB. VCCP is the PCI interface I/O clamp rail and can be either 3.3V or 5V depending on the system implementation. The PCI1510 will only signal on the PCI bus at 3.3V but is 5V tolerant. VCCCB are connected to the PC Card power rail for the PC Card socket. This terminal serves as the clamping input for the PC Card interface to the PCI1510.

2.3 Bypass Capacitors

Standard design rules for power supply bypass should be followed. A value of 0.1µF is recommended for each of the power pins VCC, VCCP, and VCCCB.

3 Power Switch Implementation

The following figure shows the parallel interface between the PCI1510 and the TPS2211A power switch:

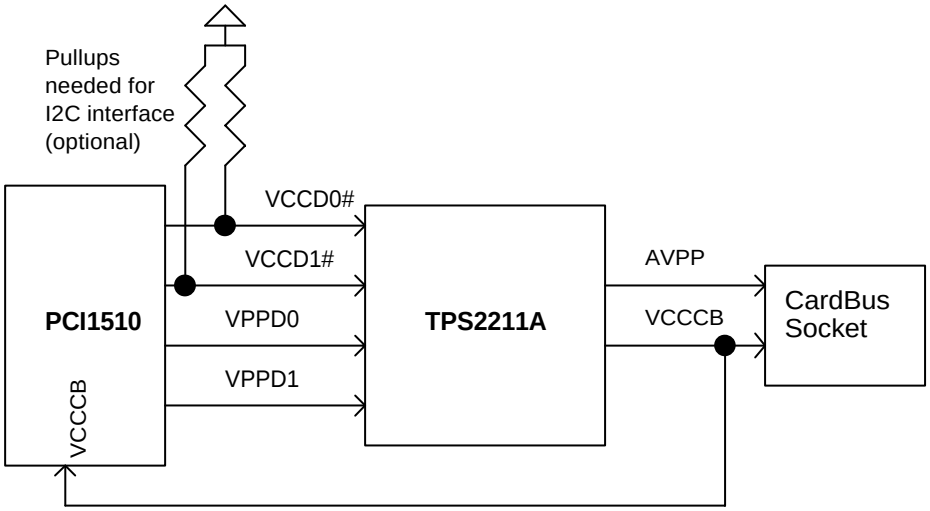

Figure 2. Power Switch Implementation

A power switch is necessary in order to control power to the PC Card socket. When the PCI1510 receives a socket power request, it sends the appropriate data across the parallel interface (VCCDX# and VPPDX). In turn, the power switch turns on the appropriate levels for VCC and VPP for that socket. 43kΩ pullups on VCCD0# and VCCD1# are used to indicate to the PCI1510 that an EEPROM is being used to initialize certain registers in the PCI1510.

4 PCI Bus Interface

The PCI1510 has a 33MHz, 32 bit PCI Interface compliant with PCI Local Bus Specification Revision 2.2.

- PCLK, PRST#, AD31:0, C/BE#3:0, PAR, DEVSEL#, FRAME#, STOP#, TRDY#, IRDY#, GNT#, and REQ# are required PCI signals. All except PCLK, GNT#, and REQ# are bussed signals. PCLK is a 33MHz point-to-point clock. GNT# and REQ# are point-to-point signals from the PCI bus arbitrator.
- PERR#, SERR#, and LOCK# are optional PCI signals. PERR# and SERR# are bussed signals and should be pulled up to VCC if unused. LOCK# is available on a Multifunction Terminal. If LOCK# is not needed for system implementation, it should not be configured as such in the Multifunction Routing register (PCI configuration offset 8Ch).
- GRST# (Global reset) and PRST# (PCI reset) are both used to initialize the PCI1510. The assertion of GRST# puts the PCI1510 in its default state. The assertion of PRST# does not initialize GRST# only bits. PRST# also does not initialize PME# context bits if PME# in enabled. More information can be found in Section 9.1 – D3 Wake Information.
- IDSEL should be resistively coupled (100Ω) to one of the address lines between AD31 and AD11. Please refer to Section 3.2.2.3.5 (System Generation of IDSEL) and Section 4.2.6, footnote 31 (Pinout Recommendation) of the PCI Local Bus Specification Revision 2.2 for more information.
- PCI Interrupts can be routed through INTA# through the Multifunction terminals. More information can be found in Section 7 – Interrupt Configurations.
- PCI CLKRUN# can be routed through Multifunction terminal 6. For more information, please refer to Section 9 – Power Management Considerations.
- PME# is used to signal Power Management Events. This signal is important for waking the PCI1510 from low power states. PME# is an open-drain signal.
- **Pullup resistors** are needed on the following PCI terminals: IRDY#, TRDY#, FRAME#, STOP#, DEVSEL#, PERR#, SERR#, LOCK#, INTA#, INTB#, CLKRUN#, and PME#. GRST# requires a pullup if the controlling output does not drive high.

5 PC Card Interface

There are two different modes on the PC Card interface. The first is 16-bit mode which is analogous to the legacy ISA bus. The second is 32-bit CardBus mode which is very similar to a PCI Bus. The terminal functions for these two modes are multiplexed and routed to the PC Card socket. The following suggestions apply to the PC Card interface:

- **Pullup resistors** for the PC Card interface have been integrated into the PCI1510. These include: A14/CPERR#, A15/CIRDY#, A19/CBLOCK#, A20/CSTOP#, A21/CDEVSEL#, A22/CTRDY#, BVD2(SPKR#)/CAUDIO, CD1#/CCD1#, CD2#/CCD2#, INPACK#/CREQ#, READY/CINT#, RESET/CRST#, VS1#/CVS1, VS2#/CVS2, WAIT#/CSERR#, WP(IOIS16#)/CCLKRUN#.
- A switchable pullup/pulldown resistor has been implemented on BVD1(STSCHG#)/CSTSCHG. The pulldown is implemented when a CardBus card is being used or when the socket is empty. A pullup is implemented when a 16-bit PC card is being used.
- A damping resistor is necessary on the CCLK terminal between the PCI1510 and the PC Card socket. The value is system dependent. If line impedance is in the range of 60-90Ω, a 47Ω resistor is recommended. For more information, please see the PC Card Standard Revision 7.2, Section 5.3.2.1.4.
- CD# line noise filtering is no longer required because the PCI1510 has an integrated digital noise filter.
- Three PC Card terminals on the socket are not necessary for CardBus mode but are necessary for 16-bit mode. These terminals are: CRSVD/D14, CRSVD/A18, and CRSVD/D2. These terminals must be connected to the PC Card Socket according to their 16-bit designations. By default, when in CardBus mode, these terminals are driven low. They can be tristated by setting bit 22 (CBRSVD) in the System Control register at PCI configuration offset 80h.

6 Miscellaneous Pin Interface

6.1 Multifunction Terminals

The multifunction terminals (MFUNC6:0) can be programmed to serve many different roles using the Multifunction Routing register at PCI configuration offset 8Ch. The discrete ISA interrupts (IRQ15:2), INTA#, and IRQSER are explained in Section 7 – Interrupt Configurations. CLKRUN#, D3STAT#, and RI_OUT# are discussed in Section 9 – Power Management Considerations. ZVSTAT and ZVSEL0# are used for ZV control. For more information, please refer to the PCI1510 Data Manual.

LED_SKT can be used to indicate socket activity. When a PC Card is being accessed, LED_SKT will be driven high for access to the socket.

GPE#, GPIx, and GPOx can be used to signal general purpose events to the system.

CAUDPWM provides a PWM output for the CAUDIO terminals (as opposed to the binary output SPKROUT).

PCI LOCK# is an optional PCI signal as mentioned in Section 4 – PCI Bus Interface.

All multifunction terminals require a 43kΩ pullup resistor because on initial powerup, they default to inputs.

6.2 SPKROUT

SPKROUT is the output to the host system that can carry SPKR# or CAUDIO through the PCI1510 from the PC Card interface. A 43k pulldown resistor is required to prevent oscillation when SPKROUT is disabled and therefore tristated.

6.3 SUSPEND#

The assertion of SUSPEND# gates PRST#, GRST#, and PCLK from the PCI1510. More information can be found in Section 9 – Power Management Considerations. A 43kΩ pullup resistor is required on SUSPEND#. SUSPEND# cannot be low during boot.

7 Interrupt Configurations

The PCI1510 provides system designers with great flexibility in configuring interrupts. The PCI1510 allows four interrupt modes which are selected via bits 2:1 of the Device Control register at PCI offset 92h.

PCI interrupts are available on INTA#. This signal is available on MFUNC0. The Multifunction Routing register at PCI configuration offset 8Ch must be programmed correspondingly. PCI interrupts can also be signaled through IRQSER.

ISA style IRQ interrupts are available on IRQ15:2. These signals are available on MFUNC6:0. These interrupts are necessary for some 16-bit PC Cards to function properly in some systems. IRQ interrupts can also be signaled through IRQSER. IRQSER is available on MFUNC3 and requires a 43k pullup resistor to VCC.

Windows XP has simplified interrupt routing by allowing many 16-bit cards which used ISA IRQs in previous OSs to use PCI interrupts. More information on interrupts in Windows XP can be found at: http://www.microsoft.com/hwdev/bus/cardbus/PCMCIA-IRQrouting.asp

7.1 Parallel PCI Interrupts Only

The parallel PCI interrupts only mode is selected by programming bits 2:1 to a value of 00b. This allows interrupts to be routed through INTA#. This is not a recommended interrupt configuration because some 16-bit PC Cards require legacy ISA interrupts and will not function properly.

7.2 Parallel IRQ and Parallel PCI Interrupts

The parallel IRQ and parallel PCI interrupts mode is selected by programming bits 2:1 to a value of 01b. This allows interrupts to be routed through IRQ15:2 and INTA#. This is not a recommended interrupt configuration because this requires all the multifunction terminals to be used as interrupts which limits other functions on the PCI1510.

7.3 Serial IRQ and Parallel PCI Interrupts

The serial IRQ and parallel PCI interrupts mode is selected by programming bits 2:1 a value of 10b. This allows interrupts to be routed through IRQSER and INTA#. This is the recommended interrupt configuration for a PCI add-in card implementation of the PCI1510. INTA# can be routed through the PCI edge connector while IRQSER must be attached to a Serial IRQ input on the motherboard. If no Serial IRQ input is available, this mode still allows CardBus cards to function properly. However, some 16-bit cards may not.

7.4 Serial IRQ and Serial PCI Interrupts

The serial IRQ and serial PCI interrupts mode is selected by programming bits 2:1 to a value of 11b. This allows all interrupts to be routed through IRQSER. This is the recommended interrupt configuration for all designs other than PCI add-in cards if an IRQSER input is available in the system. It is the simplest method of routing interrupts and allows the other multifunction terminals to be used for other purposes.

8 Software Considerations

The PCI1510 is natively supported by Windows XP. The PCI1510 will be recognized natively as a Generic CardBus Controller under Windows 2000, Windows ME, and Windows 98SE. The device will function properly using this designation. However, it is recommended that new .inf files provided by Texas Instruments be used for non-XP systems. These .inf files have a few small tweaks and allow the device to be reported in Device Manager properly.

Other operating systems are not supported directly by Texas Instruments. However, many non-Microsoft operating systems have generic CardBus device drivers which are compatible with the PCI1510. Any driver which was compatible with a previous Texas Instruments CardBus controller (such as the PCI1211, PCI1410, PCI1410A) or the Intel 82365SL should also be compatible with the PCI1510.

8.1 EEPROM Configuration

The following diagram represents the implementation of an EEPROM for the PCI1510 for configuration:

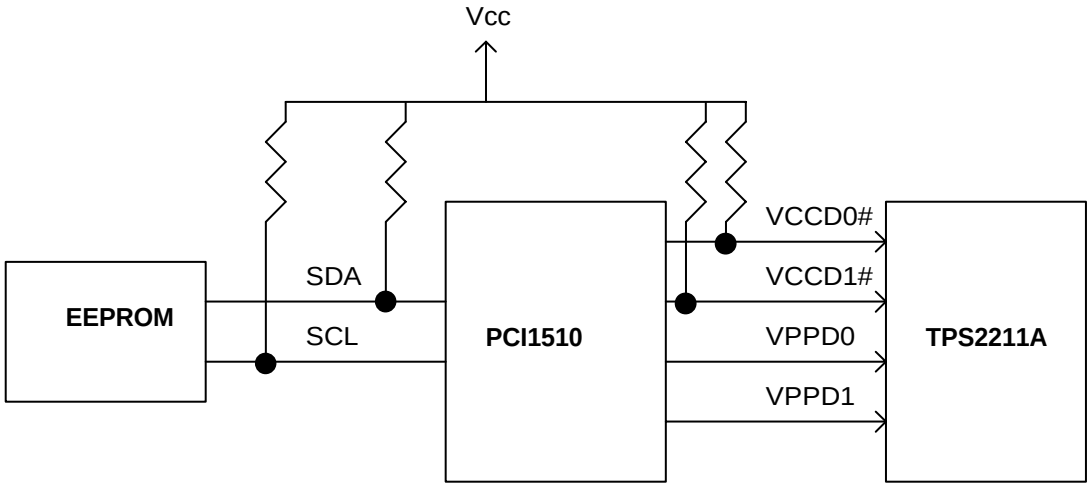

Figure 3. EEPROM Implementation

On the rising edge of GRST#, if VCCD0# and VCCD1# are high, the Serial Bus Detect bit (bit 3, PCI offset B3h) is set and the EEPROM contents are loaded into the PCI1510. MFUNC1 and MFUNC4 become SDA and SCL respectively. In order for the PCI1510 to detect the EEPROM and load configuration information, pullup resistors must be implemented on VCCD0# and VCCD1#. Pullups are needed on SDA and SCL. The EEPROM slave address should be 1010000b. If the Serial Bus Detect bit is cleared after the EEPROM data is loaded, MFUNC1 and MFUNC4 are returned to their functions as indicated by the Multifunction Routing Register (PCI offset 8Ch).

The EEPROM loading map can be found in the data manual. The following is an example data file which could be loaded into the EEPROM for use with the PCI1510:

```
; EEPROM Programming Data for the PCI1510 EVM
; Configured for IRQ serialized interrupts and parallel PCI interrupts

; Register     Data     Description

     00        0x01     ;FLAG
     01        0x00     ;04h Command Register, bit 8 (mapped from EEPROM bit 7), 6-5, 2-0
     02        0x78     ;40h Sub-System Vendor ID Byte 0
     03        0x56     ;40h Sub-System Vendor ID Byte 1
     04        0x34     ;42h Sub-System ID Byte 0
     05        0x12     ;42h Sub-System ID Byte 1
     06        0xe0     ;44h Legacy Bar Byte 0, bits 7-1
     07        0x03     ;44h Legacy Bar Byte 1
     08        0x00     ;44h Legacy Bar Byte 2
     09        0x00     ;44h Legacy Bar Byte 3
     0a        0x60     ;80h System Control Byte 0 (default)
     0b        0xd0     ;80h System Control Byte 1 (MRBURSTU=1 all others default)
     0c        0x44     ;80h System Control Byte 2 (default)
     0d        0x08     ;80h System Control Byte 3 (default)
     0e        0x02     ;8ch MFUNC Byte 0
     0f        0x1c     ;8ch MFUNC Byte 1
     10        0xc0     ;8ch MFUNC Byte 2
     11        0x01     ;8ch MFUNC Byte 3
     12        0xc0     ;90h Retry Status bits 7, 6 (PCI Retry, CardBus Retry)
     13        0x00     ;91h Card Control bits 7, 5 (Ring Indicate Enable, ZV Port Select)
     14        0x44     ;92h Dev Cntr bits 6,3-0 (3V Capable, IRQ serial and parallel PCI)
     15        0x00     ;93h Diagnostic bits 7, 4-0
     16        0x00     ;a2h Pwr Management Capabilities bit 15 (PME_Support from D3cold=0)
     17        0x84     ;00h ExCA ID and Revision bits 7-0
     18        0x00     ;Och+CB Socket Force Event Function 0 bit 27 (ZVSUPPORT=0)
```

8.2 BIOS Considerations

This section provides a high-level overview of the registers which need to be programmed by the BIOS upon initialization. In general, the only registers which must be programmed for proper operation within a Windows operating system are those registers which are EEPROM loadable. Other registers may need to be changed according to system implementation. The following sections contain explanations of registers which are frequently asked about. Microsoft provides the following reference documents concerning initialization of CardBus controllers in Windows:

http://www.microsoft.com/hwdev/bus/cardbus/cardbus1.asp

http://www.microsoft.com/hwdev/bus/pci/pcibridge-cardbus.asp

8.2.1 PCI Configuration Registers (Standard)

Cache Line Size Register (PCI offset 0Ch) – This register indicates the size in doublewords of a cache line. This register is system architecture dependent.

Latency Timer Register (PCI offset 0Dh) – This register indicates the number of PCI clocks the PCI1510 will be allowed access to the PCI bus if another master has its REQ# asserted. The recommended value is 40h. However, the value should be dependent on the system implementation and which devices need priority.

CardBus Latency Timer Register (PCI offset 1Bh) – This register indicates the number of CardBus clocks the PCI1510 will be allowed access on the CardBus interface. Because the CardBus interface is a point-to-point interface, the PCI1510 does not deassert CGNT# until a transaction is finished. Therefore, this register has little effect on the system.

Subsystem Vendor ID and Subsystem ID Registers (PCI offsets 40h and 42h) – These registers are used for subsystem and option card identification purposes. Typically, these registers contain the OEM vendor ID and an OEM identified designator. These fields can be programmed using the EEPROM or BIOS. If using BIOS, the SUBSYSRW bit (System Control register, bit 5) must be cleared to 0. The SSVID and SSID registers can now be written. The SUBSYSRW bit should be set to 1 after the registers are written.

8.2.2 PCI Configuration Registers (TI Extension)

System Control Register (PCI offset 80h) – This register contains many important system dependent variables. Please refer to the datasheet for more details. Of possible interest to the BIOS programmer: SER_STEP, INTRTIE, P2CCLK, MRBURSTDN, MRBURSTUP, and RIMUX.

Multifunction Routing Register (PCI offset 8Ch) – This register controls the seven multifunction terminals of the PCI1510. This register must be set before the interrupt mode is programmed in the Device Control register (PCI offset 92h).

Card Control Register (PCI offset 91h) – This register contains enable bits for RI_OUT# and SPKROUT.

Device Control Register (PCI offset 92h) – This register contains the interrupt mode bits.

Power Management Capabilities Register (PCI offset A2h) – This register is important for systems needing to wake from the D3 power state. Bit 15 reflects whether or not PME# is supported from D3cold. Bit 4 is tied to bit 15 indicating that if PME# is supported from D3cold, the system must be providing auxiliary power.

Power Management Control and Status Register (PCI offset A4h) – This register contains the PME# enable bit (bit 8).

8.2.3 ExCA Compatibility Registers

ExCA Interrupt and General-Control Register (ExCA offset 03/43h) – This register is used to route CSTSCHG interrupts via PCI interrupts.

8.2.4 CardBus Socket Registers

Socket Control Register and Socket Power Management Register (CB offsets 10h and 20h) – These registers can be used to characterize how CB CLKRUN# functions.

9 Power Management Considerations

9.1 D3 Wake Information

A power management event (PME) is the process by which a PCI or CardBus function can request a change of its current power consumption state. Typically, a device uses PME# to request a change from a power savings state to the fully operational state, D0. PME Context is defined as the functional state information and logic required to generate PMEs, report PME status, and enable PMEs. PCI Function Context refers to the small amounts of information held internal to the function. This includes not only the contents of the function's PCI registers, but also information about the operation states of the function including state machine context and other internal mechanisms.

When global reset (GRST#) is asserted, the PCI1510 is completely non-functional and is in a default state. Output buffers are tristated and internal registers are reset. The result of PCI reset (PRST#) being asserted is dependent on whether PME# is enabled or not. When PRST# is asserted with the function not enabled for PME#, it causes the PCI1510 to tristate all output buffers and reset all internal registers except for those considered 'GRST# Only Registers'. If PME# is enabled for the socket, the PCI1510 will maintain its 'PME# Context Registers'.

According to the PCI Bus Power Management Interface Specification for PCI to CardBus Bridges, a device returning to D0 from D3hot is required to assert an internal reset. The PCI reset may or may not be asserted by the system. However, for a device returning to D0 from D3cold however, PRST# must be asserted by the system.

For a wake from D3cold, the device needs to save its PME# context in order for software to determine the source of the wake-up event. This is accomplished using PME# enable and saving the PME# context registers. However, the device must also maintain certain registers that are normally configured by BIOS at boot time. This is accomplished using GRST# and the 'GRST# Only Registers.' This allows a system to be in a low power state and resumed quickly without needing BIOS to reprogram the device.

The sequence of events at power up are that GRST# and PRST# should be asserted. 100 µs after PCLK is stable, GRST# can be deasserted. PRST# can be deasserted at the same time as GRST# or any time there after. At this point, GRST# will stay deasserted until the system completely cycles power and reboots. Now the system can put the PCI1510 into a lower power state and may or may not assert PRST#.

The PCI1510 does not require a PCI clock to generate a PME# signal. However, it does require a voltage source such as Vaux to be supplied and the pullup on PME# must also be connected to Vaux. In addition, the VCCP pins and power switch must also have power in order to wake from a card. Vaux is limited to 200mA for the socket.

For systems not implementing wake from D3, GRST# can be tied to PRST#.

9.1.1 GRST# Only Registers

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

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

9.1.2 PME# Context Registers

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

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

9.2 PME#/RI_OUT# Behavior

PME# and RI_OUT# are very important for power management. The PME# signal is useful for PCI power management systems. The RI_OUT# (Ring Indicate Out) signal is used for legacy power management systems. PME# and RI_OUT# are multiplexed on the same pin. The PCI1510 can also provide RI_OUT# on the Multifunction terminals.

To enable passage of Ring signals from the PC Card interface, RINGEN (bit 7 ExCA offset 803) must be set to ‘1’, and RIENB (bit 7 PCI offset 91h) must be set to ‘1’.

9.3 CLKRUN# Protocol

CLKRUN# is a hardware method of clock control that can be used in parallel with other types of power management. For the PCI1510, PCI CLKRUN# can be programmed using the Multifunction Routing Register (PCI offset 8Ch) on MFUNC6. CardBus CLKRUN# is a required signal incorporated into the PC Card interface. The following bits can be used to adjust the operation of how PCI and CB CLKRUN# affect the PCI1510:

Multifunction Routing register – MFUNC6 (PCI offset 8Ch, bits 27-24 set to 0001b). Requires a 43kΩ pullup.

KEEPCLK – System Control Register (PCI offset 80h, bit 1). Setting this bit to a ‘1’ will never allow the PCI CLKRUN# protocol to stop or slow the PCI clock.

STOPCLK – Socket Control Register (CB offset 10h, bit 7). This bit determines whether the CB CLKRUN# protocol is affected by the PCI CLKRUN# protocol.

CLKCTRLEN – Socket Power Management Register (CB offset 20h, bit 16). This bit enables the CB CLKRUN# protocol.

CLKCTRL – Socket Power Management Register (CB offset 20h, bit 0). This bit determines whether the CB CLKRUN# protocol will either stop or slow CCLK.

9.4 SUSPEND#

The assertion of the SUSPEND# signal gates PCLK, GRST#, PRST# from the PCI1510. The recommended implementation for SUSPEND# is to not use it for power management and simply connect a 43kΩ pullup resistor. SUSPEND# is an unstandardized method of power management and causes many implementation problems. The following guidelines are provided to help reduce implementation issues.

The main purpose of the PCI1510 SUSPEND# pin is to prevent PCI reset from clearing all register context which would require the reconfiguration of the PCI1510 by software. Asserting the PCI1510 SUSPEND# signal will also tri-state the controllers PCI outputs and gate the PCLK internally to the controller. Due to the tri-stated PCI outputs, it is important that the PCI bus not be parked on the PCI1510 when SUSPEND# is asserted.

Another major point to note is that powerdown of a card slot due to card removal requires the use of either the Internal Oscillator or an externally supplied clock to the power switch. If an external clock is used and is removed during Suspend, the card slot will not power down and will remain powered. This opens the possibility of potential card damage. If a 3.3V card is inserted into the hot slot that was powered to 5V, card damage will most likely occur. It is therefore recommended that OSEN, bit 27 at PCI offset 80h is set to a '1' so that the Internal Oscillator is enabled.

10 Pin Compatibility with Other Devices

The PCI1510 can be designed on to the same PCB as other Texas Instruments CardBus controllers such as the dual socket PCI1520 controller even though the two devices are not pin compatible. This can be done using a dual footprint for the devices on the PCB. For example, a designer may want the option of having a single or dual socket implementation on a single PCB. In this instance, a PCI1510 BGA (GGU) footprint can be placed inside a PCI1520 QFP (PDV) footprint. The traces for the PC Card socket A on the PCI1520 footprint are then connected to the PC Card socket traces on the PCI1510 footprint. For single socket implementations, only one PC Card socket is populated along with the PCI1510 controller. For dual socket implementation, both PC Card sockets are populated along with the PCI1520 controller.

11 Migration to the PCI1510 from the PCI1410A or PCI1410

The major differences between the PCI1510 and PCI1410A are pinout, lower power consumption, and lower cost. The pinout is changed on the PCI1510 in order to incorporate an internal voltage regulator which allows the core to operate at 2.5V. The only difference between the PCI1410 and PCI1410A is an erratum fix in the PCI1410A. Please see the PCI1410 and PCI1410A errata sheets for more details. When moving from the PCI1211 to the PCI1510, please see Section 12 for the differences between the PCI1211 and PCI1410 in addition to the changes from this section.

11.1 Hardware and Pin Assignment Changes

- The pinout on the PCI1510 is significantly changed from the PCI1410A. This requires a PCB redesign.
- A low dropout voltage regulator is integrated into the PCI1510 to supply 2.5V core voltage. A voltage regulator enable pin (VR_EN#) has been added in place of one of the VCCP pins. A core voltage input/output (VRPORT) pin has been added in place of the VCCI pin. This pin is used to either input core voltage or allow for an external 1.0µF bypass capacitor depending on the value of VR_EN#. A typical implementation would enable the regulator by grounding VR_EN# and adding the bypass capacitor from VRPORT to ground. For further details, see the datasheet.
- The PCI1510 does not have a VCCI pin. Signals clamped to VCCI on the PCI1410 are clamped to VCCP on the PCI1510.
- The PCI1510 has integrated pullup resistor on the CCLKRUN#//WP(IOIS16#) terminal. All necessary pullup resistors on the PC Card interface have been integrated in the PCI1510.
- A switchable pullup/pulldown resistor has been implemented on the CSTSCHG//BVD1(STSCHG#/RI#) terminal. The pullup is active when the 16BITCARD bit (bit 4 in the Socket Present State register) is ‘1’, otherwise the pulldown resistor is activated. This prevents unexpected PME# assertion.

11.2 Configuration Register Changes

- The device ID for the PCI1510 is AC56h.
- The default value of the Multifunction Routing register (PCI offset 8Ch) has been changed from 00000000h on the PCI1410A to 00001000h in order to enable IRQSER on MFUNC3 by default.
- Bit 0 in the Diagnostic register (PCI offset 93h) is no longer Asynchronous Interrupt Enable. The functionality is no longer necessary. It is now STDZVEN which enables the new ZV register model.
- Bits 2-0 in the Power Management Capabilities register (PCI offset A2h) are now '010b' indicating that the PCI1510 is compliant to Revision 1.1 of the PCI Bus Power Management Specification.
- Bit 4 (AUX_PWR) in the Power Management Capabilities register (PCI offset A2h) is now tied to bit 15 (PME#_Support for D3Cold).
- D3_STAT# functionality has been added to MFUNC5, MFUNC4, and MFUNC2. D3_STAT# is asserted when PME# is enabled and the device is placed in D3 power state.
- Bit 27 in the Socket Present State register (Socket offset 08h) now indicates Zoom Video Support in the socket for the PCI1510. It is reserved in the PCI1410.
- Bit 27 in the Socket Force Event register (Socket offset 0Ch) now causes the ZVSUPPORT bit mentioned above to be set in the PCI1510. It is reserved in the PCI1410.
- Bits 11-9 in the Socket Control register (Socket offset 10h) were reserved and now are used for ZV control.
- Registers and bits previously referring to centralized or distributed DMA are now reserved (bits 19-16 System Control register at PCI offset 80h, DMA registers at PCI offsets 94h and 98h) (see explanation about DMA below).
- The EEPROM loading map has changed significantly to provide more control for applications needing an EEPROM (see datasheet for details).
- PME# and GRST# only context registers have changed. Please refer to each datasheet for list of registers.

11.3 Other Functional Differences

- The PCI1510 is natively supported by Windows XP. The PCI1510 will be recognized natively as a Generic CardBus Controller under Windows 2000, Windows ME, and Windows 98SE. The device will function properly using this designation. However, it is recommended that new .inf files provided by Texas Instruments be used for non-XP systems. These .inf files have a few small tweaks and allow the device to be reported in Device Manager properly.

- The latest versions of the PC Card Standard (beginning with 7.2 in November 2000) no longer support centralized or distributed DMA for PC Cards. Therefore, the PCI1510 no longer supports centralized or distributed DMA. DMA was used by very few PC Cards, most of which are obsolete (DOS-based sound cards, DVD decoders).
- A new standardized ZV register model has been implemented in the PCI1510 (see datasheet for details). The PCI1510 is backward compatible with the legacy ZV register model used in previous CardBus controllers.
- The timing condition erratum which caused sometimes cause an EEPROM to not be detected has been fixed.
- SPKROUT# signal behavior is changed. The signal will stay low during socket power on an off. A pulldown resistor is required to prevent oscillation.
- The D3_STAT# erratum which caused D3_STAT# to be asserted when the bus goes to the B3 Power State has been corrected.
- The D3 to D0 transition erratum which caused some cards to not be recognized after system standby has been corrected.

12 Reference Schematics

Please refer to the PCI1510 EVM schematics for use as reference schematics.

13 References

1. *PCI1510 GGU/PGE PC Card Controllers Data Manual (SCPS071)*
2. *PCI Local Bus Specification Revision 2.2*
3. *PC Card Standard Revision 7.2*
4. *PCI Bus Power Management Interface Specification Revision 1.1*
5. *PCI Mobile Design Guide Revision 1.0*

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

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

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

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

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor, does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.