PCI 2.3 Update Document

This document details the changes between the 2.2 and 2.3 version of the PCI specification. The 2.3 specification changes fall into the following categories:

- Support added for System Management Bus (SM Bus)
- Removed support for 5volt only Add-In cards.
- The purpose and function of Stepping was clarified in the 2.3 specification as reflected by the change in name from "Address/Data Stepping" to "IDSEL stepping."
- New Reset timing parameter added.
- Added new "low profile" PCI Add-In card form factor.
- New bit fields added to registers in configuration address space.

This document is intended to be an addendum to MindShare's PCI System Architecture book, 4th edition, based on the 2.2 version of the PCI specification. The document is organized in order of occurrence of changes that are associated with the book and the 2.3 specification. The changes described in this document are incorporated into a 5th edition book available only in electronic form. This 5th edition book is in pdf format and can be purchased at http://www.mindshare.com/books/books_main.html. This electronic edition of the book also incorporates the 4th edition errata and ECNs that include:

Table 1: Summary of 2.2 Specification ECNs

ECN	Description
SMBus	Adds a two wire management interface to the PCI connector.
Reset Timing	Adds a new timing requirement from power valid to reset deassertion.
Low Profile Add-in Card	Adds the Low Profile add-in card form factor.
Add-in Card Trace Impedance	Extends the low end of the add-in card trace impedance.
Add-in Card Keying	Deletes 5 volt only keyed add-in card support.
Class Codes	Updates specification to include new Class Codes.
Capability IDs	Updates specification to include new Capability IDs.

About This Book

The section entitled "Designation of Specification Changes" on page 3, is updated to reflect the new 2.3 specification dated 3/29/02, which includes the errata from document (file pci2_2_errata102499.pdf).

Chapter 1: Intro To PCI

Table 1-1 is extended to include new features supported by PCI. The new entries include:.

Table 1-1: New PCI 2.3 Features

Feature	Description
SM Bus	Support for a System Management bus provides the ability to manage a variety of features.
Add-In Card Size	The specification defines three card sizes: long, short and variable-height short cards, and new low-profile add-in cards.
Add-In Card Compatibility	The PCI spec and PCI-X addendum define cards that operate at different speeds and with different protocols. All 3.3 add-in cards and components including PCI 33 MHz, PCI 66 MHz, PCI-X 66 MHz, and PCI-X 133 MHz are designed to function in a PCI 33 MHz system. Note however, that some cards and components require 3.3 volts bus operation.

The contact information specified at the bottom of Page 13 for obtaining the specification has changed:

PCI Special Interest Group
5440 SW Westgate Drive
Suite 217
Portland, Oregon 97221
Phone: 800-433-5177 (Inside the U.S.)
503-291-2569 (Outside the U.S.)
Fax: 303-297-1090
E-mail: administration@pcisig.com
http://www.pcisig.com

Chapter 3: Reflected-Wave Switching

The section on RST#/REQ64# Timing on page 29 of the 4th edition has been rewritten and expanded. A timing diagram has been added to illustrate the timing relationships between power, clock, and reset. Also, the definition of Tfail has been added to the discussion and timing diagram.

The only change to this section that is brought about by the 2.3 specification is a new parameter defined as "power valid to reset high" timing (Tpvrh). The discussion of this parameter is highlighted in red below and is illustrated in figure 3-5.

RST#/REQ64# Timing

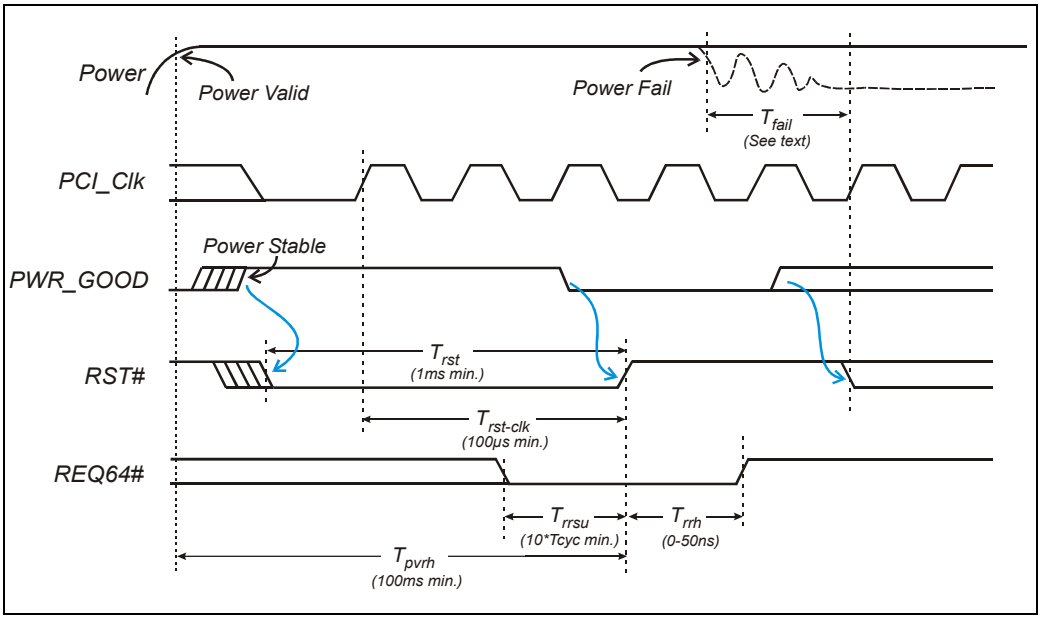
The assertion and deassertion of RST# is asynchronous to the PCI clock signal, however, the specification also permits synchronous reset. The following list describes the timing relationships between power, clock, and reset and illustrated in Figure 3-5 on page 4.

- The specification requires that RST# be asserted as early as possible during the power-up sequence. Figure 3-5 on page 4 illustrates RST# going active following the assertion of PWR_GOOD.
- Once asserted, RST# must remain asserted for a minimum of 1ms after the power has stabilized (T_{rst}).
- RST# must remain asserted for a minimum of 100 microseconds after the CLK has stabilized ($T_{rst-clk}$).
- *During the first reset sequence (i.e. power-up), RST# must remain asserted for 100ms from power valid (Tpvrh). Note that if RST# is asserted while the power supply voltages remain within their specified limits (e.g. a software reset), the minimum pulse width of RST# is Trst.* 2.3
- When RST# is asserted, all devices must float their output drivers within a maximum of 40ns. A device is not considered reset until T_{rst} and $T_{rst-clk}$ have been satisfied.
- During assertion of RST#, the system board reset logic must assert REQ64# for a minimum of 10 clock cycles (T_{rrsu}). REQ64# may remain asserted for a maximum of 50ns after RST# is deasserted (T_{rrh}). For a discussion of REQ64# assertion during reset, refer to “64-bit Cards in 32-bit Add-in Connectors” on page 268.

- If power goes out of specification on either or both power rails, the system will re-assert RST#. This forces all devices to float their outputs to protect against possible parasitic currents short circuiting the output drivers. The time from detecting "power fail" to assertion of RST# (T_{fail}) depends on the power fail condition as follows:
  - 500ns (max.) if either power rail goes out of tolerance by more than 500mV.
  - 100ns (max.) if the 5V rail falls below the 3.3V rail by more than 300mV.

2.3

Figure 3-5: Power, Clock, and Reset Timing



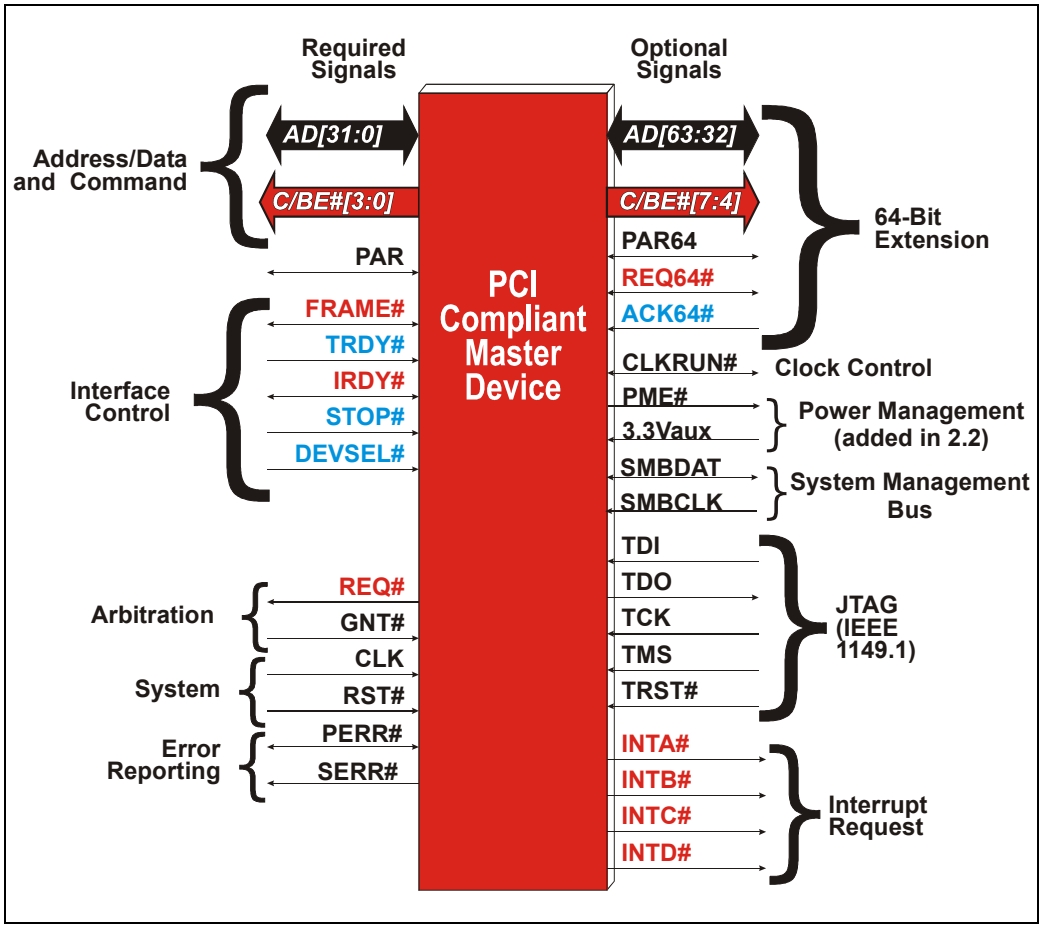
Chapter 4: The Signal Groups

The 2.3 PCI specification adds support for the System Management (SM) Bus. Two new pins were added for this support. The changes made to the 5th edition ebook include:

- Figures 4-1 and 4-2 now include the SM Bus interface signals.
- A new section on the SM Bus interface has also been added
- The section on Snoop Support Signals has been removed.
- Table 4-7 now includes the SM Bus signals.

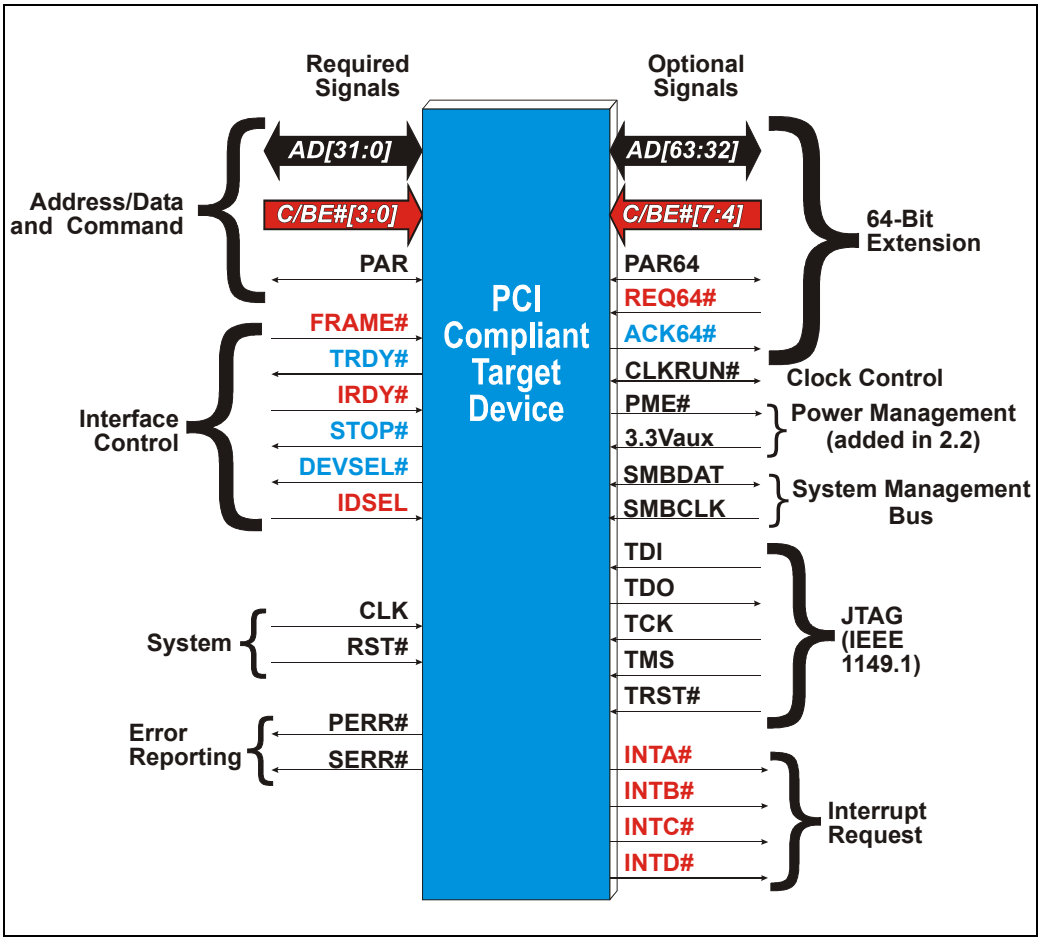
Figure 4-1: PCI-Compliant Master Device Signals

2.3



2.3

Figure 4-2: PCI-Compliant Target Device Signals



System Management Bus Interface

2.3

Table 4-4 provides a brief description of the System Management Bus interface signals.

Table 4-4: System Management Bus Interface Signals

Signal	Description
SMBCLK	*SMBCLK. This signal is an optional SMBus interface clock signal. This signal is driven by an SM Bus master when it wants to send data to or receive data from an SM Bus slave. The SM Bus specification also supports the possibility of multiple masters driving the clock simultaneously and an associated arbitration mechanism that employs SMBDAT. The clock frequency range permitted is 10kHz - 100kHz. SM Bus provides a clock synchronization that allows devices operating at different speeds to co-exist on the bus.*
SMBDAT	*SMBDAT. This signal is driven by a master when initiating a transaction to deliver the slave address and command. Subsequently, SMBDAT carries data to or from a slave device. SMBDAT is also used as part of the arbitration mechanism*

The SM Bus interface is specified in the 2.0 version of the SM Bus specification. This 2.0 specification adds high power characteristics that permits SM devices to operate in the PCI bus environment. Earlier versions of the SM Bus specification supported only lower Power environments (e.g. smart batteries).

Cache Support (Snoop Result) Signals

This section is removed from the 5th edition e-book. Support for the PCI snoop result signals and all related topics were eliminated in the 2.2 spec. The description of Cache support was left is the 4th edition book as a historical reference.

2.3 Signal Types

TABLE 4-7 HAS BEEN UPDATED TO REFLECT THE SMBCLK AND SMBDAT SIGNALS AS SHOWN BELOW.

Table 4-7: PCI Signal Types

Signal(s)	Type	Note
SMBCLK	O/D	PULLED UP ON SYSTEM BOARD.ADDED IN THE 2.3 SPECIFICATION.
SMBDAT		

2.3 Chapter 11: Fast Back-to-Back & IDSEL Stepping

IDSEL Stepping

THE 2.3 SPECIFICATION HAS RENAMED ADDRESS/DATA STEPPING TO IDSEL STEPPING. THE NAME CHANGE DENOTES USE OF STEPPING ONLY FOR TYPE ZERO CONFIGURATION TRANSACTIONS. SPECIFICALLY, 2.3 PERMITS THE USE OF STEPPING ON IDSEL PINS WHEN THEY ARE DRIVEN VIA AD SIGNALS THROUGH RESISTIVELY-COUPLED CONNECTIONS.

IDSEL STEPPING PERMITS THE AD BUS, WHICH CARRIES THE IDSEL INFORMATION DURING TYPE ZERO CONFIGURATION TRANSACTIONS, TO BE DRIVEN PRIOR TO THE ASSERTION OF FRAME#. THIS ENSURES THAT THE IDSEL# INFORMATION, WHICH MUST TRAVERSE A COUPLING RESISTOR, IS VALID WHEN FRAME# IS DRIVEN ACTIVE. THE PROBLEM IS CAUSED BY A SERIES RESISTOR THAN CONNECTS THE AD LINE TO THE IDSEL INPUT OF A TARGET DEVICE, THUS CREATING A SLOW SLEW RATE ON IDSEL. THE NUMBER OF CLOCKS THE ADDRESS BUS SHOULD BE PRE-DRIVEN IS DETERMINED FROM THE RC TIME CONSTANT ON IDSEL. ALL PRIOR USES OF STEPPING THAT WERE PERMITTED BY THE 2.2 AND EARLIER SPECIFICATIONS ARE NO LONGER ALLOWED BY THE 2.3 SPECIFICATION.

Who Must Support Stepping?

All PCI devices must be able to handle receipt of an IDSEL-stepped transaction performed by the master of a configuration type zero transaction. The ability to generate stepping, however, is optional. Devices no longer indicate their ability to perform stepping via the Stepping Control bit (this bit is now reserved in rev 2.3 of the spec).

Why Targets Don't Latch Address During Stepping Process

Because the assertion of FRAME# qualifies the address (and command) as being valid, no targets latch and use the address (including IDSEL) and command until FRAME# is sampled asserted.

Preemption While Stepping in Progress

When the PCI bus arbiter grants the bus to a bus master, the master then waits for bus idle before initiating its transaction. If, during this period of time, the arbiter detects a request from a higher-priority master, it can remove the grant from the first master before it begins a transaction (i.e., before it asserts FRAME#).

Assuming that this doesn't occur, the master retains its grant and awaits bus idle. Upon detection of the bus idle state, the master drives the address onto the AD bus, but delays the assertion of FRAME# for several clocks until a valid IDSEL is present at the target device. During this period of time, the arbiter may still remove the grant from the master. The arbiter hasn't detected FRAME# asserted and may therefore assume that the master hasn't yet started a transaction. If the arbiter receives a request from a higher-priority master, it may remove the grant from the master that is currently engaged in stepping IDSEL onto the AD bus. In response to the loss of grant, the stepping master must immediately tri-state its output drivers.

It is a rule that the arbiter cannot deassert one master's grant and assert grant to another master during the same clock cycle if the bus appears to be idle. The bus may not, in fact, be idle. A master may not have asserted FRAME# yet because it is in the act of stepping IDSEL onto the AD bus.

If the arbiter were to simultaneously remove the stepping master's GNT# and issue GNT# to another master, the following problem would result. On the next rising-edge of the clock, the stepping master detects removal of its GNT# and begins to turn off its address drivers (which takes time). At the same time, the other master detects its GNT# and bus idle (because the stepping master had not yet asserted FRAME#) and initiates a transaction. This results in a collision on the AD bus.

When the bus appears to be idle, the arbiter must remove the grant from one master, wait one clock cycle, and then assert grant to the other master. This provides a one clock cycle buffer zone for the stepping master to back off its output drivers completely before the other master detects its grant along with bus idle and starts its transaction.

It is permissible for the arbiter to simultaneously remove one master's grant and assert another's during the same clock cycle if the bus isn't idle (i.e., a transaction is in progress). There is no danger of a collision because the master that has just received the grant cannot start driving the bus until the current master idles the bus.

Broken Master

The arbiter may assume that a master is broken if the arbiter has issued GNT# to the master, the bus has been idle for 16 clocks, and the master has not asserted FRAME# to start its transaction. The arbiter is permitted to ignore all further requests for bus ownership from the broken master and may optionally report the failure to the OS (in a device-specific fashion).

Stepping Example

Figure 11-3 on page 11 provides an example of an initiator using stepping over a period of three clocks to drive the address onto the AD bus.

Clock 3. The initiator can start the transaction on clock three (GNT# sampled asserted and bus idle—FRAME# and IRDY# sampled deasserted). It then begins to drive the address and IDSEL onto the AD bus and the command onto the C/BE bus.

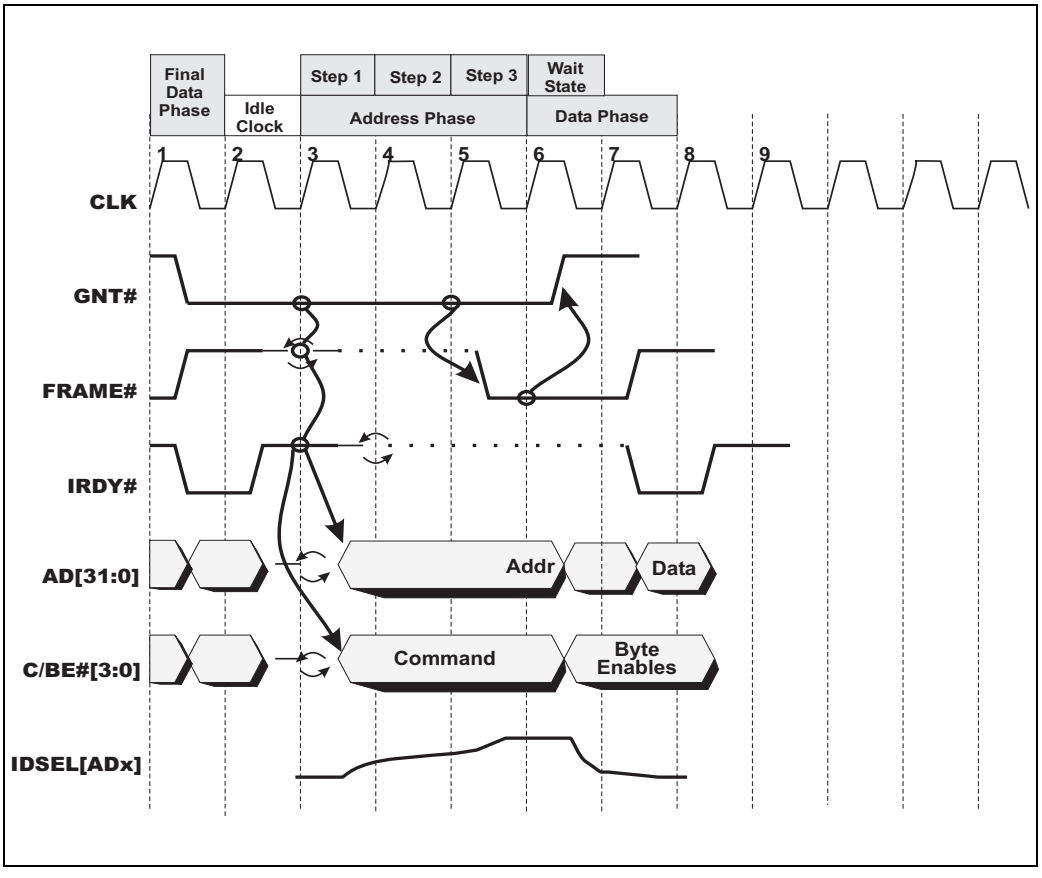
Clock 4. During clock cycle four, it continues to drive the address and IDSEL onto the AD bus; however IDSEL has not yet reached VIH at the target.

Clock 5. During the clock cycle five, IDSEL has reached VIH and the master

ASSERTS FRAME#, INDICATING THE PRESENCE OF THE ADDRESS, IDSEL, AND COMMAND.

CLOCK 6. WHEN THE TARGETS SAMPLE FRAME# ASSERTED ON THE RISING-EDGE OF CLOCK SIX (THE END OF THE ADDRESS PHASE), THEY LATCH THE ADDRESS, IDSEL, AND COMMAND AND BEGIN THE ADDRESS DECODE.

Figure 11-3: Example of IDSEL Stepping

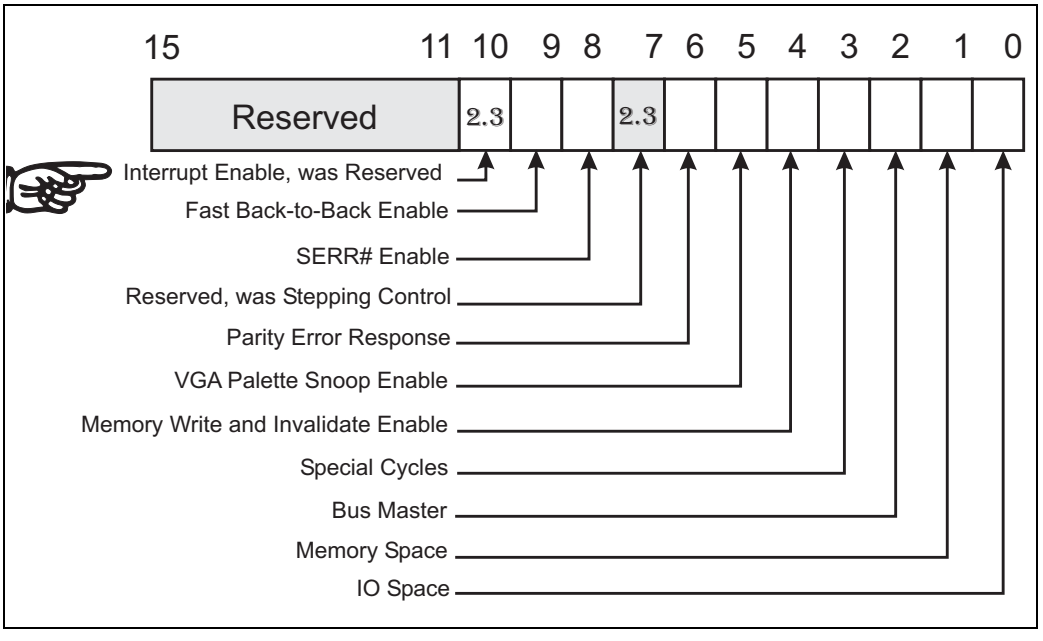


Chapter 14: Interrupts

2.3 Interrupt Disable

THE 2.3 SPECIFICATION ADDED AN INTERRUPT DISABLE BIT (BIT 10) TO THE CONFIGURATION COMMAND REGISTER. SEE FIGURE 14-4 ON PAGE 12. THE BIT IS CLEARED AT RESET PERMITTING GENERATION OF INTX SIGNAL GENERATION. SOFTWARE MAY SET THIS BIT THEREBY INHIBITING GENERATION OF INTERRUPTS VIA THE INTX SIGNALS. NOTE THAT THE INTERRUPT DISABLE BIT HAS NO EFFECT ON MESSAGE SIGNALLED INTERRUPTS (MSI). THESE INTERRUPTS ARE ENABLED VIA THE MSI CAPABILITY COMMAND REGISTER.

Figure 14-4: Interrupt Disable Bit Added to Configuration Command Register



2.3 Interrupt Status

THE 2.3 SPECIFICATION ADDED AN INTERRUPT STATUS BIT TO THE CONFIGURATION STATUS REGISTER (PICTURED IN FIGURE 14-5 ON PAGE 14). A FUNCTION MUST SET THIS STATUS BIT WHEN AN INTERRUPT IS PENDING. IN ADDITION, IF THE INTERRUPT DISABLE BIT IN THE CONFIGURATION COMMAND REGISTER IS CLEARED (I.E. INTERRUPTS ENABLED), THEN THE FUNCTION'S INTX# SIGNAL IS ASSERTED, BUT ONLY AFTER THE STATUS BIT IS SET. THIS BIT IS UNAFFECTED BY THE STATE OF THE INTERRUPT DISABLE BIT, AND THIS BIT HAS NO AFFECT ON THE MSI MECHANISM.

The specification does not comment on the intended use of this bit. The temptation is great to assume that this bit is intended to replace the function-specific interrupt pending bit that is mapped into memory or IO space. This bit is accessed by the function's ISR to determine if it has an interrupt is pending. If the interrupt pending bit is set the device needs servicing. The ISR clears the interrupt pending bit and services the interrupt. Clearing the bit causes the INTx signal to be deasserted, which of course removes the interrupt request. Because the Interrupt Status bit is read only, it cannot replace the interrupt pending bit.

Now let's look at a possible use for the Interrupt Status bit. It could be useful to the Operating System during the process of handling interrupt sharing between PCI devices. Because multiple PCI functions can share the same interrupt, software must decide which device(s) have actually signalled an interrupt request. The Operating System (OS) manages this process by maintaining a list of interrupt service routines that share the same interrupt. Prior to the implementation of the Interrupt Status bit, handling interrupt sharing involved the following steps:

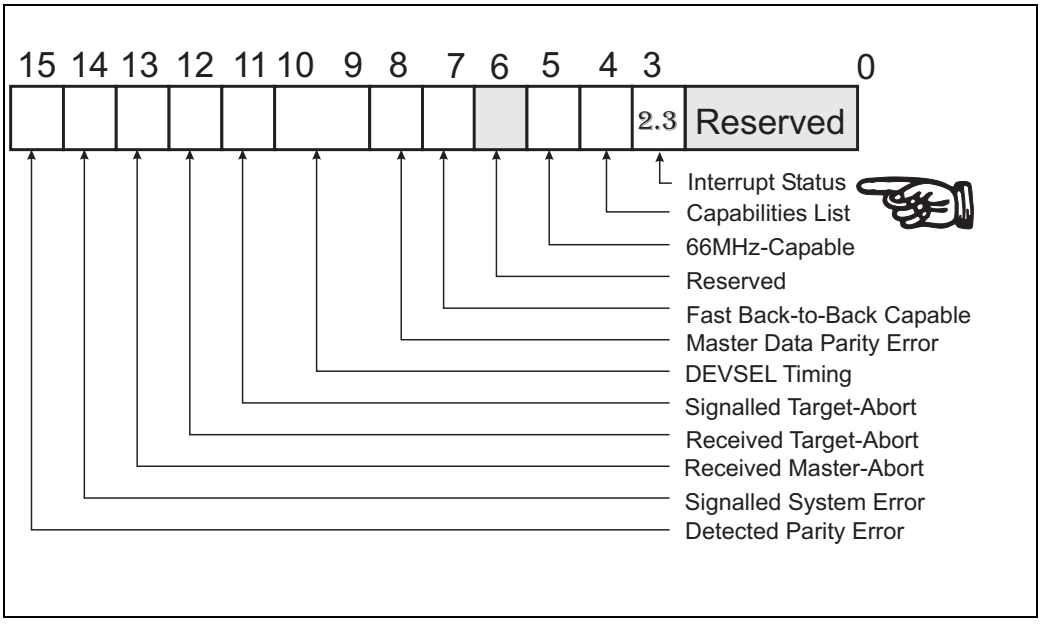
1. The OS calls the first interrupt service routine (ISR).
2. The first ISR checks its function-specific interrupt pending bit.
3. If the bit is set, the ISR performs the required service and then returns to the OS.
4. If the bit is cleared, the ISR recognized that it does not have an interrupt pending and returned to the OS.

This process is repeated until all the ISRs are called. This procedure is necessary because the OS has no other way of determining which devices have an interrupt pending.

The Interrupt Status bit introduced by the 2.3 version of the PCI specification provides the OS with a known location to check which devices currently have

interrupts pending execution. Rather than calling each ISR to check it's interrupt pending bit, the OS can simply check the Interrupt Status bit of each function that shares the interrupt to determine which need servicing. This reduces the overhead required to handle interrupt sharing, and reduces the associated latency.

Figure 14-5: Interrupt Status Bit Added to PCI Configuration Status Register



Chapter 19: Configuration Registers

Changes to the Configuration Registers chapter include:

- New Sub Class Codes and definitions
- New Command and Status Register bits
- Removed discussion of UDF

Class Code Register Changes

The following table entries have been added by the PCI 2.3 specification.

Table 19-3: Class Code 1: Mass Storage Controllers

Sub-Class	Prog. I/F	Description
05h	20h	*ATA CONTROLLER WITH SINGLE DMA*
	30h	*ATA CONTROLLER WITH CHAINED DMA*

2.3

Table 19-4: Class Code 2: Network Controllers

Sub-Class	Prog. I/F	Description
05h	00h	*WORLDFIP CONTROLLER*
06h	xxh	*PICMG 2.14 MULTI COMPUTING*

2.3

2.3

Table 19-8: Class Code 6: Bridge Devices

Sub-Class	Prog. I/F	Description
09h	40h	*SEMI-TRANSPARENT PCI-TO-PCI BRIDGE WITH THE PRIMARY PCI BUS SIDE FACING THE SYSTEM HOST PROCESSOR.*
	80h	*SEMI-TRANSPARENT PCI-TO-PCI BRIDGE WITH THE SECONDARY PCI BUS SIDE FACING THE SYSTEM HOST PROCESSOR*
0Ah	00h	*INFINIBAND-TO-PCI HOST BRIDGE*

2.3

2.3

Table 19-9: Class Code 7: Simple Communications Controllers

	Sub-Class	Prog. I/F	Description
2.3	04h	00h	GPIB (IEEE 488.1/2) CONTROLLER
2.3	05h	00h	SMART CARD

Table 19-14: Class Code C: Serial Bus Controllers

	Sub-Class	Prog. I/F	Description
2.3	06h	00h	INFINIBAND
	07h	00h	IPMI SMIC INTERFACE
2.3		01h	IPMI KYBD CONTROLLER STYLE INTERFACE
		02h	IPMI BLOCK TRANSFER INTERFACE
2.3	08h	00h	SERCOS INTERFACE STANDARD (IEC 61491)
2.3	09h	00h	CANBUS

Table 19-15: Class Code D: Wireless Controllers

	Sub-Class	Interface	Meaning
2.3	11h	00h	BLUETOOTH
2.3	12h	00h	BROADBAND

Table 19-19: Class Code 11h: Data Acquisition and Signal Processing Controllers

	Sub-Class	Interface	Meaning
2.3	01h	00h	*PERFORMANCE COUNTERS*
2.3	10h	00h	*COMMUNICATIONS SYNCHRONIZATION PLUS TIME AND FREQUENCY TEST/MEASUREMENT*
2.3	20h	00h	*MANAGEMENT CARD*

COMMAND REGISTER

Always mandatory. This register provides basic control over the device's ability to respond to and / or perform PCI accesses. The 2.3 specification has made the following changes to the command register:

- Bit 7, previously defined as Stepping Control is now Reserved.
- Bit 10, previously Reserved is now defined as Interrupt Disable.

Table 19-21 on page 18 pictures the bits that have changed with the introduction of the PCI 2.3 specification and the register is illustrated in Figure 19-3 on page 17.

Figure 19-3: Command Register Bit Assignment

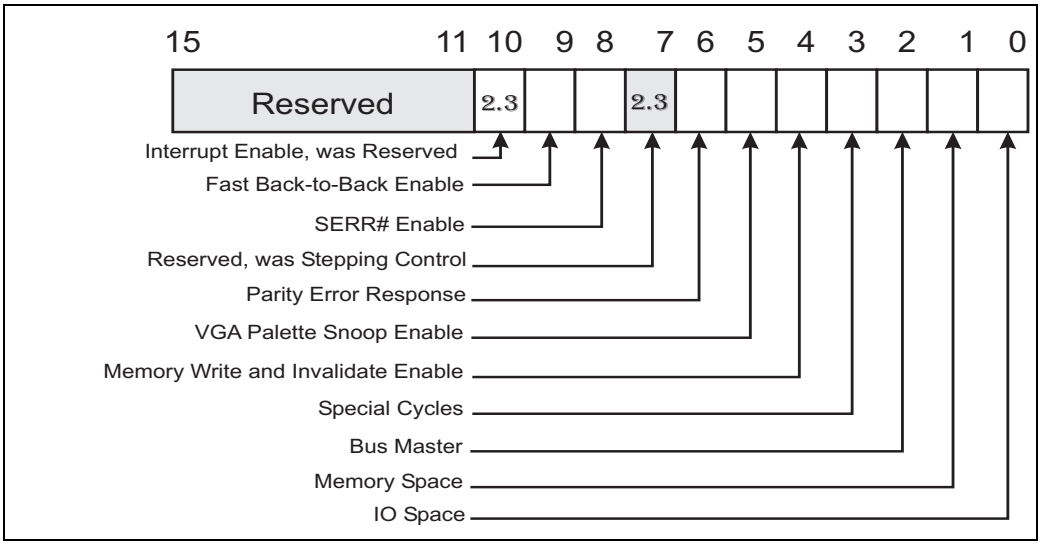


Table 19-21: Command Register Bit Assignment

Bit	Function
7	**RESERVED.** *THIS BIT WAS STEPPING CONTROL IN PCI VERSION 2.2.*
10	**INTERRUPT DISABLE**. *THE BIT DISABLES THE FUNCTION FROM ASSERTING INTx#. A VALUE OF 0 ENABLES THE ASSERTION OF ITS INTx# SIGNAL AND A VALUE OF 1 DISABLES INTx SIGNALLING. REFER TO "INTERRUPT DISABLE" ON PAGE 236 FOR DETAILS REGARDING THIS BIT. NOTE THAT THIS BIT HAS NO AFFECT ON MSI.* **REQUIRED?** *A FUNCTION THAT IS CAPABLE OF SIGNALLING INTERRUPTS VIA THE INTx SIGNAL MUST IMPLEMENT THIS BIT.* **DEFAULT SETTING:** *THE DEFAULT STATE AFTER RESET IS ZERO (INTERRUPTS ENABLED). A VALUE OF 1 DISABLES THE ASSERTION OF ITS INTx# SIGNAL.*
15:11	Reserved

2.3
2.3

Status Register

Figure 19-4 pictures the Status Register and Table 19-22 on page 19 describes the Status register bit that has been added by the PCI 2.3 spec.

Figure 19-4: Status Register Bit Assignment

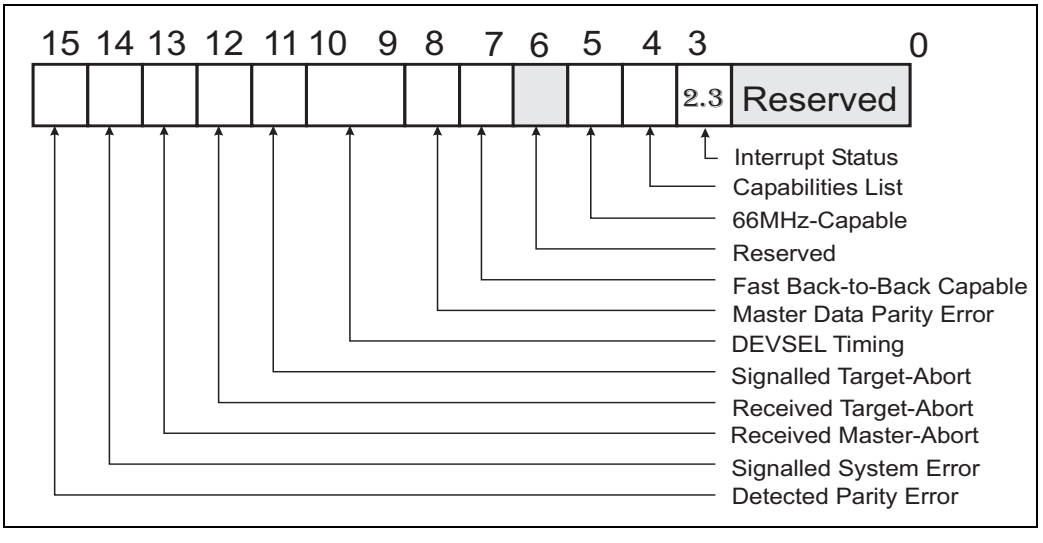


Table 19-22: Status Register Bit Assignment

Bit	R/W	Function
3:0	R	**Reserved**. Hardwired to zero.
2.3 *3*	*R*	*INTERRUPT STATUS. THIS BIT REFLECTS THE STATE OF THE INTERRUPT WITHIN THE FUNCTION. THE INTX SIGNAL IS ASSERTED WHEN THE FOLLOWING CONDITIONS ARE MET:* *1. THE INTERRUPT DISABLE BIT IN THE COMMAND REGISTER IS A 0 (INTERRUPTS ARE ENABLED).* *2. THIS INTERRUPT STATUS BIT IS A 1 (AN INTERRUPT IS PENDING)* *NOTE THAT SETTING THE INTERRUPT DISABLE BIT TO A 1 HAS NO EFFECT ON THE STATE OF THIS BIT.* ***REQUIRED? THIS BIT IS REQUIRED BY FUNCTIONS THAT ARE CAPABLE OF SIGNALLING INTERRUPTS VIA AN INTX SIGNAL.***

Chapter 21: Add-In Cards and Connectors

Add-In Connectors

32- AND 64-BIT CONNECTORS

THE 2.3 SPECIFICATION MADE TWO MAJOR CHANGES TO THE CONNECTOR AND CARD DEFINITIONS: 2.3

- *ELIMINATION OF THE 5 VOLT ONLY CARD — TO PUSH THE MIGRATION FROM 5V-ONLY IMPLEMENTATIONS TO OR 3.3V OR UNIVERSAL CARD IMPLEMENTATIONS. NOTE THAT THIS DOES NOT AFFECT 5V CONNECTORS.*
- *ADDITION OF TWO NEW PINS TO SUPPORT THE SYSTEM MANAGEMENT BUS IN PCI.*

TABLE 21-1 ON PAGE 20 ILLUSTRATES THE 2.3 PINOUT CHANGES. NOTE THAT THE SHADED COLUMN FOR 5V ONLY CARDS IS NO LONGER DEFINED WITHIN THE 2.3 SPECIFICATION, BUT IS LEFT HERE FOR BACKWARD REFERENCE.

2.3 *THIS TABLE SHOWS THE PINOUT FOR THREE CARD TYPES:*

- *5V ONLY CARD — NO LONGER SUPPORTED BY THE 2.3 SPEC.*
- 3V only card
- Universal card (the only 2.3 compliant card that will fit into a 5V connector)

2.3 *THE FOLLOWING PINOUT CHANGES WERE MADE IN THE 2.3 SPEC:*

- *PIN B38 WAS GROUND AND IS NOW DEFINED AS PCIXCAP (PCI-X CAPABILITY). THIS PIN IS DEFINED BY THE PCI-X ADDENDUM TO THE PCI 2.3 SPEC. IT IS ADDED HERE TO INDICATE ITS USE IN PCI-X AND TO INDICATE THAT THE PCI AND PCI-X CONNECTORS ARE THE SAME EXCEPT FOR THIS PIN.*
- *PIN A40 WAS RESERVED AND IS NOW DEFINED AS THE SMBCLK PIN.*
- *PIN A41 WAS RESERVED AND IS NOW DEFINED AS THE SMBDAT PIN.*

2.3 *Table 21-1: PCI Add-In Card Pinouts*

Pin	5V Card		Universal Card		3.3V Card		Comment
	Side B	Side A	Side B	Side A	Side B	Side A	
38	Ground	STOP#	*PCIXCAP*	STOP#	*PCIXCAP*	STOP#	
40	PERR#	***SMBCLK***	PERR#	***SMBCLK***	PERR#	***SMBCLK***	
41	+3.3V	***SMBDAT***	+3.3V	***SMBDAT***	+3.3V	***SMBDAT***	

Figure 21-3 on page 21 illustrates the change in support for 5V only cards that only the Universal card provides 5V connector compatibility.

Figure 21-3: 3.3V, and Universal Card (5V Only Cards Not Support by Ver. 2.3)