PCI7620/PCI7420/PCI6620/PCI6420 Implementation Guide

Computer Connectivity Solutions

ABSTRACT

This document is provided to assist platform designers using the PCIXX20 (PCI7620, PCI7420, PCI6620, or PCI6420) dual-socket CardBus and SmartCard controller with integrated 1394a and dedicated Secure Digital/Multimedia Card and Memory Stick/Memory Stick-Pro sockets. Detailed information can be found in the PCIXX20 data manual. However, this document provides design suggestions for the various options when designing in the PCIXX20 controller.

HOW TO USE THIS DOCUMENT

This document covers implementation guidance for all four PCIXX20 controllers. In order to efficiently utilize this document, each controller with its relevant sections is listed below:

Section	PCI7620	PCI7420	PCI6620	PCI6420
1	1.1	1.2	1.3	1.4
2	All	All	All	All
3	All	All	All	All
4	All	All	All	All
5	All	All	All	All
6	All	-	All	-
7	All	All	All	All
8	All	All	8.3	8.3
9	All	All	All	All
10	All	All	All	All
11	All	All	All	All
12	All	All	All	All
13	All	-	All	-

Contents

Tables

Figures

1 PCIXX20 Implementation Configuration

The PCIXX20 controllers include four different controllers to accommodate a diversity of system implementations.

- PCI7620 controller
- PCI7420 controller
- PCI6620 controller
- PCI6420 controller

1.1 PCI7620 Implementation Configuration

The PCI7620 (device ID 0xAC8Dh) controller supports five functions (CardBus, Smart Card, Secure Digital/Multimedia Card, Memory Stick/Memory Stick-Pro, and IEEE 1394a), all of which can be individually enabled or disabled. For more product information, please refer to the PCI7620 data manual at http://www.ti.com. Please refer to Table 1 for interface-related sections.

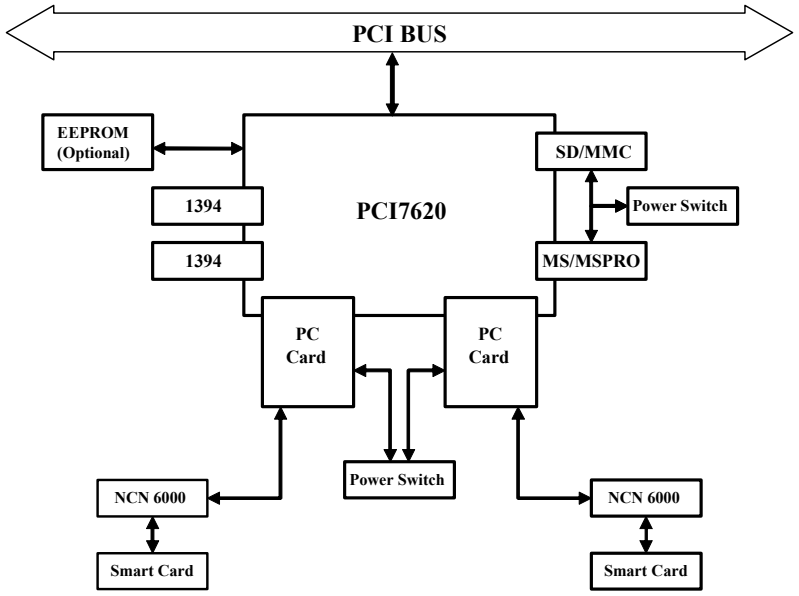


Figure 1. PCI7620 Typical System Implementation

1.2 PCI7420 Implementation Configuration

The PCI7420 (device ID 0xAC8Eh) controller supports four functions (CardBus, Secure Digital/Multimedia Card, Memory Stick/Memory Stick-Pro, and IEEE 1394a), all of which can be individually enabled or disabled. For more product information, please refer to the PCI7420 data manual at http://www.ti.com. Please refer to Table 1 for interface-related sections.

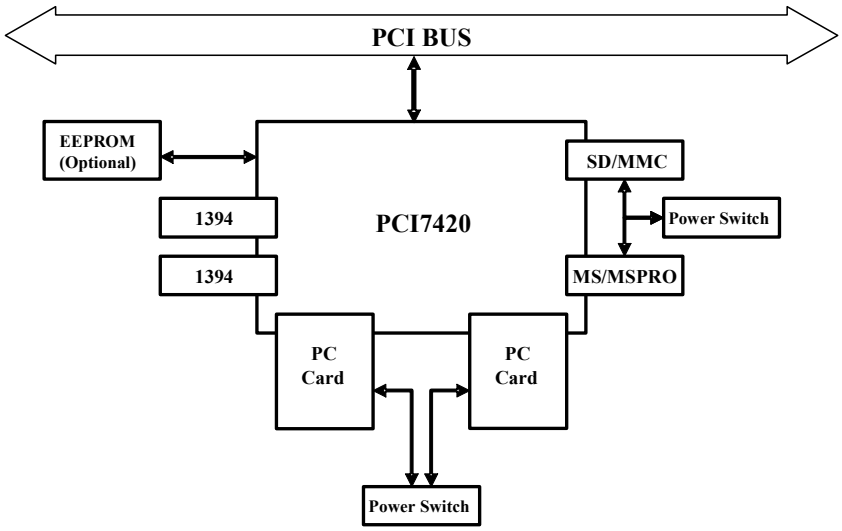


Figure 2. PCI7420 Typical System Implementation

1.3 PCI6620 Implementation Configuration

The PCI6620 (device ID 0xAC8Dh) controller supports four functions (CardBus, Smart Card, Secure Digital/Multimedia Card, and Memory Stick/Memory Stick-Pro), all of which can be individually enabled or disabled. For more product information, please refer to the PCI6620 data manual at http://www.ti.com. Please refer to Table 1 for interface-related sections.

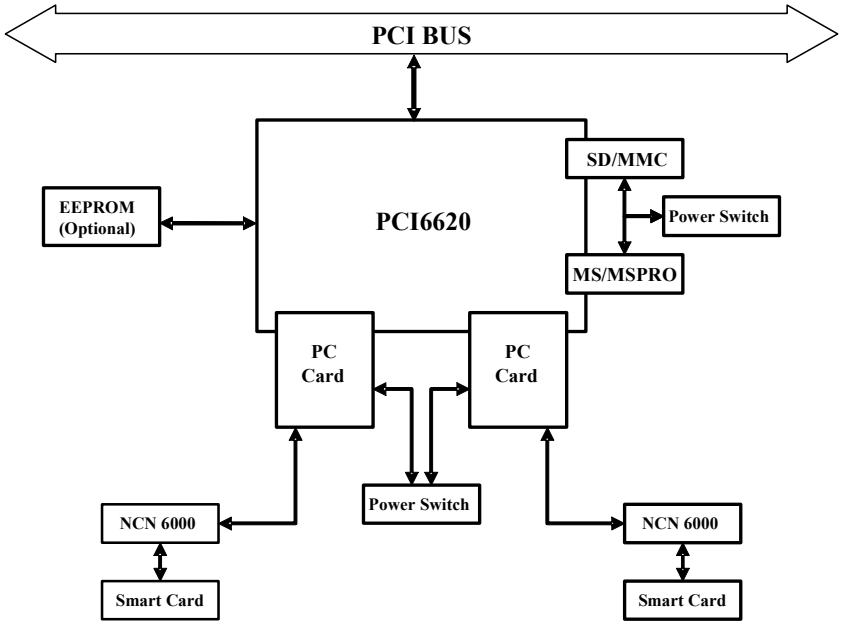


Figure 3. PCI6620 Typical System Implementation

1.4 PCI6420 Implementation Configuration

The PCI6420 (device ID 0xAC8Eh) controller supports three functions (CardBus, Secure Digital/Multimedia Card, and Memory Stick/Memory Stick-Pro), all of which can be individually enabled or disabled. For more product information, please refer to the PCI6420 data manual at http://www.ti.com. Please refer to Table 1 for interface-related sections.

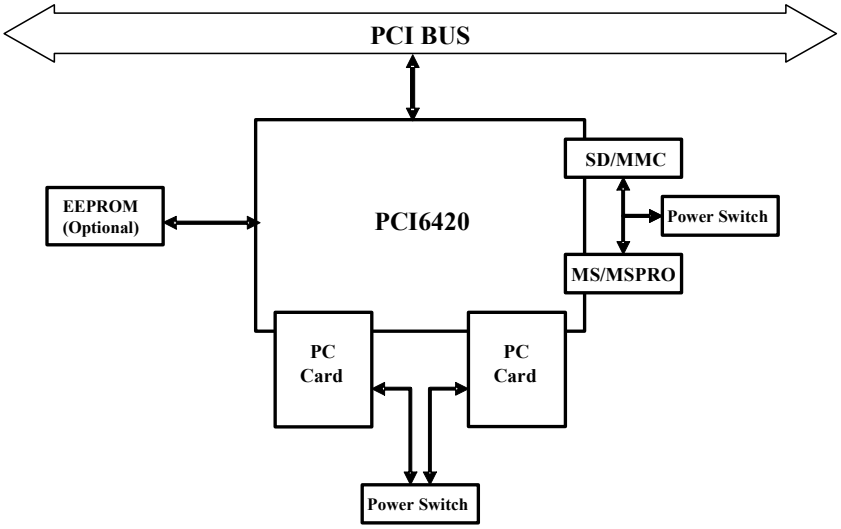


Figure 4. PCI6420 Typical System Implementation

Table 1. Document Sections by Interfaces

Interfaces	Section
Device Power	2
Power Switch	3
PCI	4
CardBus	5
EEPROM	11
Smart Card	6
Flash Media	7
1394	8

2 Power Considerations

2.1 Power Pin Description and Internal Voltage Regulator

The PCIXX20 controller contains different types of power terminals; system designers must pay close attention to the connections and power-up sequences of these terminals:

Table 2. Power Terminal Description

Terminal	Description
AVD2, AVD3, AVD4	Analog circuit power terminals
V_{CC}	Power supply terminal for I/O and internal voltage regulator. These terminals must all be tied to system-supplied 3.3-V power source.
V_{CCA}, V_{CCB}	Clamp voltage for PC Card interfaces. Matches socket signaling environment, 5 V or 3.3 V.
V_{CCP}	Clamp voltage for PCI and miscellaneous I/O, 5 V or 3.3 V
VR_PORT	1.8-V output from internal voltage regulator when $\overline{\text{VR\_EN}}$ is high, 1.8-V input when $\overline{\text{VR\_EN}}$ is low.
VDPLL	PLL circuit power terminal

The PCIXX20 controller uses an internal voltage regulator to power the core logic at 1.8 V. The voltage regulator is enabled using the $\overline{\text{VR\_EN}}$ terminal. If $\overline{\text{VR\_EN}}$ is high, then the voltage regulator is disabled and VR_PORT serves as a 1.8-V external input to power the core. If $\overline{\text{VR\_EN}}$ is low, then the internal voltage regulator is enabled, VR_PORT serves as a 1.8-V output.

2.2 Device Power Sequence

Depending on the host platform, different device power sequences are required. For 5-V tolerant systems, V_{CCP} (5 V) must be supplied first, then V_{CC}, VDPLL, and AVDx (3.3 V), finally VR_PORT (1.8 V – only if internal voltage regulator is disabled). For 3-V tolerant systems, V_{CC}, V_{CCP}, VDPLL, and AVDx (3.3 V) must be supplied before supplying VR_PORT (1.8 V – only if internal voltage regulator is disabled).

2.3 Bypass Capacitors

Standard design rules for the power supply bypass must be followed. The following sections are bypass capacitors recommended by Texas Instruments.

2.3.1 V_{CC}, V_{CCP}, V_{CCA}, and V_{CCB}

A 0.1-µF bypass capacitor is recommended for each of these power terminals.

2.3.2 VR_PORT

A 0.1-µF bypass capacitor is required on the VR_PORT terminal for decoupling.

Table 3. Requirements for Internal/External 1.8-V Core Power Supply

Supply	V_{CC}	$\overline{\text{VR\_EN}}$	VR_PORT	Note
Internal	3.3 V	GND	1.8-V output	Internal 1.8-V LDO-VR is enabled. This output is not for external use.
External	3.3 V	V_{CC}	1.8-V input	Internal 1.8-V LDO-VR is disabled. An external 1.8-V power supply, of minimum 50-mA capacity, is required.

2.3.3 VDPLL and AVDx

A parallel combination of high frequency decoupling capacitors near each terminal is suggested, such as 0.1 µF and 0.001 µF. Lower frequency 10-µF filtering capacitors are also recommended.

2.3.4 Flash Media Socket Power

A parallel combination of a 0.01-µF capacitor and a 10-µF capacitor placed on the power source close to the flash media socket is recommended.

3 Power Switch Implementation

3.1 CardBus Power Switch

The PCIXX20 controller can be used with any of the dual-slot TPS2228 and TPS222x (TPS2226, TPS2226A, TPS2224, TPS2224A, TPS2223, and TPS2223A) family of power switches. As shown in Table 4, bit 10 (12_SW_SEL) in the general control register at PCI configuration offset 86h can be configured to select the power switch that is implemented. The TPS222x dual-slot power switch is available in a pin-compatible (30-pin) package that allows system designers the ability to provide for either voltage level in a single design.

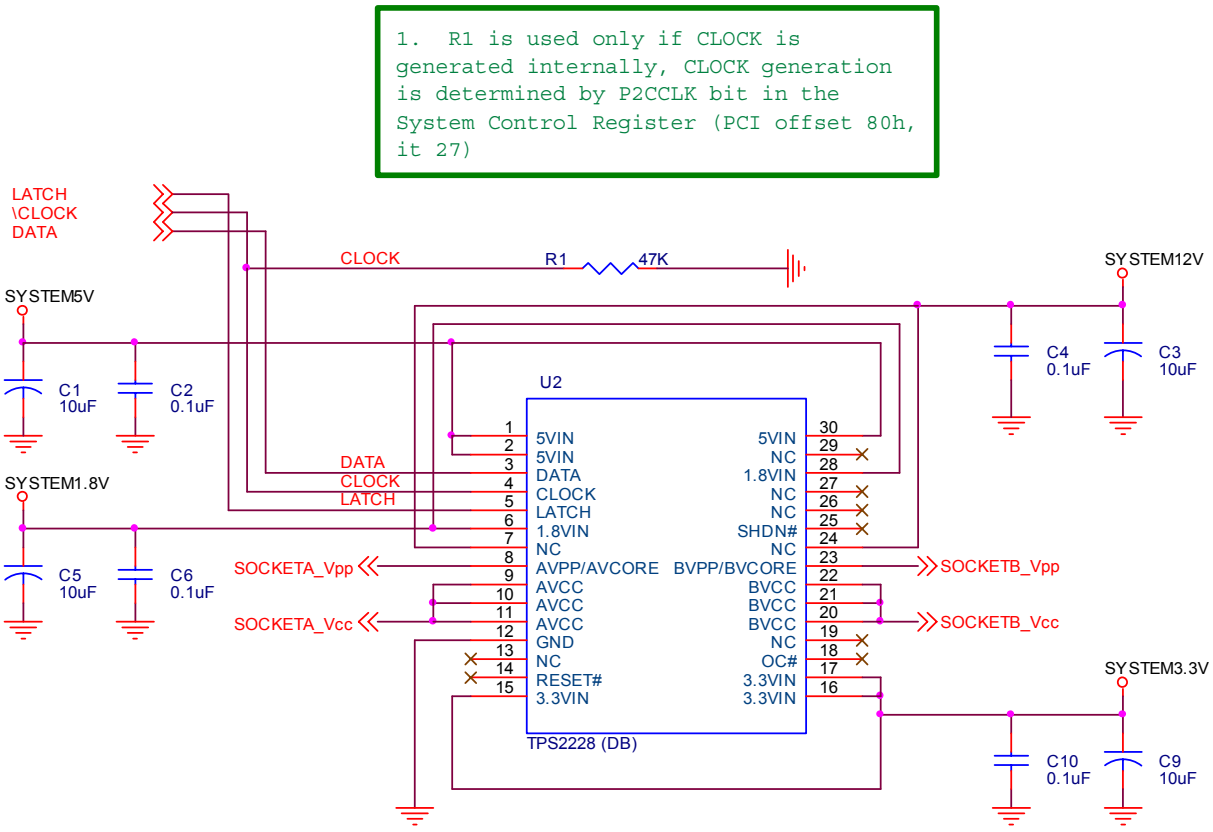


Figure 5. Power Switch Implementation

Table 4. Typical Power Switch Configuration

Voltage Requirement	Power Switch	12V_SW_SEL
1.8 V, 3.3 V, 5 V	TPS2228 (recommended)	0
3.3 V, 5 V, 12 V	TPS222x	1

A power switch is necessary in order to control power to the PC Card sockets. When the PCIXX20 controller receives a socket power request, it sends the appropriate data across the P^2C interface (CLOCK, DATA, and LATCH). In turn, the power switch turns on the appropriate levels for V_{CC} and V_{PP} for that socket. CLOCK can be provided either internally or externally depending on bit 27 (PSCCLK) in the system control register at PCI configuration offset 80h. If an external clock is used, then the frequency must be between 32 kHz and 100 kHz. If the internal clock is used, then a 43-kΩ pulldown resistor is necessary.

3.2 Flash Media Power Switch

Details on Flash Media power switch implementation are described in Section 7.

4 PCI Bus Interface

PCLK, AD31-AD0, C/$\overline{BE3}$-C/$\overline{BE0}$, PAR, $\overline{DEVSEL}$, and $\overline{FRAME}$, $\overline{STOP}$, $\overline{TRDY}$, $\overline{IRDY}$, $\overline{GNT}$, $\overline{REQ}$

These terminals can be connected directly to the system PCI bus. $\overline{GNT}$ and $\overline{REQ}$ are dedicated signals from the PCI bus arbitrator. PCLK is a peer-to-peer signal.

- $\overline{PERR}$, $\overline{SERR}$, and $\overline{LOCK}$

  $\overline{PERR}$ and $\overline{SERR}$ are required signals. $\overline{LOCK}$ is an optional signal and is available on the MFUNC terminals.

- IDSEL

  IDSEL selects the PCIXX20 controller during configuration space accesses. IDSEL can be connected to 1 of the upper 21 PCI address lines on the PCI bus.

  Please refer to the implementation note: *System Generation of IDSEL* in the *PCI Local Bus Specification*, Revision 2.3 (Section 3.2.2.3.5). *PCI Local Bus Specification*, Revision 2.3 (Section 4.2.6, footnote 31) recommends resistive coupling. A 100-Ω resistor is recommended.

- $\overline{PRST}$ and $\overline{GRST}$

  $\overline{GRST}$ initializes all registers and state machines of the PCIXX20 controller and $\overline{PRST}$ does not. $\overline{GRST}$ must be asserted during power-on and rebooting. It places the PCIXX20 controller into the initialized state. $\overline{PRST}$ does not initialize the global-rest-only bits or, if PME is enabled, the $\overline{PME}$ context bits.

- $\overline{INTA}$, $\overline{INTB}$, $\overline{INTC}$, and $\overline{INTD}$

  For more detailed information concerning these four interrupts, please refer to Section 10.

- $\overline{CLKRUN}$

  This signal is optional. However, if saving power is a concern, this signal may be implemented. Refer to Section 2 of the *PCI Mobile Design Guide* Revision 1.1.

- $\overline{PME}$

  This signal is required for ACPI systems. In a notebook PC, this signal is usually connected to the south bridge or embedded controller (EC). The $\overline{PME}$ terminal uses an open-drain type buffer.

- Pullup resistors

  Pullup resistors are needed on the following PCI terminals: $\overline{\text{IRDY}}$, $\overline{\text{TRDY}}$, $\overline{\text{FRAME}}$, $\overline{\text{STOP}}$, $\overline{\text{DEVSEL}}$, $\overline{\text{PERR}}$, $\overline{\text{SERR}}$, $\overline{\text{LOCK}}$, $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, $\overline{\text{INTC}}$, $\overline{\text{INTD}}$, $\overline{\text{CLKRUN}}$, and $\overline{\text{PME}}$. $\overline{\text{GRST}}$ requires a pullup resistor if the controlling output does not drive high. Please refer to *PCI Local Bus Specification*, Revision 2.3 for precise values of pullup resistors.

5 PC Card Interface

5.1 Operation Modes

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 PCIXX20 controller. These include: A14//$\overline{\text{CPERR}}$, A15//$\overline{\text{CIRDY}}$, A19//$\overline{\text{CBLOCK}}$, A20//$\overline{\text{CSTOP}}$, A21//$\overline{\text{CDEVSEL}}$, A22//$\overline{\text{CTRDY}}$, BVD2($\overline{\text{SPKR}}$)//CAUDIO, $\overline{\text{CD1}}$//$\overline{\text{CCD1}}$, $\overline{\text{CD2}}$//$\overline{\text{CCD2}}$, $\overline{\text{INPACK}}$//$\overline{\text{CREQ}}$, READY//$\overline{\text{CINT}}$, RESET//$\overline{\text{CRST}}$, $\overline{\text{VS1}}$//CVS1, $\overline{\text{VS2}}$//CVS2, $\overline{\text{WAIT}}$//$\overline{\text{CSERR}}$, and WP($\overline{\text{IOIS16}}$)//$\overline{\text{CCLKRUN}}$.
- Damping resistor on the CCLK terminal

  A series-damping resistor is recommended on the CCLK signal. The damping resistor is system dependent. If the line impedance is in the 60- to 90-Ω range, then a 47-Ω resistor is recommended (see *PC Card Standard*, Revision 8).
- CD line filtering

  The PCIXX20 controller features the advanced CDx line filtering circuit. It provides 90 ms of noise immunity.
- Socket power supply

  For more detailed information concerning the socket power supply, please refer to Section 3.
- 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 placed in a high-impedance state by clearing bit 22 (CBRSVD) in the system control register at PCI configuration offset 80h.

5.2 Implementing the PCIXX20 Controller as Single CardBus Socket Controller

The PCIXX20 controller can be used as a single socket controller simply by leaving the socket B interface unconnected. The DISABLE_SKTB bit (general control register, PCI offset 86h, bit 4) must be set to disable function 1 (CardBus socket B). This bit can be set via EEPROM or BIOS. Unused CardBus socket terminals must be left floating (no connection).

6 UltraMedia™ Smart Card Interfaces

In order to configure one or both of the PCIX620 (PCI7620 or PCI6620) UltraMedia™ CardBus sockets to be a Smart Card (SC) compatible socket, an SC socket connector (AMP 145300-1 or equivalent) and an SC electrical interface (NCN6000 or equivalent) must be used. An SC socket connector mechanically supports and recognizes a PCIX620-supported Smart Card. The NCN6000 power switch provides the electrical interface between the ICC (Smart Card) and the PCIX620 controller. The SC socket connector and the NCN6000 power switch function on a per-socket basis.

Once an SC insertion is detected, the PCIX620 controller asserts the socket query driver signal (SQRYDRV) high and monitors the SQRY [6:1] signals to determine the UltraMedia™ interface type and its corresponding voltage requirements. The query signal assignments are given in Tables 5, 6, and 7.

Table 5. Card Detect and Voltage Sense Connection for UltraMedia™

$\overline{CD2}$ // $\overline{CCD2}$	$\overline{CD1}$ // $\overline{CCD1}$	$\overline{VS2}$ //CVS2	$\overline{VS1}$ //CVS1	Key	Interface	Voltage
GND	CVS1	GND	$\overline{CCD1}$	LV	UltraMedia™	Per Query terminals

Table 6. Query Terminals – Voltage

SQRY2	SQRY1	Card Voltage
0	1	V_{CC} = 5 V, V_{PP} = 3.3 V

Table 7. Query Terminals – Smart Card Interface Implementation

SQRY6	SQRY5	SQRY4	SQRY3	Interface Implementation
0	1	0	0	Smart Card interface

Note:

If a 1 value is needed for a query terminal, then that terminal is connected to the query driver terminal through a 10-kΩ resistor. If a 0 value is needed for a query terminal, then that terminal is connected to ground.

For the Smart Card socket implementation, please follow the guidelines in Dedicated Smart Card Socket Implementation Schematics in Section 15. In this particular implementation, an AMP 145300-1 socket connector was used.

All non-Smart Card related terminals can be left unconnected.

Table 8. Smart Card Terminals

NAME	SLOT A	SLOT B
	GHK	GHK
$\overline{\text{MC\_CD}}$	C08	H13
SC_CLK	B08	H17
SC_FCB	E08	G18
SC_GPIO0	C07	F19
SC_GPIO1	B06	F17
SC_GPIO2	A05	G15
SC_GPIO3	B05	E18
SC_GPIO4	C05	F15
SC_GPIO5	E06	C19
SC_GPIO6	A03	D17
SC_IO	B07	H15
SC_RFU	E09	H19
SC_RST	A07	G17
SQRYDR	A06	H14
SQRY1	B08	H17
SQRY2	C06	E19
SQRY3	B02	B18
SQRY4	F06	E17
SQRY5	B04	D18
SQRY6	B01	C17
SQRY7	C03	A18
SQRY8	D03	B16
SQRY9	D01	A16
SQRY10	E03	E13

The electrical interface between the Smart Card and the reader is provided via the NCN6000 power switch. Due to the stringent voltage requirements for the socket V_{CC} and the three interface signals, there are some strict implementation requirements as well as several optional recommendations that help meet the level 1 type approval requirements.

6.1.1 Socket V_{CC}

The V_{CC} to the socket is provided via an integrated switching regulator in the NCN6000 power switch. A 22-µH inductor (L2) with an ESR less than 2 Ω is recommended to optimize the dc/dc conversion efficiency. The Murata LQH3C220K04 or equivalent is suggested.

Capacitor C14 stores the energy at the CRD_VCC output. To minimize ripple, which is critical in meeting the Europay, MasterCard, and Visa specification 2000 (EMV) V_{CC} output requirements, ceramic X7R capacitors are recommended for C14 and C15.

In addition, it has been found that placement of a 0.1-µF capacitor (C20) on SC_VCC near the socket is critical to minimize the voltage transients introduced during the V_{CC} transient tests.

The reference implementation includes a 10-kΩ resistor (R16) from SC_VCC to ground to provide a fixed load. This component also allows the output voltage to be modified based on the loading, for example, if the output on the implementation is regulating too high.

6.1.2 CLK, RST, and I/O

EMV 2000 specifies requirements for V_{OH}, V_{OL}, rise/fall time, and perturbation limits for the three interface signals. Design options have been implemented to address these requirements.

Previous level one testing has indicated that resistor and capacitor components on CLK (R18, C18) and RST (R19, C19) have been necessary to adjust the rise and fall times to reduce the overshoot and undershoot perturbations due to fast rise and fall times. In addition, the load provided by the resistor can allow for adjustment of the output high and low voltages.

It is also recommended that option pads for R17 and C17 be placed for I/O. These components, like those on CLK and RST, allow for adjustment of the rise and fall time which can affect the perturbations as well as the loading which can affect the V_{OH}/V_{OL} measurements. Past experience has shown that a 100-pF capacitor may be necessary to reduce perturbations on I/O due to fast rise and fall times. The resistor is typically not necessary. Evaluation of the V_{OH}, V_{OL}, t_r/t_f, and perturbations as well as the input requirements can determine if these components are required.

6.1.3 General Layout Recommendations

To meet the V_{OH}/V_{OL} and perturbation requirements care must be taken to avoid noise coupling on the CLK, RST, and I/O signals. If possible, it is recommended to route parallel ground traces beside CLK to reduce noise coupling of the CLK onto the RST and I/O signals.

The coupling of other signals in the system must be considered based on proximity to other noise sources such as switching regulators, high-speed clocks, etc. If the trace length between the NCN6000 power switch and the socket exceeds 1.5", then 15-mil spacing is recommended for the CLK, RST, and I/O traces (assuming a 5-mil trace width). This is to prevent noise coupling due to long parallel runs to adjacent signals. Otherwise, 2-mil spacing must be maintained.

The bulk and decoupling capacitor (C14, C15, and C16) as well as the inductor L2 must be placed near the NCN6000 power switch. Ceramic X7R capacitors are suggested for C14, C15, and C16 to minimize the ripple on the output voltage. L2 must be 22 µH with an ESR < 2 Ω. Murata LQH3C220K04 or equivalent is recommended.

Placement of C20 on SC_VCC must be near the socket to prevent large transients on the socket V_{CC} during the switched load tests.

The placement of components R17, R18, R19, C17, C18, and C19 are not as critical and can be placed near the NCN6000 power switch or the socket, whichever is convenient.

An optional ferrite (Murata BLM15AG121SN1 or equivalent) may be included to minimize V_{CC} noise to the NCN6000 power switch. If an option is included, then a 0-Ω resistor may be populated by default and populated with the ferrite only if it is found that the supply voltage input is excessively noisy.

7 Flash Media Interfaces

7.1 Dedicated Flash Media Socket

The PCIXX20 controller has two dedicated flash media sockets: one dedicated to support Secure Digital/Multimedia Card (SD/MMC), and the other to support Memory Stick/Memory Stick-Pro (MS/MSPRO). The two dedicated flash media sockets can be used concurrently.

7.1.1 Universal Power Switch Implementation

Due to the cross-platform consideration for many system designers, Texas Instruments has implemented a power switch implementation that maximizes the configurability and minimizes routing and layout changes for the flash media implementation.

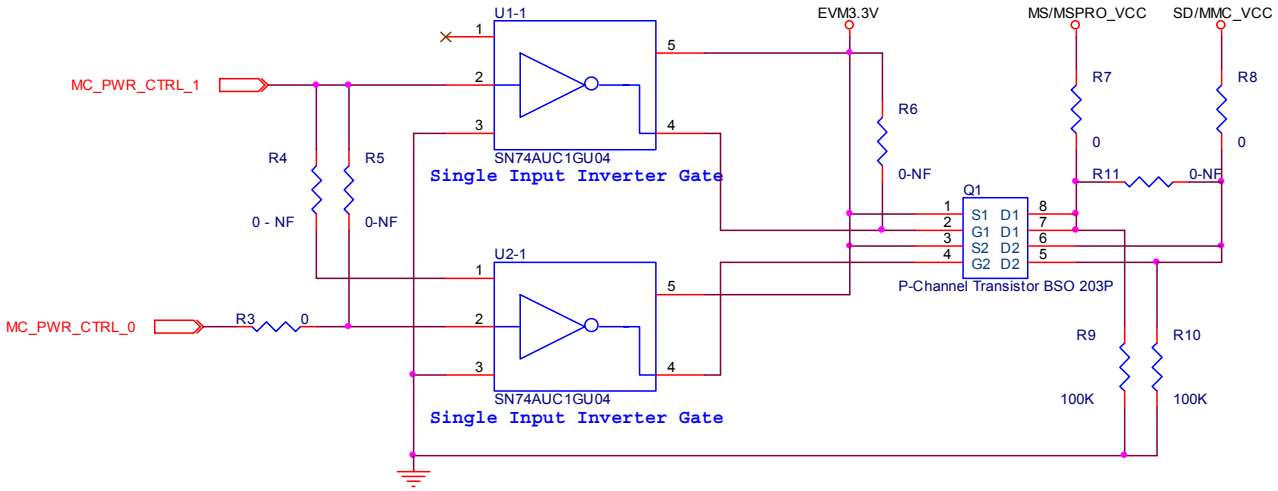


Figure 6. Universal Flash Media Power Switch Implementation

The implementation in Figure 6 allows for concurrent operation between SD/MMC and MS/MSPRO. In this implementation, a dual P-channel transistor (BSO 203P or equivalent) and two single inverting gates (SN74AUC1GU04 or equivalent) are used. Please see Sections 7.2.2.2 and 7.2.3.2 for socket connector connection.

7.1.1.1 Universal Power Switch Implementation (SD/MMC Option)

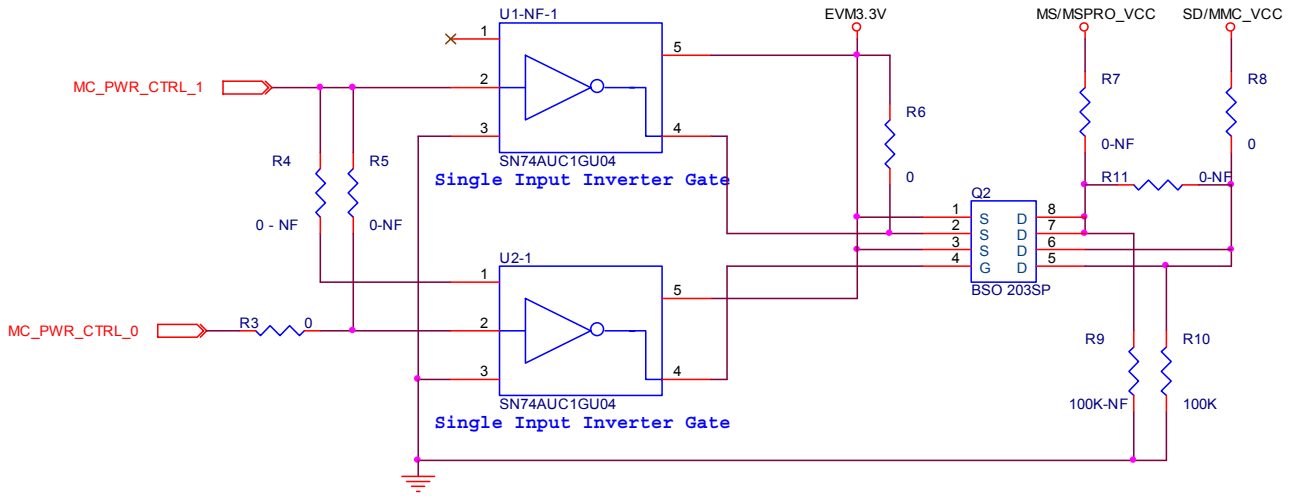


Figure 7. Universal Flash Media Power Switch Implementation (SD/MMC Option)

For system platforms that support the SD/MMC function exclusively, system designers can simply replace the BSO 203P transistor with a single-channel, P-channel transistor (BSO 203SP or equivalent), remove one of the inverting buffers (U1A), populate R6, and remove R7 and R9. These changes allow the PCIXX20 controller to support SD/MMC. To minimize development cost, all new components are in the exact same footprint as the components which were replaced, so no design or layout change is necessary. Please see Section 7.2.2.1 for socket connector selection.

7.1.1.2 Universal Power Switch Implementation (MS/MSPRO Option)

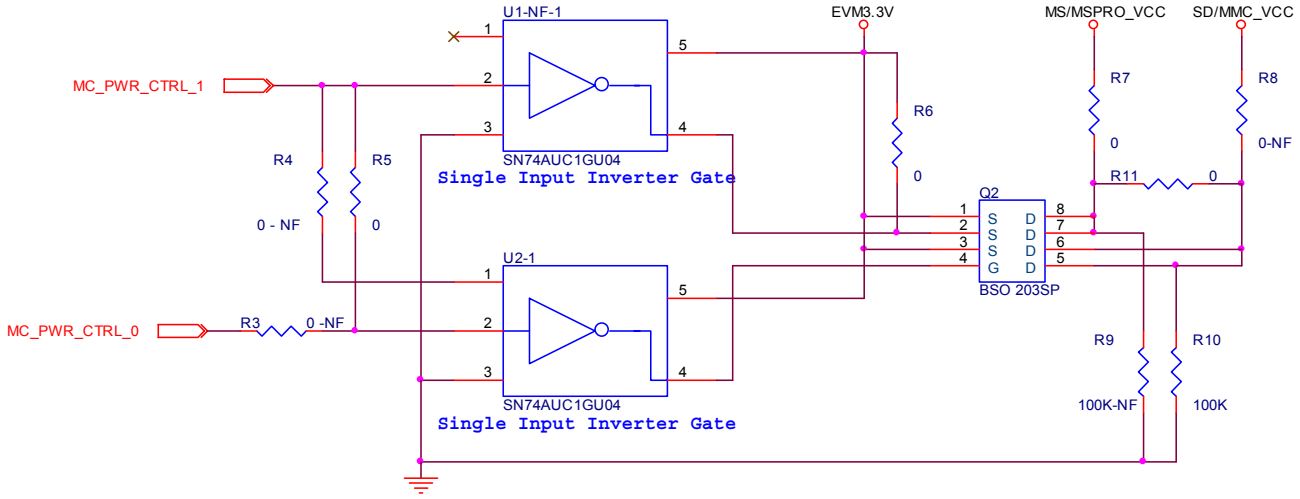


Figure 8. Universal Flash Media Power Switch Implementation (MS/MSPRO Option)

For system platforms that support the MS/MSPRO function exclusively, system designers can simply replace the BSO 203P transistor with a single-channel P-channel transistor (BSO 203SP or equivalent), remove one of the inverting buffers (U1A), populate R5, R6, and R11, and remove R3, R8, and R9. These changes allow the PCIXX20 controller to support MS/MSPRO. To minimize development cost, all new components are in the exact same footprint as the components which were replaced, so no design or layout change is necessary. Please see Section 7.2.3.1 for socket connector selection.

7.1.1.3 Universal Power Switch Implementation (2-In-1 Option)

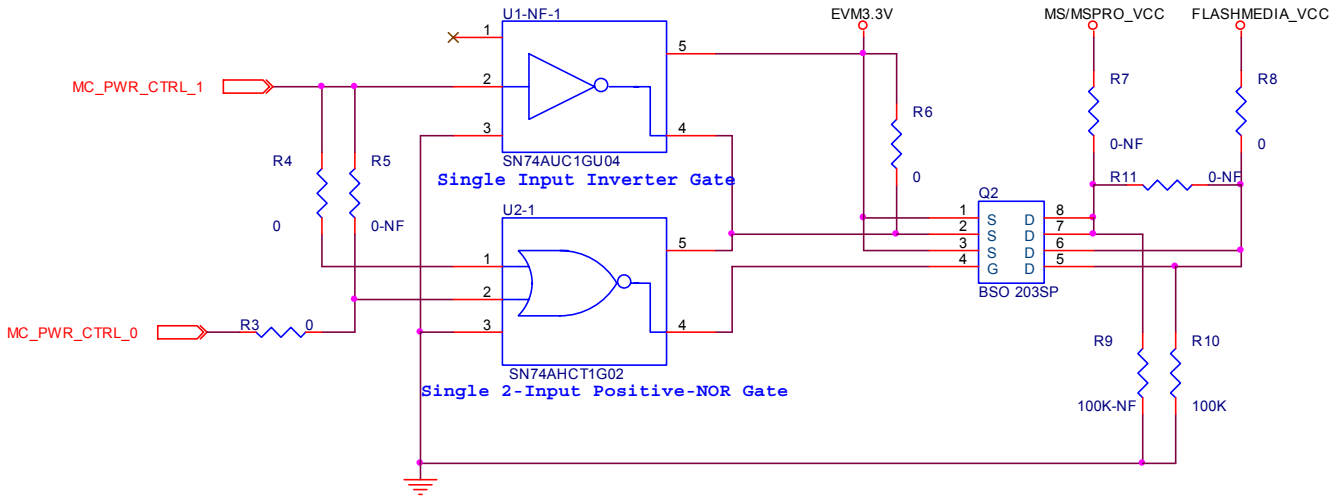


Figure 9. Universal Flash Media Power Switch Implementation (2-In-1 Option)

For system platforms that support both SD/MMC and MS/MSPRO functions nonconcurrently, system designers can simply replace the BSO 203P transistor with a single-channel P-channel transistor (BSO 203SP or equivalent), remove one of the inverting buffers (U1A), replace U2 with a 2-input NOR gate (SN74AHCT1G02 or equivalent), populate R4 and R6, and remove R7 and R9. These changes allow the PCIXX20 controller to support both SD/MMC and MS/MSPRO nonconcurrently. To minimize development cost, all new components are in the exact same footprint as the components which were replaced, so no design or layout change is necessary. Please see Section 7.2.4 for socket connector selection.

If system designers wish to employ a more optimized implementation, please see Sections 7.2.2.2 and 7.2.3.2.

7.1.2 Dedicated Secure Digital/Multimedia Card Socket

In order to take advantage of the dedicated SD/MMC socket, a SD/MMC flash media socket connector and a 3.3-V power switch are recommended.

7.1.2.1 Secure Digital/Multimedia Card Socket Connector

For SD/MMC applications, a Yamaichi FPS009-2203-10 or equivalent is recommended. If a socket connector other than the recommended Yamaichi is desired, then care must be taken to ensure that the chosen socket connector has built-in support for the write-protect (SD_WP) pin. If one without the SD_WP support is chosen, then the SD/MMC write-protection is not functional.

In addition, 0.01-µF and 10-µF capacitors placed close to the connector are recommended to eliminate any noise on the SD/MMC electrical interface. Due to variations on layout constraints, series resistors are placed on the interface between the PCIXX20 controller and the SD/MMC socket connector in case signal integrity tuning is required. A series resistor on SD_CLK is required due to the internal feedback circuitry of the controller; series resistors on other SD/MMC signals are recommended but not required. These tuning series resistors must be placed as close to the socket connector as possible. Trace lengths on all SD/MMC interface signals must be kept similar to each other to minimize the difference in propagation delay and capacitance.

Table 9. SD/MMC Terminals

Name	GHK Terminals
$\overline{\text{MC\_CD\_0}}$	E02
SD_CLK	H01
SD_CMD	H02
SD_DAT0	H03
SD_DAT1	J07
SD_DAT2	J01
SD_DAT3	J02
SD_WP	J03
MC_PWR_CTRL_0	F01

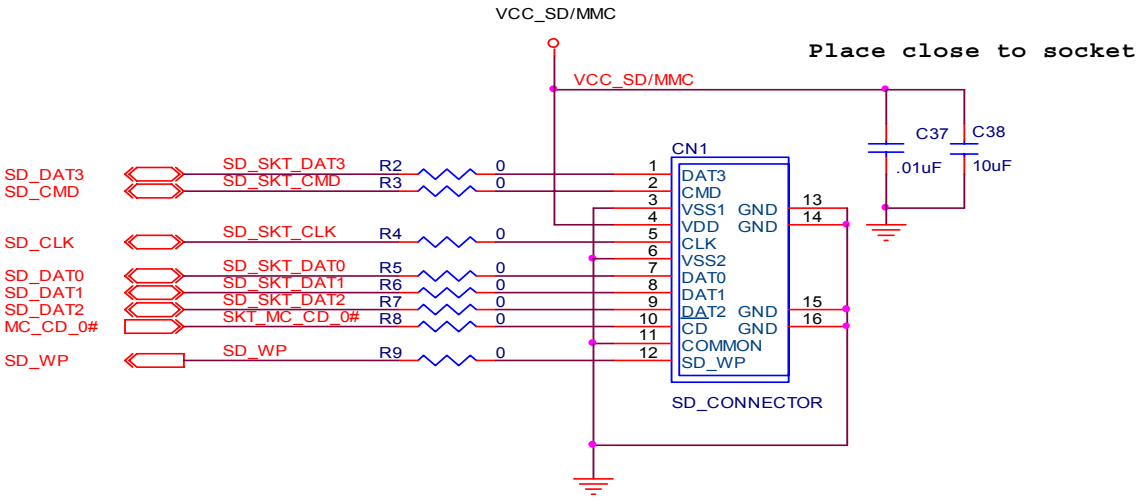


Figure 10. SD/MMC Socket Connector Implementation

7.1.2.2 Secure Digital/Multimedia Card Power Switch

System designers can power the SD/MMC socket connector by using either a power switch (TPS2030 or equivalent) or a FET power switch implementation.

If a power switch is chosen, then a 0.1-µF bypass capacitor connected between SD/MMC V_{CC} and GND is recommended.

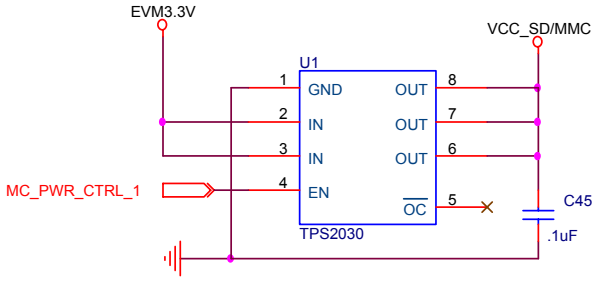


Figure 11. SD/MMC Power Switch Implementation

If the FET implementation is chosen, then a single inverter gate (SN74AUC1GU04 or equivalent) is recommended in combination with a P-channel transistor (BSO 203SP or equivalent).

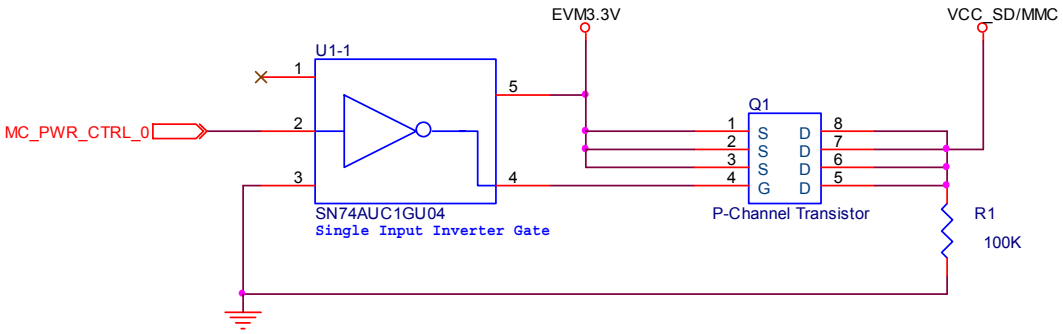


Figure 12. SD/MMC FET Power Switch Implementation

7.1.3 Dedicated Memory Stick/Memory Stick-Pro Card Socket

In order to take advantage of the dedicated MS/MSPRO socket, a MS/MSPRO flash media socket connector and a 3.3-V power switch are recommended.

7.1.3.1 Memory Stick/Memory Stick-Pro Socket Connector

For MS/MSPRO applications, a Hirose CB1AA-10S-1-5H or equivalent is recommended. In addition, 0.01-µF and 10-µF capacitors placed close to the socket are recommended to eliminate any noise on the MS/MSPRO electrical interface. Due to variations on layout constraints, series resistors are placed on the interface between the controller and the MS/MSPRO socket connector in case signal integrity tuning is required. A series resistor on MS_CLK is required due to the internal feedback circuitry of the controller; series resistors on other MS/MSPRO signals are recommended but not required. These tuning series resistors must be placed as close to the socket connector as possible. Trace lengths on all MS/MSPRO interface signals must be kept similar to each other to minimize the difference in propagation delay and capacitance.

Table 10. MS/MSPRO Terminals

Name	GHK Terminals
$\overline{\text{MC\_CD\_1}}$	E01
MS_CLK	G05
MS_BS	F02
MS_SDIO (DATA0)	G01
MS_DATA1	G02
MS_DATA2	G03
MS_DATA3	H07
MC_PWR_CTRL_1	F03

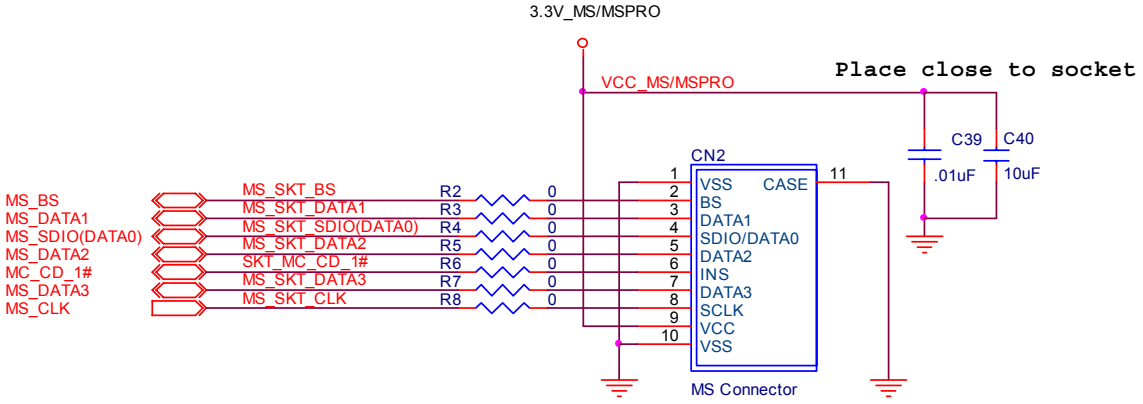


Figure 13. MS/MSPRO Socket Connector Implementation

7.1.3.2 Memory Stick/Memory Stick-Pro Card Power Switch

System designers can power the MS/MSPRO socket connector by using either a power switch (TPS2030 or equivalent) or a FET power switch implementation.

If a power switch is chosen, then a 0.1-µF bypass capacitor connected between MS/MSPRO V_{cc} and GND is recommended.

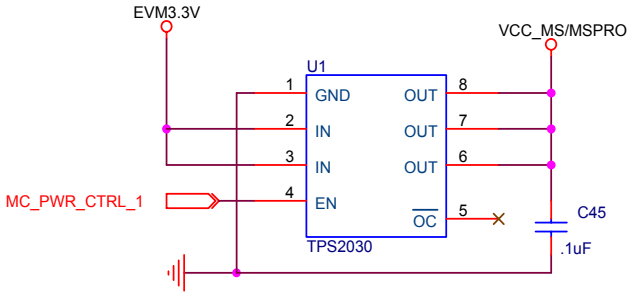


Figure 14. MS/MSPRO Power Switch Implementation

If the FET implementation is chosen, then a single inverter gate (SN74AUC1GU04 or equivalent) is recommended in combination with a P-channel transistor (BSO 203SP or equivalent).

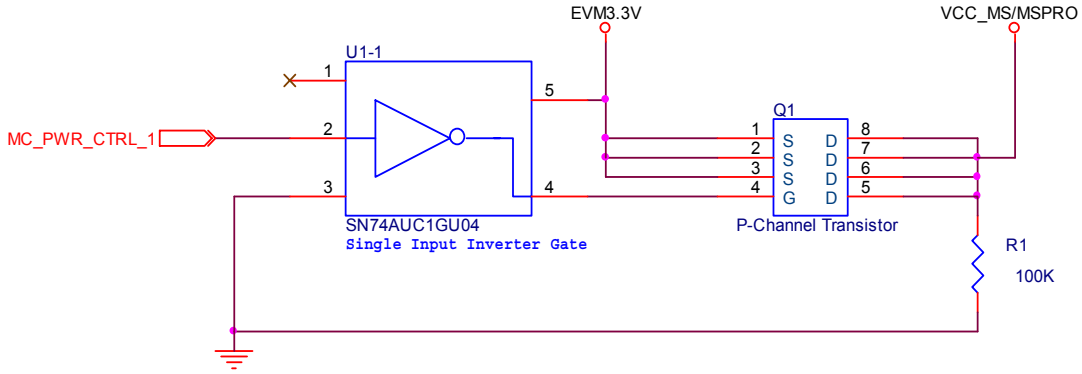


Figure 15. MS/MSPRO FET Power Switch Implementation

7.1.4 Flash Media Support With 2-In-1 Socket Connector

For system designers who would like to support both SD/MMC and MS/MSPRO but do not see the need for the concurrent operation of both types of media cards, a 2-in-1 socket connector (Alps SCDB series or equivalent) is recommended. With this implementation, system designers are able to provide flash media support with either SD/MMC or MS/MSPRO inserted into the 2-in-1 socket connector. Due to variations on layout constraints, series resistors are placed on the interface between the controller and MS/MSPRO socket connector, and also on the controller and SD/MMC socket connector in case signal integrity tuning is required. Series resistors on MS_CLK and SD_CLK are required due to the internal feedback circuitry of the controller; series resistors on other MS/MSPRO and SD/MMC signals are recommended but not required. These tuning series resistors must be placed as close to the socket connector as possible. Trace lengths on all SD/MMC and MS/MSPRO interface signals must be kept similar to each other to minimize the difference in propagation delay and capacitance.

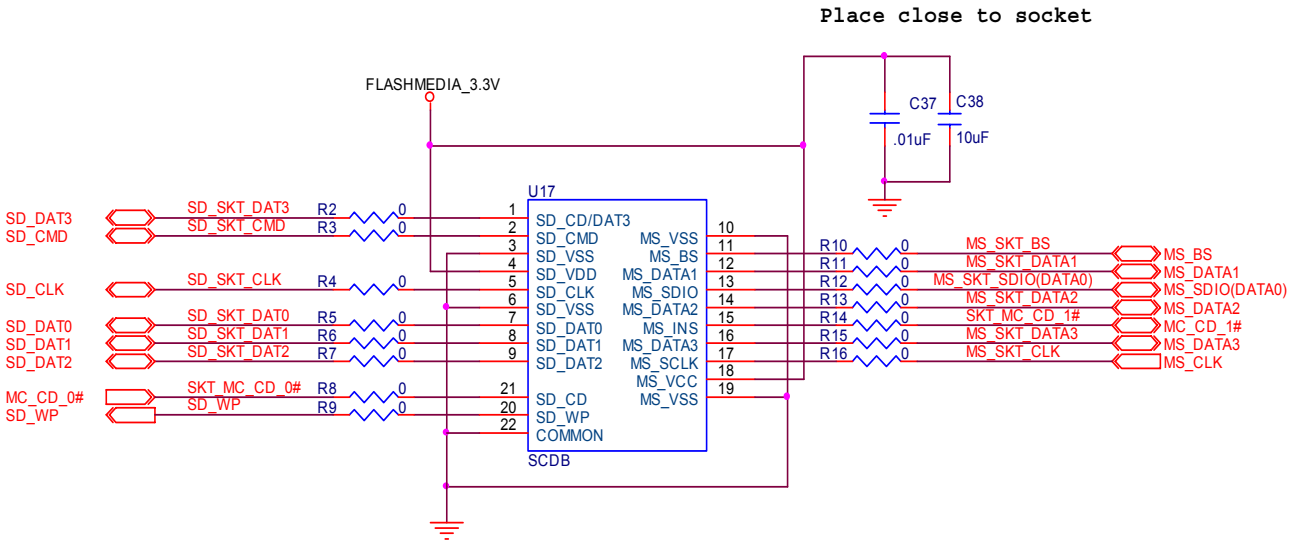


Figure 16. 2-In-1 Socket Connector Implementation

7.1.4.1 Flash Media Support With 2-In-1 Socket Connector Power Switch Implementation

System designers can power the 2-in-1 socket connector by using either a power switch (TPS2030 or equivalent) with an OR gate (SN74AHCT1G32 or equivalent) or a FET power switch implementation.

If a power switch with an OR gate is chosen, then a 0.1-µF bypass capacitor connected between flash media V_{CC} and GND is recommended.

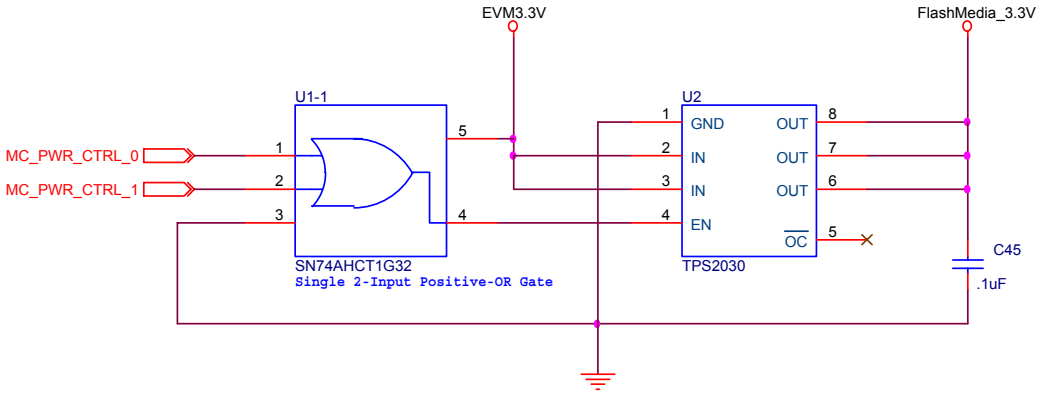


Figure 17. 2-In-1 Power Switch Implementation

If the FET power switch implementation is desired, then a NOR gate (SN75AHCT1G02 or equivalent) and a P-channel transistor (BSO 203SP or equivalent) are recommended.

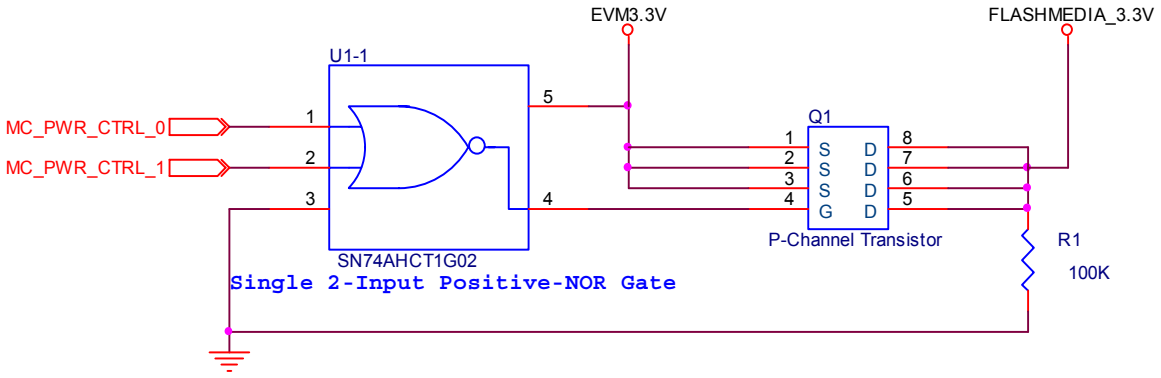


Figure 18. 2-In-1 FET Power Switch Implementation

8 PHY Port Cable Connection

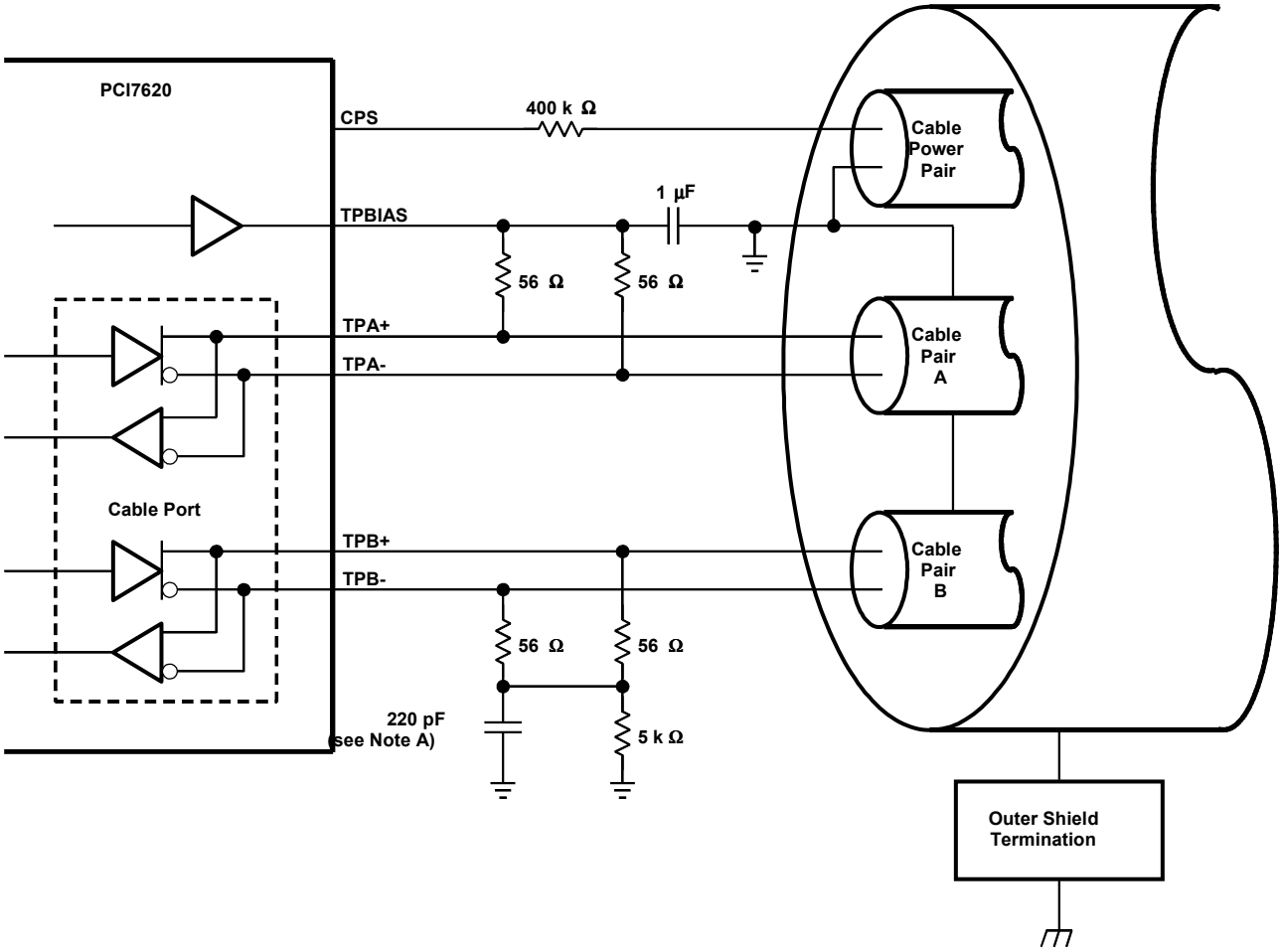


Figure 19. TP Cable Connections

NOTE A: IEEE Std 1394–1995 calls for a 250–pF capacitor, which is a nonstandard component value. A 220–pF capacitor is recommended.

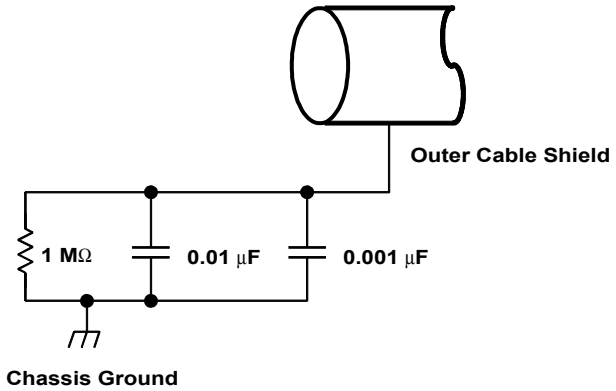


Figure 20. Typical Compliant DC Isolated Outer Shield Termination

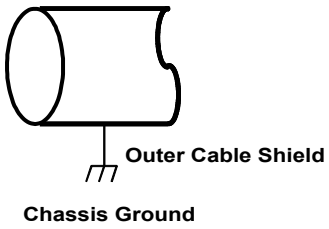


Figure 21. Non-DC Isolated Outer Shield Termination

8.1 Crystal Selection

The PCI7X20 (PCI7620 or PCI7420) controller is designed to use an external 24.576-MHz crystal connected between the XI and XO terminals to provide the reference for an internal oscillator circuit. This oscillator in turn drives a PLL circuit that generates the various clocks required for transmission and resynchronization of data at the S100 through S400 media data rates.

A variation of less than ±100 ppm from nominal for the media data rates is required by IEEE Std 1394–1995. Adjacent PHYs may therefore have a difference of up to 200 ppm from each other in their internal clocks, and PHY devices must be able to compensate for this difference over the maximum packet length. Large clock variations may cause resynchronization overflows or underflows, resulting in corrupted packet data.

The following are some typical specifications for crystals used with the PHYs from TI in order to achieve the required frequency accuracy and stability:

- Crystal mode of operation: Fundamental
- Frequency tolerance @ 25°C: Total frequency variation for the complete circuit is ±100 ppm. A crystal with ±30 ppm frequency tolerance is recommended for adequate margin.
- Frequency stability (over temperature and age): A crystal with ±30 ppm frequency stability is recommended for adequate margin.

Note:
The total frequency variation must be kept below ±100 ppm from nominal with some allowance for error introduced by board and device variations. Trade-offs between frequency tolerance and stability may be made as long as the total frequency variation is less than ±100 ppm. For example, the frequency tolerance of the crystal may be specified at 50 ppm and the temperature tolerance may be specified at 30 ppm to give a total of 80 ppm possible variation due to the crystal alone. Crystal aging also contributes to the frequency variation.

- Load capacitance: For parallel resonant mode crystal circuits, the frequency of oscillation is dependent upon the load capacitance specified for the crystal. Total load capacitance (CL) is a function of not only the discrete load capacitors, but also board layout and circuit. It is recommended that load capacitors with a maximum of ±5% tolerance be used.

The load specified for the crystal includes the load capacitors (C9 and C10), the loading of the PHY pins (C_{PHY}), and the loading of the board itself (C_{BD}). The value of C_{PHY} is typically about 1 pF, and C_{BD} is typically 0.8 pF per centimeter of board etch; a typical board can have 3 pF to 6 pF or more. The load capacitors C9 and C10 combine as capacitors in series, so that the total load capacitance is:

$$C_L = \frac{C9 \cdot C10}{C9+C10} + C_{PHY} + C_{BD}$$

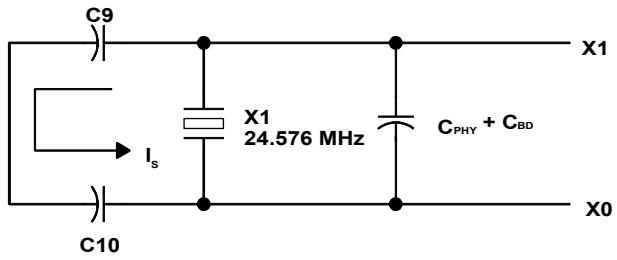


Figure 22. Load Capacitance for the PCI7X20 PHY

The layout of the crystal portion of the PHY circuit is important for obtaining the correct frequency, minimizing noise introduced into the PHY phase-lock loop, and minimizing any emissions from the circuit. The crystal and two load capacitors must be considered as a unit during layout. The crystal and the load capacitors must be placed as close as possible to one another while minimizing the loop area created by the combination of the three components. Varying the size of the capacitors may help in this. Minimizing the loop area minimizes the effect of the resonant current (I_S) that flows in this resonant circuit. This layout unit (crystal and load capacitors) must then be placed as close as possible to the PHY X1 and X0 pins to minimize etch lengths, as shown in Figure 22.

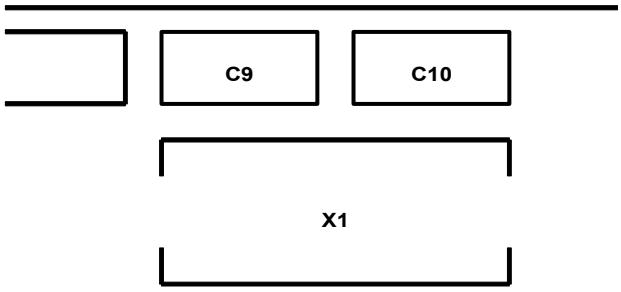


Figure 23. Recommended Crystal and Capacitor Layout

For more details on crystal selection, see application report SLLA051 available from the TI website: http://www.ti.com/sc/1394.

8.2 EMI Guidelines

To minimize EMI emissions, add decoupling capacitors with a ferrite bead at the VDPLL and AVDx terminals on the controller. This array must be as close as possible to the controller in order to minimize the inductance of the line and noise contributions to the system. Figure 24 shows a suggested array. In the case of the AVDx terminals, it is recommended to tie them up to a single low-impedance point on the board and then add the decoupling capacitors in addition to the ferrite bead. This array of capacitors and ferrite bead improve EMI and jitter performance. Both EMI and jitter must be taken into account before altering the configuration.

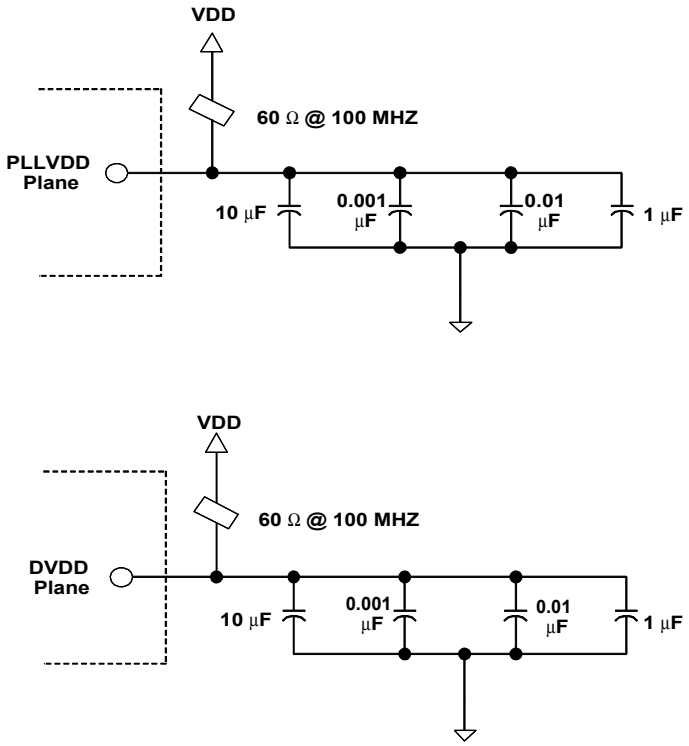


Figure 24. Suggested Array at VDPLL and AVDx in Order to Minimize EMI

For additional electromagnetic interference (EMI) guidelines and recommendations send a request via e-mail to 1394-EMI@list.ti.com.

8.3 Disabling IEEE1394 Ports

System designers must set bit 3 (DISABLE_OHCI) in the general control register at PCI configuration offset 86h to disable the 1394 function. This bit can be set via EEPROM or BIOS.

8.3.1 Pin Strapping of the Unused 1394 Terminals

Refer to Table 11 for pin strapping instructions.

Table 11. Pin Strapping of the Unused 1394 Terminals

Signal Name	Comments
CNA	No connect
CPS	Pull to V_{CC} via 1-kΩ resistor
FILTER0	Pull directly to GND
FILTER1	Pull directly to GND
PC0	Pull directly to GND
PC1	Pull directly to GND
PC2	Pull directly to GND
R0	No connect
R1	Pull directly to V_{CC}
TPA0P	No connect
TPA0N	No connect
TPA1P	No connect
TPA1N	No connect
TPBIAS0	No connect
TPBIAS1	No connect
TPB0P	Pull directly to GND
TPB0N	Pull directly to GND
TPB1P	Pull directly to GND
TPB1N	Pull directly to GND
XI	Pull directly to GND
XO	No connect

9 Miscellaneous Pin Interface

9.1 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), $\overline{\text{INTA}}$, and IRQSER are explained in Section 10, *Interrupt Configurations*. $\overline{\text{CLKRUN}}$, $\overline{\text{D3STAT}}$, and $\overline{\text{RI\_OUT}}$ are discussed in Section 12, *Power Management Considerations*. For more information, please refer to the PCIXX20 data manual.

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

$\overline{\text{GPE}}$, 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 $\overline{\text{LOCK}}$ is an optional PCI signal as mentioned in Section 5, *PCI Bus Interface*.

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

9.2 SPKROUT

SPKROUT is the output to the host system that can carry $\overline{\text{SPKR}}$ or CAUDIO through the PCIXX20 controller from the PC Card interface. If SPKROUT is enabled for both sockets, then it is driven as an exclusive-OR of the two inputs. A 43-kΩ pulldown resistor is required to prevent oscillation when SPKROUT is disabled and therefore in a high-impedance state.

9.3 $\overline{\text{SUSPEND}}$

The assertion of $\overline{\text{SUSPEND}}$ signal gates the $\overline{\text{PRST}}$ and $\overline{\text{GRST}}$ signals from the PCIXX20 controller. $\overline{\text{SUSPEND}}$ also gates PCLK inside the PCIXX20 controller in order to minimize power consumption. Gating PCLK makes the IRQSER state machine stop until $\overline{\text{SUSPEND}}$ is deasserted. Two requirements for implementing the suspend mode are:

- The PCI bus must not be parked on the PCIXX20 controller.
- IRQSER signaling is not proceeding when $\overline{\text{SUSPEND}}$ is asserted.

10 Interrupt Configurations

The PCIXX20 controller provides system designers with great flexibility in configuring interrupts. The PCIXX20 controller allows three interrupt modes which are selected via bits 2:1 of the device control register at PCI configuration offset 92h.

PCI interrupts are available on $\overline{\text{INTA}}$. 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.

Windows XP has simplified the 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

10.1 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 $\overline{\text{INTA}}$. This is not a recommended interrupt configuration because some 16-bit PC Cards require legacy ISA interrupts and do not function properly.

When using one of the parallel PCI interrupt modes, $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, $\overline{\text{INTC}}$, and $\overline{\text{INTD}}$ must be connected to the PCI interrupt lines. If the INTRTIE bit (system control register, PCI offset 80h, bit 29) is set, then both functions signal and report $\overline{\text{INTA}}$, therefore, only $\overline{\text{INTA}}$ needs to be routed. The TIEALL bit (system control register, PCI offset 80h, bit 28) ties $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, $\overline{\text{INTC}}$, and $\overline{\text{INTD}}$ together internally. The INTRTIE and TIEALL bits affect the read-only value provided through accesses to the interrupt pin register (PCI offset 3Dh, see Section 4.24). Table 12 summarizes the interrupt signaling modes.

Table 12. Interrupt Pin Register Cross Reference

INTIE Bit	TIEALL Bit	INTPIN Function 0 CardBus	INTPIN Function 1 CardBus	INTPIN Function 2 1394 OHCI	INTPIN Function 3 Flash Media
0	0	01h $\overline{\text{INTA}}$	02h $\overline{\text{INTB}}$	03h $\overline{\text{INTC}}$	Determined by bits 6-5 (INT_SEL_FIELD) in flash media general control register (PCI offset 0x86h)
1	0	01h $\overline{\text{INTA}}$	01h $\overline{\text{INTA}}$	03h $\overline{\text{INTC}}$	
X	1	01h $\overline{\text{INTA}}$	01h $\overline{\text{INTA}}$	01h $\overline{\text{INTA}}$	01h $\overline{\text{INTA}}$

10.2 Serial IRQ and Parallel PCI Interrupts

The serial IRQ and parallel PCI interrupts mode is selected by programming bits 2:1 to a value of 10b. This allows interrupts to be routed through IRQSER and $\overline{\text{INTA}}$. This is the recommended interrupt configuration for a PCI add-in card implementation of the PCIXX20 controller. $\overline{\text{INTA}}$ 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, then this mode still allows CardBus cards to function properly. However, some 16-bit cards may not.

10.3 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.

11 Software Considerations

11.1 CardBus and UltraMedia™

The PCI7X20 1394 and PCIXX20 CardBus functions are natively supported by Windows XP and Windows 2000. The PCI7X20 controller is recognized as a Texas Instruments IEEE 1394 device and the PCIXX20 controller as a generic CardBus controller. The controller functions properly using these designations. However, due to the added support for UltraMedia™ functions, new drivers and .inf files provided by Texas Instruments must be used. These drivers and .inf files allow the Flash Media function to be reported in device manager properly. To shorten system setup time, an install shield package for Windows 2000 and XP is provided by Texas Instruments to simplify the configuration process.

11.2 EEPROM Configuration

The following diagram represents a sample implementation of a generic EEPROM for the PCIXX20 configuration:

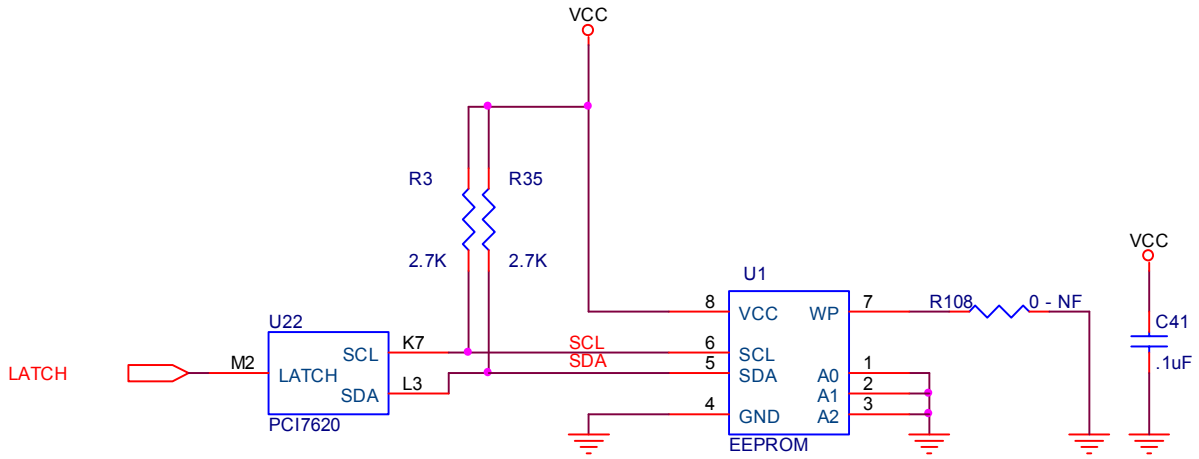


Figure 25. EEPROM Implementation

On the rising edge of $\overline{\text{GRST}}$, the serial bus detect bit (bit 3, PCI offset B3h) is set and the EEPROM contents are loaded into the PCIXX20 controller. In order for the controller to detect the EEPROM and load configuration information, pullup resistors are needed on SDA and SCL. The EEPROM slave address must be 101 0000b.

The following is EEPROM map which could be loaded into the EEPROM for use with the PCIXX20 controller:

Table 13. PCIXX20 EEPROM Loading Map

Serial ROM Offset	Byte Description						
00h	CardBus function indicator (00h)						
01h	Number of bytes (20h)						
02h	PCI 04h, command register, function 0, bits 8, 6-5, 2-0						
	[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
03h	PCI 04h, command register, function 1, bits 8, 6-5, 2-0						
	[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, subsystem vendor ID, byte 0						
05h	PCI 41h, subsystem vendor ID, byte 1						
06h	PCI 42h, subsystem ID, byte 0						
07h	PCI 43h, subsystem ID, byte 1						
08h	PCI 44h, PC Card 16-bit I/F legacy mode base address register, byte 0, bits 7-1						
09h	PCI 45h, PC Card 16-bit I/F legacy mode base address register, 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	Reserved load all zeros (PCI 82h, system control, byte 2)						
10h	PCI 83h, system control, byte 3						
11h	PCI 8Ch, multifunction routing, byte 0						
12h	PCI 8Dh, multifunction routing, byte 1						
13h	PCI 8Eh, multifunction routing, byte 2						
14h	PCI 8Fh, multifunction routing, byte 3						
15h	PCI 90h retry status, bits 7, 6						
16h	PCI 91h, card control, bit 7						
17h	PCI 92h, device control, bits 6, 5, 3-0						
18h	PCI 93h, diagnostic, bits 7, 6, 4-0						
19h	PCI A2h, power management capabilities, function 0, bit 15 (bit 7 of EEPROM offset 16h corresponds to bit 7)						
1Ah	PCI A2h, power management capabilities, function 1, bit 15 (bit 7 of EEPROM offset 16h corresponds to bit 7)						
1Bh	Reserved, load all zeros (CB socket + 0Ch, function 0 socket force event)						
1Ch	Reserved, load all zeros (CB socket + 0Ch, function 1 socket force event)						
1Dh	ExCA 800h, ExCA identification and revision, bits 7-0						
1Eh	PCI 86h, general control, byte 0, bits 5, 4, 3, 1, 0						
1Fh	PCI 87h, general control, byte 1, bits 4-2						
20h	PCI 89h, $\overline{\text{GPE}}$ enable, bits 7, 6, 4-0						
21h	PCI 8Bh, general-purpose output, bits 4-0						

Serial ROM Offset	Byte Description					
22h	1394 OHCI function indicator (02h) (RSVD for PCI6X20)					
23h	Number of bytes (17h) (RSVD for PCI6X20)					
24h	PCI 3Fh, MaxLat, bits 3-0 (RSVD for PCI6X20)			PCI 3Eh, MinGnt, bits 3-0 (RSVD for PCI6X20)		
25h	PCI 2Ch, subsystem vendor ID, byte 0 (RSVD for PCI6X20)					
26h	PCI 2Dh, subsystem vendor ID, byte 1 (RSVD for PCI6X20)					
27h	PCI 2Eh, subsystem ID, byte 0 (RSVD for PCI6X20)					
28h	PCI 2Fh, subsystem ID, byte 1 (RSVD for PCI6X20)					
29h	PCI F4h, Link_Enh, byte 0, bits 7, 2, 1 (RSVD for PCI6X20) OHCI 50h, host controller control, bit 23 (RSVD for PCI6X20)					
	[7] Link_Enh. enab_unfair	[6] HCControl. program PHY enable	[5:3] RSVD	[2] Link_Enh, bit 2	[1] Link_Enh. enab_accel	[0] RSVD
2Ah	Mini-ROM address, this byte is used to indicate the Mini ROM offset into the EEPROM 00h = No Mini ROM Other values = Mini ROM offset (RSVD for PCI6X20)					
2Bh	OHCI 24h, GUIDHi, byte 0 (RSVD for PCI6X20)					
2Ch	OHCI 25h, GUIDHi, byte 1 (RSVD for PCI6X20)					
2Dh	OHCI 26h, GUIDHi, byte 2 (RSVD for PCI6X20)					
2Eh	OHCI 27h, GUIDHi, byte 3 (RSVD for PCI6X20)					
2Fh	OHCI 28h, GUIDLo, byte 0 (RSVD for PCI6X20)					
30h	OHCI 29h, GUIDLo, byte 1 (RSVD for PCI6X20)					
31h	OHCI 2Ah, GUIDLo, byte 2 (RSVD for PCI6X20)					
32h	OHCI 2Bh, GUIDLo, byte 3 (RSVD for PCI6X20)					
33h	Checksum (reserved – no bit loaded) (RSVD for PCI6X20)					
34h	PCI F4h, Link_Enh, byte 1, bits 7, 6, 5, 4 (RSVD for PCI6X20)					
35h	PCI F0h, PCI miscellaneous, byte 0, bits 5, 4, 2, 1, 0 (RSVD for PCI6X20)					
36h	PCI F1h, PCI miscellaneous, byte 1, bits 7, 3, 2, 1, 0 (RSVD for PCI6X20)					
37h	Reserved					
38h	Reserved (CardBus CIS pointer) (RSVD for PCI6X20)					
39h	Reserved					
3Ah	PCI ECh, PCI PHY control, bits 7, 3, 1 (RSVD for PCI6X20)					
3Bh	Flash media core function indicator (03h)					
3Ch	Number of bytes (05h)					
3Dh	PCI 2Ch, subsystem vendor ID, byte 0					
3Eh	PCI 2Dh, subsystem vendor ID, byte 1					
3Fh	PCI 2Eh, subsystem ID, byte 0					
40h	PCI 2Fh, subsystem ID, byte 1					
41h	PCI 4Ch, general control bits 6-0					
42h	End-of-list indicator (80h)					

11.3 EEPROM Programming Guide

This section provides a description of how to program the EEPROM or BIOS appropriately according to specific system needs. Each serial ROM offset (except the ones which are noted as reserved) from Table 13 is described sequentially.

11.3.1 Serial ROM Offset 00h – CardBus Function Indicator

To ensure proper EEPROM programming, this offset must be programmed as follows:

Serial ROM Offset 00h

Bit Number	7	6	5	4	3	2	1	0
Required value	0	0	0	0	0	0	0	0

11.3.2 Serial ROM Offset 01h – Number of Bytes in EEPROM Map

To ensure proper EEPROM programming, this offset must be programmed as follows:

Serial ROM Offset 01h

Bit Number	7	6	5	4	3	2	1	0
Required value	0	0	1	0	0	0	0	0

11.3.3 Serial ROM Offset 02h – Command Register (Function 0)

Serial ROM offset 02h corresponds to bits 8, 6-5, and 2-0 of the command register (PCI offset 04h) for function 0.

Serial ROM Offset 02h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	0	0	0	0

Bit	Field Name	Description
7	SERR_EN	System error ($\overline{\text{SERR}}$) enable. This bit controls the enable for the SERR driver on the PCI interface. $\overline{\text{SERR}}$ can be asserted after detecting an address parity error on the PCI bus. Both this bit and PERR_EN must be set for the PCIXX20 controller to report address parity errors.
6	PERR_EN	Parity error response enable. This bit controls the PCIXX20 response to parity errors through the $\overline{\text{PERR}}$ signal. Data parity errors are indicated by asserting $\overline{\text{PERR}}$, while address parity errors are indicated by asserting $\overline{\text{SERR}}$.
5	VGA_EN	VGA palette snoop. When set to 1, palette snooping is enabled (that is, the PCIXX20 controller does not respond to palette register writes and snoops the data). When the bit is 0, the PCIXX20 controller treats all palette accesses like all other accesses.
4	RSVD	Reserved
2	MAST_EN	Bus master control. This bit controls whether or not the PCIXX20 controller can act as a PCI bus initiator (master). The PCIXX20 controller can take control of the PCI bus only when this bit is set.
1	MEM_EN	Memory space enable. This bit controls whether or not the PCIXX20 controller can claim cycles in PCI memory space.
0	IO_EN	I/O space control. This bit controls whether or not the PCIXX20 controller can claim cycles in PCI I/O space.

11.3.4 Serial ROM Offset 03h – Command Register (Function 1)

Serial ROM offset 03h corresponds to bits 8, 6-5, and 2-0 of the command register (PCI offset 04h) for function 1.

Serial ROM Offset 03h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	0	0	0	0

Bit	Field Name	Description
7	SERR_EN	System error ($\overline{\text{SERR}}$) enable. This bit controls the enable for the SERR driver on the PCI interface. $\overline{\text{SERR}}$ can be asserted after detecting an address parity error on the PCI bus. Both this bit and PERR_EN must be set for the PCIXX20 controller to report address parity errors.
6	PERR_EN	Parity error response enable. This bit controls the PCIXX20 response to parity errors through the $\overline{\text{PERR}}$ signal. Data parity errors are indicated by asserting $\overline{\text{PERR}}$, while address parity errors are indicated by asserting $\overline{\text{SERR}}$.
5	VGA_EN	VGA palette snoop. When set to 1, palette snooping is enabled (that is, the PCIXX20 controller does not respond to palette register writes and snoops the data). When the bit is 0, the PCIXX20 controller treats all palette accesses like all other accesses.
4	RSVD	Reserved
2	MAST_EN	Bus master control. This bit controls whether or not the PCIXX20 controller can act as a PCI bus initiator (master). The PCIXX20 controller can take control of the PCI bus only when this bit is set.
1	MEM_EN	Memory space enable. This bit controls whether or not the PCIXX20 controller can claim cycles in PCI memory space.
0	IO_EN	I/O space control. This bit controls whether or not the PCIXX20 controller can claim cycles in PCI I/O space.

11.3.5 Serial ROM Offset 04h – Subsystem Vendor ID (Byte 0)

Serial ROM offset 04h corresponds to byte 0 of the subsystem vendor ID register (PCI offset 40h). To ensure proper device driver functionality, this offset must be set as follows:

Serial ROM Offset 04h

Bit Number	7	6	5	4	3	2	1	0
Recommended value	0	1	0	0	1	1	0	0

11.3.6 Serial ROM Offset 05h – Subsystem Vendor ID (Byte 1)

Serial ROM offset 05h corresponds to byte 1 of the subsystem vendor ID register (PCI offset 41h). To ensure proper device driver functionality, this offset must be set as follows:

Serial ROM Offset 05h

Bit Number	7	6	5	4	3	2	1	0
Recommended value	0	0	0	1	0	0	0	0

11.3.7 Serial ROM Offset 06h – Subsystem ID (Byte 0)

Serial ROM offset 06h corresponds to byte 0 of the subsystem ID register (PCI offset 42h). To ensure proper device driver functionality, this offset must be set as follows:

Serial ROM Offset 06h

Bit Number	7	6	5	4	3	2	1	0
Recommended value	0	1	0	0	1	0	0	0

11.3.8 Serial ROM Offset 07h – Subsystem ID (Byte 1)

Serial ROM offset 07h corresponds to byte 1 of the subsystem ID register (PCI offset 43h). To ensure proper device driver functionality, this offset must be set as follows:

Serial ROM Offset 07h

Bit Number	7	6	5	4	3	2	1	0
Required value	1	0	1	0	1	1	0	0

11.3.9 Serial ROM Offset 08h – PC Card Legacy-Mode Base Address Register (Byte 0)

Serial ROM offset 08h corresponds to byte 0 bits 7-1 of the PC Card legacy-mode base address register (PCI offset 44h).

Serial ROM Offset 08h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	0	0	0	0

11.3.10 Serial ROM Offset 09h – PC Card Legacy-Mode Base Address Register (Byte 1)

Serial ROM offset 09h corresponds to byte 1 of the PC Card legacy-mode base address register (PCI offset 45h).

Serial ROM Offset 09h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	0	0	0	0

11.3.11 Serial ROM Offset 0Ah – PC Card Legacy-Mode Base Address Register (Byte 2)

Serial ROM offset 0Ah corresponds to byte 2 of the PC Card legacy-mode base address register (PCI offset 46h).

Serial ROM Offset 0Ah

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	0	0	0	0

11.3.12 Serial ROM Offset 0Bh – PC Card Legacy-Mode Base Address Register (Byte 3)

Serial ROM offset 0Bh corresponds to byte 3 of the PC Card legacy-mode base address register (PCI offset 47h).

Serial ROM Offset 0Bh

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	0	0	0	0

11.3.13 Serial ROM Offset 0Ch – System Control Register Byte 0 (Function 0)

Serial ROM offset 0Ch corresponds to byte 0 bits 6-0 of the system control register (PCI offset 80h).

Serial ROM Offset 0Ch

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	1	1	0	0	0	0	0

Bit	Field Name	Description
7	RSVD	Reserved
6	PWRSAVINGS	Power saving mode enable. When this bit is set, the PCIXX20 controller consumes less power with no performance loss. This bit is shared between the two PCIXX20 CardBus functions.
5	SUBSYSRW	Subsystem ID and subsystem vendor ID, ExCA ID and revision register read/write enable. This bit also controls read/write for the function 3 subsystem ID register.
4	CB_DPAR	CardBus data parity $\overline{\text{SERR}}$ signaling enable
3	RSVD	Reserved
2	EXCAPOWER	ExCA power control bit. 0 = Enables 3.3 V 1 = Enables 5 V
1	KEEPCLK	When this bit is set, the PCIXX20 controller follows the $\overline{\text{CLKRUN}}$ protocol to maintain the system PCLK and the CCLK. This bit is global to the PCIXX20 functions. 0 = Allows system PCLK and CCLK clocks to stop. 1 = Never allows system PCLK or CCLK clock to stop.
0	RIMUX	$\overline{\text{PME}}$ / $\overline{\text{RI\_OUT}}$ select bit. When this bit is set, the PME signal is routed to the $\overline{\text{PME}}$ / $\overline{\text{RI\_OUT}}$ terminal. When this bit is cleared and bit 7 (RIENB) of the card control register is 1, the $\overline{\text{RI\_OUT}}$ signal is routed to the $\overline{\text{PME}}$ / $\overline{\text{RI\_OUT}}$ terminal. If this bit cleared and bit 7 (RIENB) of the card control register is 0, then the output is placed in a high-impedance state. 0 = $\overline{\text{RI\_OUT}}$ signal is routed to the $\overline{\text{PME}}$ / $\overline{\text{RI\_OUT}}$ terminal if bit 7 of the card control register is 1. 1 = $\overline{\text{PME}}$ signal is routed to the $\overline{\text{PME}}$ / $\overline{\text{RI\_OUT}}$ terminal.

11.3.14 Serial ROM Offset 0Dh – System Control Register Byte 0 (Function 1)

Serial ROM offset 0Dh corresponds to byte 0 bit 2 of the system control register (PCI offset 80h).

Serial ROM Offset 0Dh

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	1	1	0	0	0	0	0

Bit	Field Name	Description
7	RSVD	Reserved
6	PWRSAVINGS	Power saving mode enable. When this bit is set, the PCIXX20 controller consumes less power with no performance loss. This bit is shared between the two PCIXX20 CardBus functions.
5	SUBSYSRW	Subsystem ID and subsystem vendor ID, ExCA ID and revision register read/write enable. This bit also controls read/write for the function 3 subsystem ID register.
4	CB_DPAR	CardBus data parity $\overline{\text{SERR}}$ signaling enable
3	RSVD	Reserved
2	EXCAPOWER	ExCA power control bit. 0 = Enables 3.3 V 1 = Enables 5 V
1	KEEPCLK	When this bit is set, the PCIXX20 controller follows the $\overline{\text{CLKRUN}}$ protocol to maintain the system PCLK and the CCLK. This bit is global to the PCIXX20 functions. 0 = Allows system PCLK and CCLK clocks to stop. 1 = Never allows system PCLK or CCLK clock to stop.
0	RIMUX	$\overline{\text{PME}}$ / $\overline{\text{RI\_OUT}}$ select bit. When this bit is set, the PME signal is routed to the $\overline{\text{PME}}$ / $\overline{\text{RI\_OUT}}$ terminal. When this bit is cleared and bit 7 (RIENB) of the card control register is 1, the $\overline{\text{RI\_OUT}}$ signal is routed to the $\overline{\text{PME}}$ / $\overline{\text{RI\_OUT}}$ terminal. If this bit cleared and bit 7 (RIENB) of the card control register is 0, then the output is placed in a high-impedance state. 0 = $\overline{\text{RI\_OUT}}$ signal is routed to the $\overline{\text{PME}}$ / $\overline{\text{RI\_OUT}}$ terminal if bit 7 of the card control register is 1. 1 = $\overline{\text{PME}}$ signal is routed to the $\overline{\text{PME}}$ / $\overline{\text{RI\_OUT}}$ terminal.

11.3.15 Serial ROM Offset 0Eh – System Control Register Byte 1

Serial ROM offset 0Eh corresponds to byte 1 of the system control register (PCI offset 81h).

Serial ROM Offset 0Eh

Bit Number	7	6	5	4	3	2	1	0
Typical value	1	0	0	0	0	0	0	0

Bit	Field Name	Description
7	MRBURSTDN	Memory read burst enable downstream. When this bit is set, the PCIXX20 controller allows memory read transactions to burst downstream.
6	MRBBURSTUP	Memory read burst enable upstream. When this bit is set, the PCIXX20 controller allows memory read transactions to burst upstream.
5	SOCACTIVE	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.
4	RSVD	Reserved
3	PWRSTREAM	Power-stream-in-progress status bit. When set, this bit indicates that a power stream to the power switch is in progress and a power change has been requested. When this bit is cleared, it indicates that the power stream is complete.
2	DELAYUP	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.
1	DELAYDOWN	Power-down delay-in-progress status bit. When set, this bit indicates that a power-down stream has been set to the power switch, and proper power may not yet be stable. This bit is cleared when the power-down delay has expired.
0	INTERROGATE	Interrogation in progress. When set, this bit indicates an interrogation is in progress, and clears when the interrogation completes.

11.3.16 Serial ROM Offset 0Fh – System Control Register Byte 2

Serial ROM offset 0Fh corresponds to byte 2 of the system control register (PCI offset 82h). This offset is reserved and all 0s must be loaded.

Serial ROM Offset 0Fh

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	0	0	0	0

11.3.17 Serial ROM Offset 10h – System Control Register Byte 3

Serial ROM offset 10h corresponds to byte 3 of the system control register (PCI offset 83h).

Serial ROM Offset 10h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	1	1	1	0	0	0

Bit	Field Name	Description
7-6	SER_STEP	Serial input stepping. In the 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 = $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, $\overline{\text{INTC}}$, $\overline{\text{INTD}}$ signal in $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, $\overline{\text{INTC}}$, $\overline{\text{INTD}}$ slots 01 = $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, $\overline{\text{INTC}}$, $\overline{\text{INTD}}$ signal in $\overline{\text{INTB}}$, $\overline{\text{INTC}}$, $\overline{\text{INTD}}$, $\overline{\text{INTA}}$ slots 10 = $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, $\overline{\text{INTC}}$, $\overline{\text{INTD}}$ signal in $\overline{\text{INTC}}$, $\overline{\text{INTD}}$, $\overline{\text{INTA}}$, $\overline{\text{INTB}}$ slots 11 = $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, $\overline{\text{INTC}}$, $\overline{\text{INTD}}$ signal in $\overline{\text{INTD}}$, $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, $\overline{\text{INTC}}$ slots
5	INTRTIE	This bit ties $\overline{\text{INTA}}$ to $\overline{\text{INTB}}$ internally (to $\overline{\text{INTA}}$), and reports this through the interrupt pin register (PCI offset 3Dh). This bit has no effect on $\overline{\text{INTC}}$ or $\overline{\text{INTD}}$.
4	TIEALL	This bit ties $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, $\overline{\text{INTC}}$, and $\overline{\text{INTD}}$ internally (to $\overline{\text{INTA}}$), and reports this through the interrupt pin register (PCI offset 3Dh).
3	PSCCLK	P2C power switch clock. The PCIXX20 CLKCK signal clocks the serial interface power switch and the internal state machine. The default state for this bit is 1, allowing the internal oscillator to provide the clock signal. Clearing this bit requires the external clock source being provided to the CLOCK terminal.
2	SMIROUTE	SMI interrupt routing. This bit is shared between functions 0 and 1, and selects whether IRQ2 or CSC is signaled when a write occurs to power a PC card socket. 0 = SMI interrupt is signaled. 1 = SMI interrupt is not signaled.
1	RSVD	Reserved
0	SMIENB	SMI interrupt mode enable. When this bit is set, the SMI interrupt signaling generates an interrupt when a write to the socket power control occurs. This bit is shared between functions 0 and 1.

11.3.18 Serial ROM Offset 11h – Multifunction Routing Register Byte 0

Serial ROM offset 11h corresponds to byte 0 of the multifunction routing register (PCI offset 8Ch). This offset is intended to configure MFUNC0 and MFUNC1 terminals.

Bit	Field Name	Description
7-4	MFUNC1	MFUNC1 select. This bit controls the mapping of MFUNC1 as: 0000 = GPI1 0100 = OHCI_LED 1000 = CAUDPWM 1100 = LEDA1 0001 = GPO1 0101 = IRQ5 1001 = IRQ9 1101 = LEDA2 0010 = $\overline{\text{INTB}}$ 0110 = RSVD 1010 = IRQ10 1110 = $\overline{\text{GPE}}$ 0011 = IRQ3 0111 = RSVD 1011 = IRQ11 1111 = IRQ15
3-0	MFUNC0	MFUNC0 select. This bit controls the mapping of MFUNC0 as: 0000 = GPI0 0100 = IRQ4 1000 = CAUDPWM 1100 = LEDA1 0001 = GPO0 0101 = IRQ5 1001 = IRQ9 1101 = LEDA2 0010 = $\overline{\text{INTA}}$ 0110 = RSVD 1010 = IRQ10 1110 = $\overline{\text{GPE}}$ 0011 = IRQ3 0111 = RSVD 1011 = IRQ11 1111 = IRQ15

11.3.19 Serial ROM Offset 12h – Multifunction Routing Register Byte 1

Serial ROM offset 12h corresponds to byte 1 of the multifunction routing register (PCI offset 8Ch). This offset is intended to configure MFUNC2 and MFUNC3 terminals.

Bit	Field Name	Description
7-4	MFUNC3	MFUNC3 select. This bit controls the mapping of MFUNC3 as: 0000 = RSVD 0100 = IRQ4 1000 = IRQ8 1100 = IRQ12 0001 = IRQSER 0101 = IRQ5 1001 = IRQ9 1101 = IRQ13 0010 = IRQ2 0110 = IRQ6 1010 = IRQ10 1110 = IRQ14 0011 = IRQ3 0111 = IRQ7 1011 = IRQ11 1111 = IRQ15
3-0	MFUNC2	MFUNC2 select. This bit controls the mapping of MFUNC2 as: 0000 = GPI2 0100 = IRQ4 1000 = CAUDPWM 1100 = $\overline{\text{RI\_OUT}}$ 0001 = GPO2 0101 = IRQ5 1001 = IRQ9 1101 = TEST_MUX 0010 = $\overline{\text{PCREQ}}$ 0110 = RSVD 1010 = IRQ10 1110 = $\overline{\text{GPE}}$ 0011 = IRQ3 0111 = RSVD 1011 = $\overline{\text{INTC}}$ 1111 = IRQ7

11.3.20 Serial ROM Offset 13h – Multifunction Routing Register Byte 2

Serial ROM offset 13h corresponds to byte 2 of the multifunction routing register (PCI offset 8Ch). This offset is intended to configure MFUNC4 and MFUNC5 terminals.

Bit	Field Name	Description
7-4	MFUNC5	MFUNC5 select. This bit controls the mapping of MFUNC5 as: 0000 = GPI4 0100 = IRQ4 1000 = CAUDPWM 1100 = LEDA1 0001 = GPO4 0101 = IRQ5 1001 = IRQ9 1101 = LED_SKT 0010 = $\overline{\text{PCGNT}}$ 0110 = RSVD 1010 = IRQ10 1110 = $\overline{\text{GPE}}$ 0011 = IRQ3 0111 = RSVD 1011 = OHCI_LED 1111 = IRQ15
3-0	MFUNC4	MFUNC4 select. This bit controls the mapping of MFUNC4 as: 0000 = GPI3 0100 = IRQ4 1000 = CAUDPWM 1100 = $\overline{\text{RI\_OUT}}$ 0001 = GPO3 0101 = IRQ5 1001 = IRQ9 1101 = LED_SKT 0010 = $\overline{\text{LOCK}}$ 0110 = RSVD 1010 = $\overline{\text{INTD}}$ 1110 = $\overline{\text{GPE}}$ 0011 = IRQ3 0111 = RSVD 1011 = IRQ11 1111 = IRQ15

11.3.21 Serial ROM Offset 14h – Multifunction Routing Register Byte 3

Serial ROM offset 14h corresponds to byte 3 of the multifunction routing register (PCI offset 8Ch). This offset is intended to configure MFUNC6 terminal.

Bit	Field Name	Description
7-4	RSVD	Reserved
3-0	MFUNC6	MFUNC6 select. This bit controls the mapping of MFUNC6 as: 0000 = RSVD 0100 = IRQ4 1000 = IRQ8 1100 = IRQ12 0001 = $\overline{\text{CLKRUN}}$ 0101 = IRQ5 1001 = IRQ9 1101 = IRQ13 0010 = IRQ2 0110 = IRQ6 1010 = IRQ10 1110 = IRQ14 0011 = IRQ3 0111 = IRQ7 1011 = IRQ11 1111 = IRQ15

11.3.22 Serial ROM Offset 15h – Retry Status Register

Serial ROM offset 15h corresponds to bits 7 and 6 of the retry status register (PCI offset 90h).

Serial ROM Offset 15h

Bit Number	7	6	5	4	3	2	1	0
Typical value	1	1	0	0	0	0	0	0

Bit	Field Name	Description
7	PCIRETRY	PCI retry time-out counter enable. This bit is encoded as: 0 = PCI retry counter disabled 1 = PCI retry counter enabled
6	CBRETRY	CardBus retry time-out counter enable. This bit is encoded as: 0 = CardBus retry counter disabled 1 = CardBus retry counter enabled
5-0	RSVD	Reserved

11.3.23 Serial ROM Offset 16h – Card Control Register

Serial ROM offset 16h corresponds to bit 7 of the card control register (PCI offset 91h).

Serial ROM Offset 16h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	0	0	0	0

Bit	Field Name	Description
7	RIENB	Ring indicate enable. When this bit is set, the $\overline{\text{RI\_OUT}}$ output is enabled.
6-0	RSVD	Reserved

11.3.24 Serial ROM Offset 17h – Device Control Register

Serial ROM offset 17h corresponds to bits 6-5, 3-0 of the device control register (PCI offset 92h).

Serial ROM Offset 17h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	1	1	0	0	1	0	0

Bit	Field Name	Description
7	SKTPWR_LOCK	Socket power lock bit. When this bit is set, 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	3-V socket capable force bit 0 = Not 3-V capable 1 = 3-V capable
3	TEST	TI test bit. Write only 0 to this bit.
2-1	INTMODE	Interrupt mode. These bits select the interrupt signaling mode. The interrupt mode bits are encoded: 00 = Parallel PCI interrupts only 01 = Reserved 10 = IRQ serialized interrupts and parallel PCI interrupts $\overline{\text{INTA}}$, $\overline{\text{INTB}}$, $\overline{\text{INTC}}$, $\overline{\text{INTD}}$ 11 = IRQ and PCI serialized interrupts
0	RSVD	Reserved

11.3.25 Serial ROM Offset 18h – Diagnostic Register

Serial ROM offset 18h corresponds to bits 7-6 and 4-0 of the diagnostic register (PCI offset 93h).

Serial ROM Offset 18h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	1	1	0	0	0	0	0

Bit	Field Name	Description
7	TRUE_VAL	This bit is decoded as follows: 0 = Reads true values in PCI vendor ID and PCI device ID registers 1 = Returns all 1s to reads from the PCI vendor ID and PCI device ID registers
6-5	RSVD	Reserved
4	DIAG4	Diagnostic RETRY_DIS. Delayed transaction disable.
3	DIAG3	Diagnostic RETRY_EXT. Extends the latency from 16 to 64.
2	DIAG2	Diagnostic DISCARD_TIM_SEL_CB. Set = 2^{10}, reset = 2^{15}
1	DIAG1	Diagnostic DISCARD_TIM_SEL_PCI. Set = 2^{10}, reset = 2^{15}
0	RSVD	Reserved

11.3.26 Serial ROM Offset 19h – Power Management Capabilities Register (Function 0)

Serial ROM offset 19h corresponds to bit 15 of the power management capabilities register (PCI offset A2h) for function 0.

Serial ROM Offset 19h

Bit Number	7	6	5	4	3	2	1	0
Typical value	1	0	0	0	0	0	0	0

Bit	Field Name	Description
7	PMESTAT	PME status. This bit is set when the CardBus function would normally assert the $\overline{\text{PME}}$ signal, independent of the state of the PME_EN bit. This bit is cleared by a write back of 1, and this also clears the $\overline{\text{PME}}$ signal if $\overline{\text{PME}}$ was asserted by this function. Writing a 0 to this bit has no effect.
6-0	RSVD	Reserved

11.3.27 Serial ROM Offset 1Ah – Power Management Capabilities Register (Function 1)

Serial ROM offset 1Ah corresponds to bit 15 of the power management capabilities register (PCI offset A2h) for function 1.

Serial ROM Offset 1Ah

Bit Number	7	6	5	4	3	2	1	0
Typical value	1	0	0	0	0	0	0	0

Bit	Field Name	Description
7	PMESTAT	PME status. This bit is set when the CardBus function would normally assert the $\overline{\text{PME}}$ signal, independent of the state of the PME_EN bit. This bit is cleared by a write back of 1, and this also clears the $\overline{\text{PME}}$ signal if $\overline{\text{PME}}$ was asserted by this function. Writing a 0 to this bit has no effect.
6-0	RSVD	Reserved

11.3.28 Serial ROM Offset 1Dh – ExCA Identification and Revision Register

Serial ROM offset 1Dh corresponds to bits 7-0 of the ExCA identification and revision register (CardBus socket address offset 800h).

Serial ROM Offset 1Dh

Bit Number	7	6	5	4	3	2	1	0
Typical value	1	0	0	0	0	1	0	0

Bit	Field Name	Description
7-6	IFTYPE	Interface type. These bits, which are hardwired as 10b, identify the 16-bit PC Card support provided by the PCIXX20 controller. The PCIXX20 controller supports both I/O and memory 16-bit PC Cards.
5-4	RSVD	Reserved
3-0	365REV	82365SL-DF revision. This field stores the Intel 82365SL-DF revision supported by the PCIXX20 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.

11.3.29 Serial ROM Offset 1Eh – General Control Register

Serial ROM offset 1Eh corresponds to byte 0 bits 5-0 of the general control register (PCI offset 86h).

Serial ROM Offset 1Eh

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	0	0	1	1

Bit	Field Name	Description
7-6	RSVD	Reserved
5	DISABLE_FM	When this bit is set, the flash media function is completely nonaccessible and nonfunctional.
4	DISABLE_SKTB	When this bit is set, CardBus socket B (function 1) is completely nonaccessible and nonfunctional.
3	DISABLE_OHCI	When set, the OHCI 1394 controller function is completely nonaccessible and nonfunctional.
2	RSVD	Reserved
1-0	ARB_CTRL	Controls top level arbitration: 00 = 1394 OHCI priority 10 = Flash media priority 01 = CardBus priority 11 = Fair round robin

11.3.30 Serial ROM Offset 1Fh – General Control Register

Serial ROM offset 1Fh corresponds to byte 1 bits 4-2 of the general control register (PCI offset 87h).

Serial ROM Offset 1Fh

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	0	0	0	0

Bit	Field Name	Description
7-5	RSVD	Reserved
4	IO_LIMIT_SEL	When this bit is set, bit 0 in the I/O limit register (PCI offsets 30h and 38h) for both CardBus functions is set. 0 = Bit 0 in the I/O limit register is 0. 1 = Bit 0 in the I/O limit register is 1.
3	IO_BASE_SEL	When this bit is set, bit 0 in the I/O base register (PCI offsets 2Ch and 34h) for both CardBus functions is set. 0 = Bit 0 in the I/O base register is 0. 1 = Bit 0 in the I/O base register is 1.
2	12V_SW_SEL	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. 1 = A 12-V capable power switch (TPS2226) is used.
1-0	RSVD	Reserved

11.3.31 Serial ROM Offset 20h – General-Purpose Event Enable Register

Serial ROM offset 20h corresponds to bits 7-6, and 4-0 of the general-purpose event enable register (PCI offset 89h).

Serial ROM Offset 20h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	0	0	0	0

Bit	Field Name	Description
7	PWR_EN	Power change $\overline{\text{GPE}}$ enable. When this bit is set, $\overline{\text{GPE}}$ is signaled on PWR_STS events.
6	VPP12_EN	12V VPP $\overline{\text{GPE}}$ enable. When this bit is set, $\overline{\text{GPE}}$ is signaled on VPP12_STS events.
5	RSVD	Reserved
4	GP4_EN	GPI4 $\overline{\text{GPE}}$ enable. When this bit is set, $\overline{\text{GPE}}$ is signaled on GP4_STS events.
3	GP3_EN	GPI3 $\overline{\text{GPE}}$ enable. When this bit is set, $\overline{\text{GPE}}$ is signaled on GP3_STS events.
2	GP2_EN	GPI2 $\overline{\text{GPE}}$ enable. When this bit is set, $\overline{\text{GPE}}$ is signaled on GP2_STS events.
1	GP1_EN	GPI1 $\overline{\text{GPE}}$ enable. When this bit is set, $\overline{\text{GPE}}$ is signaled on GP1_STS events.
0	GP0_EN	GPI0 $\overline{\text{GPE}}$ enable. When this bit is set, $\overline{\text{GPE}}$ is signaled on GP0_STS events.

11.3.32 Serial ROM Offset 21h – General-Purpose Output Register

Serial ROM offset 21h corresponds to bits 4-0 of the general-purpose output register (PCI offset 8Bh).

Serial ROM Offset 21h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	0	0	0	0

Bit	Field Name	Description
7-5	RSVD	Reserved
4	GPO4_DATA	This bit represents the logical value of the data driven to GPO4.
3	GPO3_DATA	This bit represents the logical value of the data driven to GPO3.
2	GPO2_DATA	This bit represents the logical value of the data driven to GPO2.
1	GPO1_DATA	This bit represents the logical value of the data driven to GPO1.
0	GPO0_DATA	This bit represents the logical value of the data driven to GPO0.

11.3.33 Serial ROM Offset 22h – 1394 OHCI Function Indicator

Serial ROM offset 22h corresponds to the 1394 OHCI function indicator. This serial ROM offset is set to 02h for the PCI7X20 controller and RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 22h

Bit Number	7	6	5	4	3	2	1	0
Recommended value	0	0	0	0	0	0	1	0

11.3.34 Serial ROM Offset 23h – Number of Bytes in the 1394 EEPROM Map

Serial ROM offset 23h corresponds to the number of bytes in the 1394 port within the EEPROM map. This serial ROM offset is set to 17h for the PCI7X20 controller and RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 23h

Bit Number	7	6	5	4	3	2	1	0
Recommended value	0	0	0	1	0	1	1	1

11.3.35 Serial ROM Offset 24h – Minimum Grant and Maximum Latency Register

Serial ROM offset 24h corresponds to byte 0 bits 3-0, and byte 1 bits 3-0 of the minimum grant and maximum latency register (PCI offset 3Eh) for function 2. This serial ROM offset is RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 24h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	1	0	0	0	0	1	1

Bit	Field Name	Description
7-4	MAX_LAT	Maximum latency. The contents of this field may be used to host BIOS to assign an arbitration priority level to the PCI7X20 controller. The default for this register indicates that the PCI7X20 controller may need to access the PCI bus as often as every 0.25 µs; thus, an extremely high priority level is requested.
3-0	MIN_GNT	Minimum grant. The contents of this field may be used by host BIOS to assign a latency timer register value to the PCI7X20 controller. The default for this register indicates that the PCI7X20 controller may need to sustain burst transfers for nearly 64 µs and thus request a large value be programmed in bits 15-8 of the PCI7X20 latency timer and class cache line size register at PCI offset 0Ch.

11.3.36 Serial ROM Offset 25h Subsystem Identification Register

Serial ROM offset 25h corresponds to byte 0 of the subsystem identification register (PCI offset 2Ch) for function 2. This serial ROM offset is set to 2Ch for the PCI7X20 controller and RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 25h

Bit Number	7	6	5	4	3	2	1	0
Required value	0	1	0	0	1	1	0	0

Bit	Field Name	Description
7-0	OHCI_SSVID	Subsystem vendor ID. This field indicates the first byte of the subsystem vendor ID.

11.3.37 Serial ROM Offset 26h Subsystem Identification Register

Serial ROM offset 26h corresponds to byte 1 of the subsystem identification register (PCI offset 2Ch) for function 2. This serial ROM offset is set to 10h for the PCI7X20 controller and RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 26h

Bit Number	7	6	5	4	3	2	1	0
Recommended value	0	0	0	1	0	0	0	0

Bit	Field Name	Description
7-0	OHCI_SSVID	Subsystem vendor ID. This field indicates the second byte of the subsystem vendor ID.

11.3.38 Serial ROM Offset 27h Subsystem Identification Register (Byte 0)

Serial ROM offset 27h corresponds to byte 2 of the subsystem identification register (PCI offset 2Ch) for function 2. This serial ROM offset is set to 23h for the PCI7X20 controller and RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 27h

Bit Number	7	6	5	4	3	2	1	0
Recommended value	0	0	1	0	0	0	1	1

Bit	Field Name	Description
7-0	OHCI_SSID	Subsystem device ID. This field indicates the first byte of the subsystem device ID.

11.3.39 Serial ROM Offset 28h Subsystem Identification Register (Byte 1)

Serial ROM offset 28h corresponds to byte 3 of the subsystem identification register (PCI offset 2Ch) for function 2. This serial ROM offset is set to 80h for the PCI7X20 controller and RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 28h

Bit Number	7	6	5	4	3	2	1	0
Recommended value	1	0	0	0	0	0	0	0

Bit	Field Name	Description
7-0	OHCI_SSID	Subsystem device ID. This field indicates the second byte of the subsystem device ID.

11.3.40 Serial ROM Offset 29h Link Enchancement Register and Host Controller Control Register

Serial ROM offset 29h corresponds to byte 0 bits 7, 2, and 1 of the link enhancement register (PCI offset F4h) for function 2, and bit 23 of the host controller control register (OHCI offset 50h). This serial ROM offset is RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 29h

Bit Number	7	6	5	4	3	2	1	0
Typical value	1	1	0	0	0	0	1	0

Bit	Field Name	Description
7	enab_unfair	Enable asynchronous priority request. OHCI-Lynx™ compatible. Setting this bit to 1 enables the link to respond to request with the priority arbitration.
6	programPhyEnable	This bit informs upper-level software than lower-level software has consistently configured the IEEE 1394a-2000 enhancements in the link and PHY layers. When this bit is set, generic software such as the OHCI driver is responsible for configuring IEEE 1394a-2000 enhancements in the PHY layer and bit 22 (aPhyEnhanceEnable)of host controller control register (OHCI offset 50h). When this bit is cleared, the generic software may not modify the IEEE 1394a-2000 enhancements in the PHY layer and cannot interpret the setting of the aPhyEnhanceEnable bit.
5-2	RSVD	Reserved
1	enab_accel	Enable acceleration enhancements. OHCI-Lynx™ compatible. When this bit is set, the PHY layer is notified that the link supports the IEEE Std 1394a-2000 acceleration enhancements, that is, ack-accelerated, fly-by concatenation, etc.
0	RSVD	Reserved

11.3.41 Serial ROM Offset 2Ah Mini-ROM Offset in EEPROM Map

Serial ROM offset 2Ah is used to indicate the mini-ROM offset into the EEPROM map. A value of 00h in this offset implies that there is no mini-ROM. This serial ROM offset is set to 20h for the PCI7X20 controller and RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 2Ah

Bit Number	7	6	5	4	3	2	1	0
Required value	0	0	1	0	0	0	0	0

11.3.42 Serial ROM Offset 2Bh GUID High Register (Byte 0)

Serial ROM offset 2Bh corresponds to byte 0 of the GUID high register (OHCI offset 24h). This serial ROM offset is RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 2Bh

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	1	0	1	0	1	1	0

11.3.43 Serial ROM Offset 2Ch GUID High Register (Byte 1)

Serial ROM offset 2Ch corresponds to byte 1 of the GUID high register (OHCI offset 24h). This serial ROM offset is RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 2Ch

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	1	0	1	0	0	0

11.3.44 Serial ROM Offset 2Dh GUID High Register (Byte 2)

Serial ROM offset 2Dh corresponds to byte 2 of the GUID high register (OHCI offset 24h). This serial ROM offset is RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 2Dh

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	0	0	0	0

11.3.45 Serial ROM Offset 2Eh GUID High Register (Byte 3)

Serial ROM offset 2Eh corresponds to byte 3 of the GUID high register (OHCI offset 24h). This serial ROM offset is RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 2Eh

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	1	0	0	0

11.3.46 Serial ROM Offset 2Fh GUID Low Register (Byte 0)

Serial ROM offset 2Fh corresponds to byte 0 of the GUID low register (OHCI offset 28h). This serial ROM offset is RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 2Fh

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	1	0	0	0

11.3.47 Serial ROM Offset 30h GUID Low Register (Byte 1)

Serial ROM offset 30h corresponds to byte 1 of the GUID low register (OHCI offset 28h). This serial ROM offset is RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 30h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	1	0	0	0

11.3.48 Serial ROM Offset 31h GUID Low Register (Byte 2)

Serial ROM offset 31h corresponds to byte 2 of the GUID low register (OHCI offset 28h). This serial ROM offset is RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 31h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	1	0	0	0

11.3.49 Serial ROM Offset 32h GUID Low Register (Byte 3)

Serial ROM offset 32h corresponds to byte 3 of the GUID low register (OHCI offset 28h). This serial ROM offset is RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 32h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	1	0	0	0

11.3.50 Serial ROM Offset 34h Link Enhancement Control Register

Serial ROM offset 34h corresponds to byte 1 bits 7-4 of the link enhancement control register (PCI offset F4h) for function 2. This serial ROM offset is RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 34h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	1	0	0	0	0

Bit	Field Name	Description
7	dis_at_pipeline	Disable AT pipelining. When this bit is set, out-of-order AT pipelining is disabled.
6	RSVD	Reserved
5-4	atx_thresh	This field sets the initial AT threshold value, which is used until the AT FIFO is underrun. When the PCI7X20 controller retries the packet, it uses 2K-byte threshold, resulting in a store-and-forward operation. 00 = Threshold – 2K bytes result in a store-and-forward operation 01 = Threshold – 1.7K bytes 10 = Threshold – 1K bytes 11 = Threshold – 512 bytes These bits fine-tune the asynchronous transmit threshold. For most applications the 1.7K-byte threshold is optimal. Changing this value may increase or decrease the 1394 latency depending on the average PCI bus latency. Setting the AT threshold to 1.7K, 1K, or 512 bytes results in data being transmitted at these thresholds or when an entire packet has been checked into the FIFO. If the packet to be transmitted is larger than the AT threshold, then the remaining data must be received before the AT FIFO is emptied; otherwise, an underrun condition occurs, resulting in a packet error at the receiving node. As a result, the link then commences a store-and-forward operation. Wait until it has the complete packet in the FIFO before retransmitting it on the second attempt to ensure delivery. An AT threshold of 2K results in a store-and-forward operation, which means that asynchronous data is not transmitted until an end-of-packet token is received. Restated, setting the AT threshold to 2K results in only complete packets being transmitted.
3-0	RSVD	Reserved

11.3.51 Serial ROM Offset 35h PCI Miscellaneous Configuration Register (Byte 0)

Serial ROM offset 35h corresponds to byte 0 bits 5-0 of the PCI miscellaneous configuration register (PCI offset F0h) for function 2. This serial ROM offset is RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 35h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	0	0	0	0

Bit	Field Name	Description
7-5	RSVD	Reserved
4	DIS_TGT_ABT	This bit provides OHCI-Lynx™ compatible target abort signaling. When this bit is set, it enables the no-target-abort mode, in which the PCI7X20 controller reruns indeterminate data instead of signaling target abort. The PCI7X20 LLC is divided into the PCLK and SCLK domains. If software tries to access registers in the link that are not active because the SCLK is disabled, then a target abort is issued by the link. On some systems, this can cause a problem resulting in a fatal system error. Enabling this bit allows the link to respond to these types of requests by returning FFh.
3	GP2IIC	When this bit is set, the GPIO3 and GPIO2 signals are internally routed to the SCL and SDA, respectively. The GPIO3 and GPIO2 terminals are also placed in the high-impedance state.
2-1	RSVD	Reserved
0	KEEP_PCLK	When this bit is set, the PCI clock is always kept running through the $\overline{\text{CLKRUN}}$ protocol. When this bit is cleared, the PCI clock can be stopped using $\overline{\text{CLKRUN}}$ on the MFUNC terminals.

11.3.52 Serial ROM Offset 36h PCI Miscellaneous Configuration Register (Byte 1)

Serial ROM offset 36h corresponds to byte 1 bits 7 and 3-0 of the PCI miscellaneous configuration register (PCI offset F0h) for function 2. This serial ROM offset is RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 36h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	0	0	0	0

Bit	Field Name	Description
7	PME_D3COLD	$\overline{\text{PME}}$ support from $D3_{cold}$. This bit programs bit 15 (PME_D3COLD) in the power management capabilities register at offset 46h in the PCI configuration space.
6-4	RSVD	Reserved
3	PCI2_3_EN	The PCI7X20 1394 OHCI function always conforms to the PCI2.3 specification. Therefore, this bit is tied to 1.
2	ignore_mstrintEna_for_pme	Ignore IntMask.masterIntEnable bit for $\overline{\text{PME}}$ generation. When this bit is set, it causes the $\overline{\text{PME}}$ generation behavior to be changed (please refer to Section 3.9.1 of the PCI7X20 data manual). It also causes bit 26 of the OHCI vendor ID register (OHCI offset 40h) to read 1. 0 = PME behavior generated from unmasked interrupt bits and IntMask.masterIntEnable bit. 1 = PME generation does not depend on the value of IntMask.masterIntEnable.
1-0	MR_ENHANCE	This field selects the read command behavior of the PCI master for read transactions of greater than two data phases. For read transactions of one or two data phases, a memory read command is used. 00 = Memory read line 01 = Memory read 10 = Memory read multiple 11 = Reserved

11.3.53 Serial ROM Offset 3Ah PCI PHY Layer Control Register

Serial ROM offset 3Ah corresponds to byte 0 bits 7, 3, and 1 of the PCI PHY layer control register (OHCI offset ECh). This serial ROM offset is RSVD for the PCI6X20 controller since the PCI6X20 controller does not have a 1394 function.

Serial ROM Offset 3Ah

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	1	0	0	0

Bit	Field Name	Description
7-0	wrData	This field is the data to be written to a PHY register and is ignored for reads.

11.3.54 Serial ROM Offset 3Bh Flash Media Core Function Indicator

Serial ROM offset 3Bh corresponds to the flash media core function indicator in the EEPROM map.

Serial ROM Offset 3Bh

Bit Number	7	6	5	4	3	2	1	0
Required value	0	0	0	0	0	0	1	1

11.3.55 Serial ROM Offset 3Ch Number of Bytes in EEPROM Map (FlashMedia)

Serial ROM offset 3Ch corresponds to the number of bytes of flash media core function in the EEPROM map. To ensure proper device driver functionality, this offset must be set as follows:

Serial ROM Offset 3Ch

Bit Number	7	6	5	4	3	2	1	0
Recommended value	0	0	0	0	0	1	0	1

11.3.56 Serial ROM Offset 3Dh Subsystem Vendor Identification Register (Byte 0)

Serial ROM offset 3Dh corresponds to byte 0 of the subsystem vendor identification register (PCI offset 2Ch) of function 3. To ensure proper device driver functionality, this offset must be set as follows:

Serial ROM Offset 3Dh

Bit Number	7	6	5	4	3	2	1	0
Recommended value	0	1	1	1	1	0	0	0

11.3.57 Serial ROM Offset 3Eh Subsystem Vendor Identification Register (Byte 1)

Serial ROM offset 3Eh corresponds to byte 1 of the subsystem vendor identification register (PCI offset 2Ch) of function 3. To ensure proper device driver functionality, this offset must be set as follows:

Serial ROM Offset 3Eh

Bit Number	7	6	5	4	3	2	1	0
Recommended value	0	1	0	1	0	1	1	0

11.3.58 Serial ROM Offset 3Fh Subsystem Identification Register (Byte 0)

Serial ROM offset 3Fh corresponds to byte 0 of the subsystem identification register (PCI offset 2Eh) of function 3. To ensure proper device driver functionality, this offset must be set as follows:

Serial ROM Offset 3Fh

Bit Number	7	6	5	4	3	2	1	0
Recommended value	0	0	1	1	0	1	0	0

11.3.59 Serial ROM Offset 40h Subsystem Vendor Identification Register (Byte 1)

Serial ROM offset 40h corresponds to byte 1 of the subsystem identification register (PCI offset 2Eh) of function 3. To ensure proper device driver functionality, this offset must be set as follows:

Serial ROM Offset 40h

Bit Number	7	6	5	4	3	2	1	0
Recommended value	0	0	0	1	0	0	1	0

11.3.60 Serial ROM Offset 41h General Control Register (Function 3)

Serial ROM offset 41h corresponds to bits 6-0 of the general control register (PCI offset 4Ch) for function 3.

Serial ROM Offset 41h

Bit Number	7	6	5	4	3	2	1	0
Typical value	0	0	0	0	0	0	0	0

Bit	Field Name	Description
6-5	INT_SEL	Interrupt select. These bits are used to program the INTPIN register and set which interrupt output is used. This field is ignored if one of the USE_INTx terminals is asserted. 00 = $\overline{\text{INTA}}$ 01 = $\overline{\text{INTB}}$ 10 = $\overline{\text{INTC}}$ 11 = $\overline{\text{INTD}}$
4	D3_COLD	$D3_{cold}$ $\overline{\text{PME}}$ support. This bit sets and clears the $D3_{cold}$ $\overline{\text{PME}}$ support bit in the power management capabilities register.
3-2	RSVD	Reserved
1	MMC_SD_DIS	MMC/SD disable. Setting this bit disables support for MMC/SD cards. The flash media controller reports a MMC/SD card as an unsupported card if this bit is set. Also, if this bit is set, all of the SD_SUPPORT bits in the socket enumeration registers are 0.
0	MS_DIS	Memory Stick disable. Setting this bit disables support for Memory Stick cards. The flash media controller reports a Memory Stick card as an unsupported card if this bit is set. Also, if this bit is set, all of the MS_SUPPORT bits in the socket enumeration registers are 0.

11.3.61 Serial ROM Offset 42h End-of-List Indicator

Serial ROM offset 42h corresponds to an end-of-list indicator denoting the end of EEPROM map. This offset has a recommended value of 80h.

Serial ROM Offset 40h

Bit Number	7	6	5	4	3	2	1	0
Recommended value	1	0	0	0	0	0	0	0

11.4 Socket Power Lock

Socket power can be protected from software control in the $D3_{hot}$ state. It can be accomplished with the SKTPWR_LOCK bit (device control register, PCI offset 92h, bit 7).

11.5 V_{CC} Protection

The VCCPROT bit (system control register, PCI offset 80h, bit 21) controls V_{CC} protection for 16-bit cards. This feature protects against applying the wrong (higher) V_{CC} to the 16-bit card. If a 3.3-V-only card is inserted, then it protects against applying 5 V to the card. Default is 0 (V_{CC} protection enabled).

11.6 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

12 Power Management Considerations

12.1 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 controller uses $\overline{PME}$ 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 generating PMEs, reporting PME status, and enabling 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 ($\overline{GRST}$) is asserted, the PCXX20 controller is completely nonfunctional and is in its default state. Output buffers are in a high-impedance state and internal registers are reset. The result of PCI reset ($\overline{PRST}$) being asserted is dependent on whether $\overline{PME}$ is enabled or not. When $\overline{PRST}$ is asserted with the function not enabled for $\overline{PME}$, it causes the PCIXX20 controller to place all output buffers in a high-impedance state and reset all internal registers except for those considered $\overline{GRST}$-only registers. If $\overline{PME}$ is enabled for the socket, then the PCIXX20 controller maintains its $\overline{PME}$ context registers.

According to the *PCI Bus Power Management Interface Specification* for PCI to CardBus bridges, a controller returning to D0 from $D3_{hot}$ is required to assert an internal reset. The PCI reset may or may not be asserted by the system. However, for a controller returning to D0 from $D3_{cold}$, $\overline{PRST}$ must be asserted by the system.

For a wake from $D3_{cold}$, the controller needs to save its $\overline{PME}$ context in order for software to determine the source of the wake-up event. This is accomplished using $\overline{PME}$ enable and saving the $\overline{PME}$ context registers. However, the controller must also maintain certain registers that are normally configured by BIOS at boot time. This is accomplished using $\overline{GRST}$ and the $\overline{GRST}$-only registers. This allows a system to be in a low power state and resumed quickly without needing BIOS to reprogram the controller.

The sequences of events at power up are that $\overline{GRST}$ must be asserted after power is supplied to the PCIXX20 controller. $\overline{GRST}$ can be deasserted asynchronously, then 100 µs after PCLK is stable, $\overline{PRST}$ can be deasserted. At this point, $\overline{GRST}$ stays deasserted until the system completely cycles power and reboots. Now the system can put the PCIXX20 controller into a lower power state and may or may not assert $\overline{PRST}$.

The PCIXX20 controller does not require a PCI clock to generate a $\overline{PME}$ signal. However, it does require a voltage source such as Vaux to be supplied and the pullup resistor on $\overline{PME}$ must also be connected to Vaux. In addition, the V_{CCP} pins and power switch must also have power in order to wake from a card. Vaux is limited to 200 mA for the socket.

12.1.1 $\overline{GRST}$ -Only Registers

Global reset places all registers in their default state regardless of the state of the 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.

The CardBus global reset-only bits (functions 0 and 1) are:

- Status register (PCI offset 06h) bits 15-11, 8
- Secondary status register (PCI offset 16h) bits 15-11, 8
- 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 31-0
- System control register (PCI offset 80h) bits 31-24, 22-13, 11-8, 6-0
- MC_CD debounce register (PCI offset 84h): bits 7-0
- General control register (PCI offset 86h): bits 13-10, 7, 5-3, 1-0
- General-purpose event status register (PCI offset 88h) bits 7-6, 4-0
- General-purpose event enable register (PCI offset 89h) bits 7-6, 4-0
- General-purpose output (PCI offset 8Bh) bits 4-0
- Multifunction routing status register (PCI offset 8Ch) bits 31-0
- Retry status register (PCI offset 90h) bits 7-5, 3,1
- Card control register (PCI offset 91h) bits 7, 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
- Power management CSR register (PCI offset A4h): bits 15, 8
- 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/status register (PCI offset B3h) bits 7, 3-0
- ExCA identification and revision register (ExCA offset 800h/840h) bits 7-0
- ExCA global control register (ExCA offset 81Eh/85Eh) bits 2-0

- CardBus socket power management register (CardBus 20h): bits 25-24

The 1394 global reset-only bits (function 2) are:

- Subsystem vendor ID register (PCI offset 2Ch): bits 15-0
- Subsystem ID register (PCI offset 2Eh): bits 15-0
- Minimum grant and maximum latency register (PCI offset 3Eh): bits 15-0
- Power management control and status register (PCI offset 48h): bits 15, 8, 1, 0
- Miscellaneous configuration register (PCI offset F0h): bits 15, 11-8, 5-0
- Link enhancement control register (PCI offset F4h): bits 15-12, 10, 8-7, 2-1
- Bus options register (OHCI offset 20h): bits 15-12
- GUID high register (OHCI offset 24h): bits 31-0
- GUID low register (OHCI offset 28h): bits 31-0
- Host controller control register (OHCI offset 50h and 54h): bit 23
- Link control register (OHCI offset E0h and E4h): bit 6

The Flash Media global reset-only bits (function 3) are:

- Subsystem vendor ID register (PCI offset 2Ch): bits 15-0
- Subsystem ID register (PCI offset 2Eh): bits 15-0
- Power management control status register (PCI offset 48h): bits 15, 8, 1-0
- General control (PCI offset 4Ch): bits 6-5, 4, 2-0
- PLL control register (PCI offset 54h): bits 31-0

12.1.2 $\overline{\text{PME}}$ Context Registers

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

- Bridge control register (PCI offset 3Eh) bit 6
- System control register (PCI offset 80h) bits 10, 9, 8
- Power-management control/status register (PCI offset A4h) bits 15
- ExCA power control register (ExCA offset 802h/842h) bit 5 (82365SL mode only), 7, 4-3, 1-0
- ExCA interrupt and general control register (ExCA offset 803h/843h) bits 6-5
- ExCA card status change register (ExCA offset 804h/844h) bits 3-0
- ExCA card status changed interrupt configuration register (ExCA offset 805h/845h) bits 3-0
- ExCA card detect and general control register (ExCA 816h/856h): bits 7-6
- Socket event register (CardBus offset 00h) bits 3-0
- Socket mask register (CardBus offset 04h) bits 3-0
- Socket present state register (CardBus offset 08h) bits 13-7, 5-1
- CardBus socket control register (CardBus offset 10h) bits 6-4, 2-0

12.2 $\overline{\text{PME}}$ / $\overline{\text{RI\_OUT}}$ Behavior

$\overline{\text{PME}}$ and $\overline{\text{RI\_OUT}}$ are very important for power management. The $\overline{\text{PME}}$ signal is useful for PCI power management systems. The $\overline{\text{RI\_OUT}}$ (ring indicate out) signal is used for legacy power management systems. $\overline{\text{PME}}$ and $\overline{\text{RI\_OUT}}$ are multiplexed on the same pin. The PCIXX20 controller can also provide $\overline{\text{RI\_OUT}}$ on the multifunction terminals.

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

Table 14. CardBus CSTSCHG and Wake-Up Signals Truth Table

RINGEN	RIMUX	RIENB	PME_EN	PME_STAT	$\overline{\text{RI\_OUT}}$ / $\overline{\text{PME}}$	MFUNC7
0	0	0	0	Latched	-	-
0	0	0	1	Latched	Latched CSTSCHG	-
0	0	1	0	Latched	-	-
0	0	1	1	Latched	-	-
0	1	0	0	Latched	-	-
0	1	0	1	Latched	Latched CSTSCHG	-
0	1	1	0	Latched	-	-
0	1	1	1	Latched	Latched CSTSCHG	-
1	0	0	0	Latched	-	-
1	0	0	1	Latched	Latched CSTSCHG	-
1	0	1	0	Latched	CSTSCHG	CSTSCHG
1	0	1	1	Latched	CSTSCHG	CSTSCHG
1	1	0	0	Latched	-	-
1	1	0	1	Latched	Latched CSTSCHG	-
1	1	1	0	Latched	-	CSTSCHG
1	1	1	1	Latched	Latched CSTSCHG	CSTSCHG

Table 15. 16 Bit Card RI/STSCHG and Wake-Up Signals Truth Table

RINGEN	RIMUX	RIENB	PME_EN	PME_STAT	$\overline{\text{RI\_OUT}}$ / $\overline{\text{PME}}$	MFUNC7
0	0	0	0	Latched	-	-
1	0	0	0	Latched	-	-
1	0	0	1	Latched	Latched RI	-
1	0	1	0	Latched	RI	RI
1	0	1	1	Latched	RI	RI
1	1	0	0	Latched	-	-
1	1	0	1	Latched	Latched RI	-
1	1	1	0	Latched	-	RI
1	1	1	1	Latched	Latched RI	RI

12.3 $\overline{\text{CLKRUN}}$ Protocol

$\overline{\text{CLKRUN}}$ is a hardware method of clock control that can be used in parallel with other types of power management. For the PCIXX20 controller, PCI $\overline{\text{CLKRUN}}$ can be programmed using the multifunction routing register (PCI offset 8Ch) on MFUNC6. CardBus $\overline{\text{CLKRUN}}$ 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 $\overline{\text{CLKRUN}}$ affect the PCIXX20 controller:

- Multifunction routing register – MFUNC6 (PCI offset 8Ch, bits 27-24 set to 0001b). This register requires a 43-kΩ pullup resistor.
- KEEPCLK – System control register (PCI offset 80h, bit 1). Setting this bit to 1 does not allow the PCI $\overline{\text{CLKRUN}}$ protocol to stop or slow the PCI clock.
- STOPCLK – Socket control register (CB offset 10h, bit 7). This bit determines whether the CB $\overline{\text{CLKRUN}}$ protocol is affected by the PCI $\overline{\text{CLKRUN}}$ protocol.
- CLKCTRLEN – Socket power management register (CB offset 20h, bit 16). This bit enables the CB $\overline{\text{CLKRUN}}$ protocol.
- CLKCTRL – Socket power management register (CB offset 20h, bit 0). This bit determines whether the CB $\overline{\text{CLKRUN}}$ protocol either stops or slows CCLK.

12.4 $\overline{\text{SUSPEND}}$

The assertion of the $\overline{\text{SUSPEND}}$ signal gates PCLK, $\overline{\text{GRST}}$, $\overline{\text{PRST}}$ from the PCIXX20 controller. The recommended implementation for $\overline{\text{SUSPEND}}$ is to not use it for power management and simply connect a 43-kΩ pullup resistor. $\overline{\text{SUSPEND}}$ is a nonstandard method of power management and causes many implementation problems. The following guidelines are provided to help reduce implementation issues.

The main purpose of the PCIXX20 $\overline{\text{SUSPEND}}$ pin is to prevent PCI reset from clearing all register context which would require the reconfiguration of the PCIXX20 controller by software. Asserting the PCIXX20 $\overline{\text{SUSPEND}}$ signal also places the controller's PCI outputs in a high-impedance state and gates the PCLK internally to the controller. Due to the high-impedance PCI outputs, it is important that the PCI bus not be parked on the PCIXX20 controller when $\overline{\text{SUSPEND}}$ is asserted.

Another major point to note is that power-down 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 $\overline{\text{SUSPEND}}$, the card remains powered. This creates the possibility of potential card damage. If a 3.3-V card is inserted into the hot slot that was powered to 5 V, then card damage may occur. It is therefore recommended that P2CCLK, bit 27 at PCI offset 80h, is set to 1 so that the internal oscillator is enabled.

13 Migration to PCIXX20 From PCI7X10

There are many differences between the PCIXX20 series and the PCI7X10 controllers. The major differences are: different package and pin-count, dual CardBus support, the addition of simultaneous PCI-based dedicated support for MMC/SD and MS/MSPRO functions, removal of dedicated SmartMedia and SmartCard support (SmartCard now supports via both CardBus sockets), removal of zoomed video support, and the removal of the firmware loader function. Please refer to Section 1 for the PCIXX20 system block diagram.

13.1 Hardware and Pin Assignment Changes

- The PCIXX20 controller is available in the 288 MicroStar BGA™ package (GHK).
- Simultaneous PCI-based dedicated SD/MMC and MS/MSPRO functions and associated signal terminals are added to the PCIXX20 controller. Please refer to Section 7 for dedicated Flash Media implementation.
- SD write-protection is now a high asserted signal with logic 1 indicated that a SD card is write-protected. Please refer to Section 7 for dedicated Flash Media.
- Zoomed video is no longer supported in the PCIXX20 controller.
- Additional V_{CC}, GND, and V_{CCA} terminals have been added, along with the removal of VDx pins. Please refer to Section 3 for detailed information on device power implementation.

13.2 Software Changes

- The PCIXX20 controller no longer uses firmware loader to download firmware onto the controller. The function of loading firmware now resides in the system driver.
- EEPROM loading map of the PCIXX20 controller is significantly different than that of the PCI1620 controller, please refer to Section 11 for detailed EEPROM implementation.
- $\overline{\text{INTD}}$ is now supported by the PCIXX20 controller.
- PCI function 1 is changed to support CardBus socket A, and function 3 is changed to support the dedicated Flash Media sockets.

13.3 Configuration Register Changes

13.3.1 Function 0 and Function 1 - CardBus

13.3.1.1 PCI Configuration Registers

- Device ID for the PCIX620 controller is AC8Dh and device ID for the PCIX420 controller is AC8Eh
- Status register (PCI offset 06h)
  - Bit 3 is designated INT_STATUS bit. Read-update
- CardBus I/O limit register (PCI offset 30h/38h)
  - Bit 1 default value is changed to don't care
- Firmware loader function subsystem vendor ID register (PCI offset 6Ch)
  - This register is removed because the firmware loader is no longer used
- Firmware loader function subsystem ID register (PCI Offset 6Eh)
  - This register is removed because the firmware loader is no longer used
- System control register (PCI offset 6Ch)
  - Default value of this register is changed to 0840 9060h
  - Bit 27 is designated to PSCCLK bit. Read-write, default 0.
  - Bit 20 is changed to RSVD bit. Read-only, returns 0 when read.
  - Bit 3 is changed to RSVD bit. Read-only, returns 0 when read.
- UM_CD debounce register (PCI offset 84h)
  - This register is removed from the PCIXX20 controller.
- General control register (PCI offset 86h)
  - Default value of this register is changed to 0080h.
  - Bit 13 is designated SIM_MODE bit. Read-write, default 0.
  - Bit 10 (12V_SW_SEL bit), the definition is changed. Read-write, default 0.
  - Bits 9-8 are changed to RSVD bits. Read-only, return 0 when read.
  - Bit 7 is designated PCI2_3_EN bit. Read-only, default 1.
  - Bit 5 is changed to DISABLE_FM bit. Read-write, default 0.
  - Bit 4 is changed to DISABLE_SKTB bit. Read-write, default 0.

- Multifunction routing status register (PCI offset 8Ch)
  - Bits 23-20 values:
    - 0110 is changed to RSVD
  - Bits 19-16 values:
    - 0110 is changed to RSVD
    - 1010 is designated to $\overline{\text{INTD}}$
  - Bits 11-8 values:
    - 0110 is changed to RSVD
    - 0111 is changed to RSVD
  - Bits 7-4 values:
    - 0110 is changed to RSVD
    - 0111 is changed to RSVD
  - Bits 3-0 values:
    - 0110 is changed to RSVD
    - 0111 is changed to RSVD
- Card control register (PCI offset 91h)
  - Bits 6-5 are changed to RSVD bits. Read-only, returns 0 when read.
- Diagnostic register (PCI offset 93h)
  - Bit 0 is changed to RSVD bit. Read-only, returns 0 when read.

13.3.1.2 CardBus Socket Registers

- Socket present state register (CardBus socket address + 08h)
  - Bit 27 is changed to RSVD bit. Read-only, returns 0 when read.
- Socket force event register (CardBus socket address + 0Ch)
  - Bit 27 is changed to RSVD bit. Read-only, returns 0 when read.
- Socket control register (CardBus socket address + 10h)
  - Bit 11 is changed to RSVD bit. Read-only, returns 0 when read.
  - Bit 10 is changed to RSVD bit. Read-only, returns 1 when read.
  - Bit 9 is changed to RSVD bit. Read-only, returns 0 when read.

13.3.2 Function 2 – 1394

13.3.2.1 PCI Configuration Registers

- Device ID register (PCI offset 02h)
  - Default value is changed to 802Eh
- Interrupt pin register (PCI offset 3Dh)
  - Default value is changed to 03h

13.3.3 Function 3 – Flash Media

A set of PCI configuration registers in function 3 of the PCIXX20 controller is significantly different from that of the PCI7X10 controller. Please refer to the PCIXX20 data manual which can be located at the IT web portal at http://www.ti.com.

14 Migration to PCIXX20 From PCI1620

There are many differences between the PCIXX20 series and the PCI1620 controllers. The major differences are: different package and pin-count, the addition of 1394a and simultaneous PCI-based dedicated MMC/SD and MS/MSPRO functions, removal of zoomed video support, and Flash Media is no longer supported via the CardBus sockets.

14.1 Hardware and Pin Assignment Changes

- The PCIXX20 controller is available in the 288 MicroStar BGA™ package (GHK).
- The 1394a function and associated power and signals are added to the PCI7X20 controller. Please refer to Section 8 for detailed information on 1394 implementation.
- PCI-based dedicated SD/MMC and MS/MSPRO functions and associated signal terminals are added to the PCIXX20 controller. Please refer to Section 7 for dedicated Flash Media implementation.
- SD write-protection is now a high asserted signal with logic 1 indicated that a SD card is write-protected. Please refer to Section 7 for dedicated Flash Media.
- Dedicated SCL and SDA signals are added to the PCIXX20 controller; hence, the LATCH signal is no longer needed for EEPROM loading.
- Zoomed video is no longer supported in the PCIXX20 controller.
- $\overline{\text{INTC}}$ and $\overline{\text{INTD}}$ are now supported by PCIXX20 controller.
- Additional V_{CC}, GND, V_{CCA}, V_{CCB}, V_{CCP}, and VR_PORT terminals have been added, also, with the addition of 1394 functionality, analog power (AVDx and VDPLL) and ground terminals (AGNx and VSPLL) are added. Please refer to Section 3 for detailed information on device power implementation.
- Factory test pins (TEST0 and PHY_TEST_MA) have also been added for internal testing purposes.

14.2 Software Changes

- The PCIXX20 controller no longer uses firmware loader to download firmware onto the controllers. The function of loading firmware now resides in the system driver.
- EEPROM loading map of the PCIXX20 controller is significantly different than that of the PCI1620 controller, please refer to Section 11 for detailed EEPROM implementation.
- PCI function 2 now supports 1394 for PCI7X20 controller, and function 3 is added to support the dedicated Flash Media sockets.

14.3 Configuration Register Changes

14.3.1 Function 0 and Function 1 - CardBus

14.3.1.1 PCI Configuration Registers

- Device ID for PCIX620 is AC8Dh and device ID for PCIX420 is AC8Eh
- Command register (PCI offset 04h)
  - Bit 10 is designated INT_DISABLE bit. Read-write, default 1.
  - Bit 7 is changed to RSVD bit. Read-only, returns 0 when read.
- Status register (PCI offset 06h)
  - Bit 3 is designated INT_STATUS bit. Read-update
- Revision ID register (PCI offset 08h)
  - Default value is changed to 00h
- CardBus I/O limit register (PCI offset 30h/38h)
  - Bit 1 default value is changed to don't care
- Firmware loader function subsystem vendor ID register (PCI offset 6Ch)
  - This register is removed because firmware loader is no longer used
- Firmware loader function subsystem ID register (PCI offset 6Eh)
  - This register is removed because firmware loader is no longer used
- System control register (PCI offset 6Ch)
  - Default value of this register is changed to 0840 9060h
  - Bit 28 is designated to TIEALL bit. Read-write, default 0.
  - Bit 27 is designated to PSCCLK bit. Read-write, default 0.
  - Bit 20 is changed to RSVD bit. Read-only, returns 0 when read.
- General control register (PCI offset 86h)
  - Default value of this register is changed to 0080h
  - Bit 13 is designated SIM_MODE bit. Read-write, default 0.
  - Bit 12 is designated IO_LIMIT_SEL bit. Read-write, default 0.
  - Bit 11 is designated IO_BASE_SEL bit. Read-write, default 0.
  - Bit 10 is designated 12V_SW_SEL bit. Read-write, default 0.

  - Bits 9-8 are changed to RSVD bits. Read-only, return 0 when read.
  - Bit 7 is designated PCI2_3_EN bit. Read-only, default 1.
  - Bit 5 is changed to DISABLE_FM bit. Read-write, default 0.
  - Bit 4 is designated to DISABLE_SKTB bit. Read-write, default 0.
  - Bit 3 is changed to DISABLE_OHCI bit. Read-write, default 0.
  - Bits 1-0 are designated to ARB_CTRL bits. Read-write, default 11.
- Multifunction routing status register (PCI offset 8Ch)
  - Bits 23-20 values:
    - 0110 is changed to RSVD
    - 0111 is changed to RSVD
    - 1011 is designated to OHCI_LED
  - Bits 19-16 values:
    - 0110 is changed to RSVD
    - 0111 is changed to RSVD
    - 1010 is designated to $\overline{\text{INTD}}$
  - Bits 11-8 values:
    - 0110 is changed to RSVD
    - 0111 is changed to RSVD
    - 1011 is designated to $\overline{\text{INTC}}$
    - 1101 is changed to TEST_MUX
  - Bits 7-4 values:
    - 0100 is changed to OHCI_LED
    - 0110 is changed to RSVD
    - 0111 is changed to RSVD
  - Bits 3-0 values:
    - 0110 is changed to RSVD
    - 0111 is changed to RSVD
- Card control register (PCI offset 91h)
  - Bits 6-5 are changed to RSVD bits. Read-only, returns 0 when read.

- Diagnostic register (PCI offset 93h)
  - Bit 0 is changed to RSVD bit. Read-only, returns 0 when read.

14.3.1.2 CardBus Socket Registers

- Socket present state register (CardBus socket address + 08h)
  - Bit 27 is changed to RSVD bit. Read-only, returns 0 when read.
- Socket force event register (CardBus socket address + 0Ch)
  - Bit 27 is changed to RSVD bit. Read-only, returns 0 when read.
- Socket control register (CardBus socket address + 10h)
  - Bit 11 is changed to RSVD bit. Read-only, returns 0 when read.
  - Bit 10 is changed to RSVD bit. Read-only, returns 1 when read.
  - Bit 9 is changed to RSVD bit. Read-only, returns 0 when read.
  - Bits 2-0 are designated to VPPCTRL bits. Read-write, default 00.

14.3.2 Function 2 – 1394 (PCI7X20 Only)

A set of PCI configuration registers, OHCI registers, TI extension registers, and PHY registers are added to function 2 of the PCI7X20 controller. Please refer to the PCI7X20 data manual which can be located at the TI web portal at http://www.ti.com.

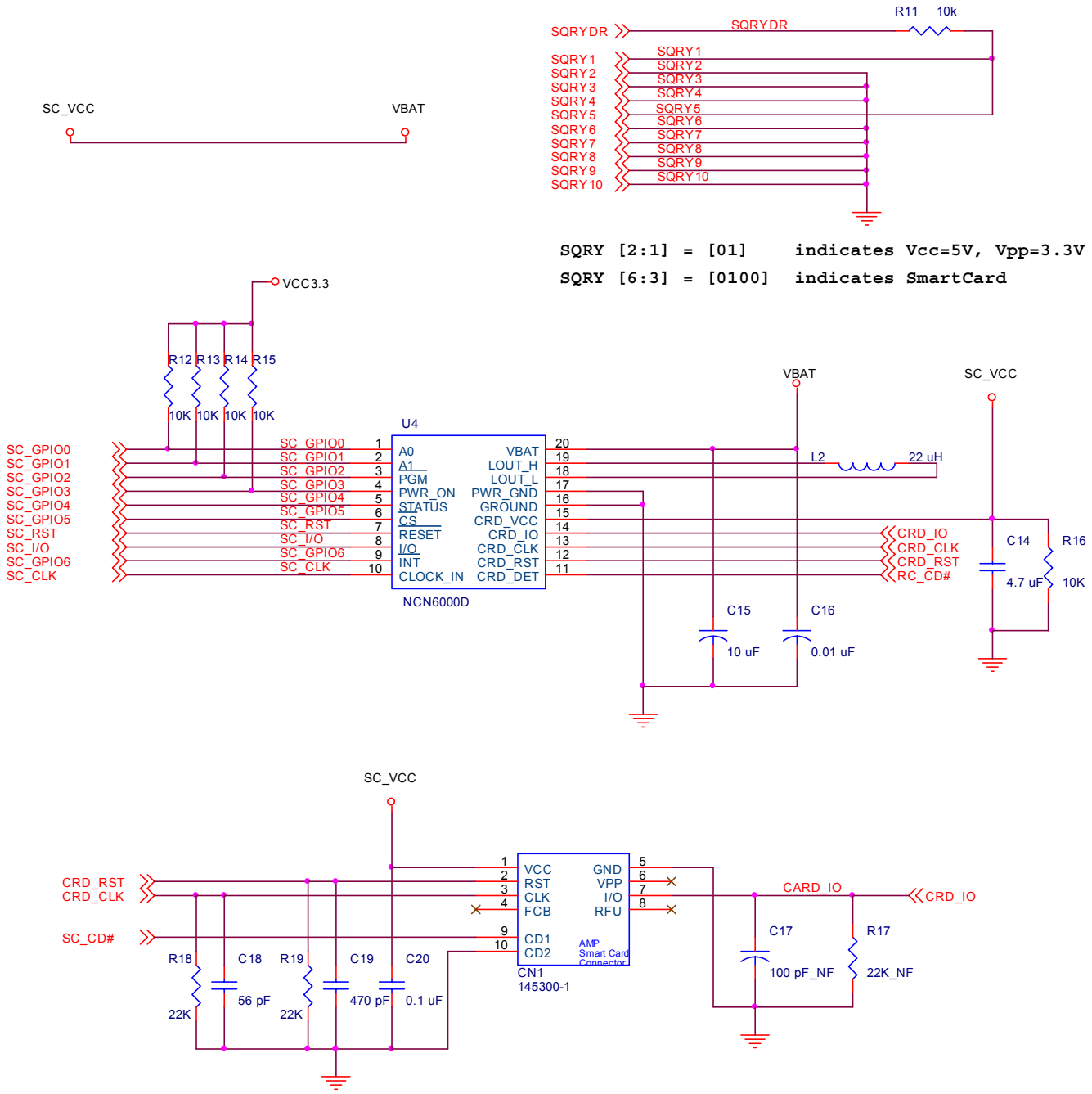
- Interrupt pin register (PCI offset 3Dh)
  - Default value is changed to 03h

14.3.3 Function 3 – Flash Media

A set of PCI configuration registers are added to function 3 of the PCIXX20 controller. Please refer to the PCIXX20 data manual which can be located at the IT web portal at http://www.ti.com.

- Interrupt pin register (PCI offset 3Dh)
  - Default value is changed to 04h

15 Dedicated SmartCard Implementation Schematic



16 Reference Schematics

Please refer to the PCIXX20 EVM schematics for use as reference schematics.

17 References

1. *PCI7620/PCI7420 Dual Socket CardBus and SmartCard Controller With Integrated 1394 and Dedicated SD/MS-PRO Sockets Data Manual from the Texas Instruments Web portal at http://www.ti.com*
2. *PCI6620/PCI6420 Dual Socket CardBus and SmartCard Controller With Dedicated SD/MS-PRO Sockets Data Manual from the Texas Instruments Web portal at http://www.ti.com*
3. *PCI Local Bus Specification Revision 2.3*
4. *PC Card Standard Revision 8.0*
5. *PCI Bus Power Management Interface Specification Revision 1.1*
6. *PCI Mobile Design Guide Revision 1.1*
7. *PCMCIA Proposal (262)*

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