



# **Intel® 450NX PCIset Specification Update**

Release Date: February 1999

Order Number: 243848-008

The Intel® 450NX PCIset may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are documented in this Specification Update.

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The Intel® 450NX PCIset may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

The Specification Update should be publicly available following the last shipment date for a period of time equal to the specific product's warranty period. Hardcopy Specification Updates will be available for one (1) year following End of Life (EOL). Web access will be available for three (3) years following EOL.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725 or by visiting Intel's website at <http://www.intel.com>

Copyright © Intel Corporation 1999.

\* Third-party brands and names are the property of their respective owners.

CONTENTS

REVISION HISTORY ..... v

PREFACE ..... vi

**Specification Update for the Intel® 450NX PCIset ..... 1**

GENERAL INFORMATION..... 3

ERRATA..... 6

DOCUMENTATION CHANGES ..... 16



## REVISION HISTORY

Date of Revision	Version	Description
June 1998	-001	This is the first revision of the Intel® 450NX PCIsset Specification Update.
July 1998	-002	Added Errata 20 through 22.
August 1998	-003	Removed Erratum 10 (referred to preliminary specification), renumbered Errata 11 through 21, added Erratum 22.
September 1998	-004	Updated Basic Information table, added Document Change section, added Document Changes 1 and 2.
October 1998	-005	Updated Erratum 22.
November 1998	-006	Added Erratum 23. Added S-Spec information.
January 1999	-007	Added Erratum 24. Updated Summary Table of Changes.
February 1999	-008	Added Document Change 3. Updated S-Spec information. Updated status in Summary Table.

## PREFACE

This document is an update to the specifications contained in the *Intel® 450NX PCIset* datasheet (Order Number 243771-001). It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools. It contains S-Specs, Specification Changes, Errata, Specification Clarifications, and Documentation Changes.

## Nomenclature

**Specification Changes** are modifications to the current published specifications for the Intel® 450NX PCIset. These changes will be incorporated in the next release of the specifications.

**S-Specs** are exceptions to the published specifications, and apply only to the units assembled under that S-spec.

**Specification Clarifications** describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in the next release of the specifications.

**Documentation Changes** include typos, errors, or omissions from the current published specifications. These changes will be incorporated in the next release of the specifications.

**Errata** are design defects or errors. Errata may cause the Intel 450NX PCIset's behavior to deviate from published specifications. Hardware and software designed to be used with any given PCIset stepping must assume that all errata documented for that processor stepping are present on all devices.

## Identification Information

Intel® 450NX PCIset Identification Information

Type	Stepping	S-Spec
82451 (MIOC)	B1	SL2RV
82452 (RCG)	B0	SL2RW
82453 (MUX)	B0	SL2RX
82454 (PXB)	B0	SL2RU
82454 (PXB)	B1	SL2ZA
82454 (PXB)	C0	SL36R

# **Specification Update for the Intel® 450NX PCIset**

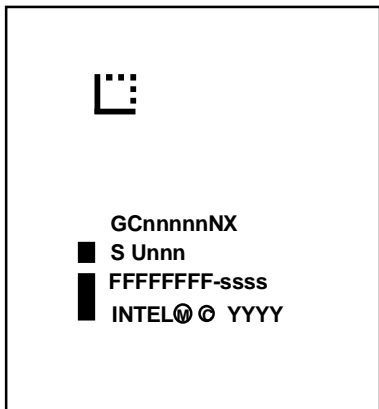




## GENERAL INFORMATION

### Top Markings

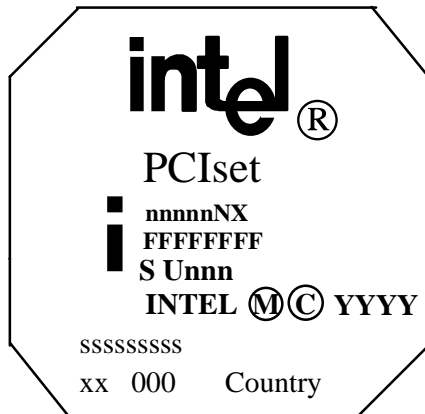
PXB and MIOC B-Step Prodn. Units, PBGA:



#### NOTES:

- nnnnnNX = Product Number
- S Unnn = S-spec Number
- FFFFFFFF-ssss = FPO and Serial Numbers
- YYYY = Copyright Dates

RCG and MUX B-Step Prodn. Units, BGA:



#### NOTES:

- nnnnnNX = Product Number
- FFFFFFFF = FPO Number
- S Unnn = S-spec Number
- YYYY = Copyright Dates
- ssssssss= Serial Number
- Country= Country of Origin

## Summary Table of Changes

The following table indicates the Errata which apply to Intel 450NX PCISSET. Intel intends to fix some of the errata in a future stepping of the components, and to account for the other outstanding issues through documentation or specification changes as noted. This table uses the following notations:

### Codes Used in Summary Table

X: Specification Change, Erratum, Specification Clarification, or Documentation Change applies to the given PCISSET stepping.

Doc: Intel intends to update the appropriate documentation in a future revision.

Fix: This erratum is intended to be fixed in a future stepping of the component.

Fixed: This erratum has been previously fixed.

NoFix: There are no plans to fix this erratum.

(No mark) or (blank box): This item is fixed in or does not apply to the given stepping.

Shaded: This item is either new or modified from the previous version of the document.

NO.	MIOC B1	PXB B0	PXB B1	MUX B0	Plans	ERRATA
1	X				NoFix	Performance Counter Master Enable (PCME) bit in MIOC and PXB CONFIG register does not function as specified
2	X				NoFix	Accesses to SMM space might be erroneously handled if retried
3	X				NoFix	Packets still recognized by MIOC when Live Port flag for F16 is zero
4	X	X	X		NoFix	Inverted sampling of output data cells in boundary scan
5		X	X		NoFix	"Check connection" packets to PXB may not be recognized
6		X	X		NoFix	Deadlock due to simultaneous inbound and outbound PXB locks with PHOLD# asserted
7		X	X		NoFix	64-bit PXB PCI Bus nonconformance with 32-bit target
8	X				NoFix	MIOC does ECC checking on data that is not targeted to itself
9	X				NoFix	Hard reset implementation using CF9h register causes lockup condition
10		X	X		NoFix	PXB will not disable I/O address masking for I/O addresses 64K + 1, 2, 3
11	X				NoFix	MIOC prematurely asserts IO_REQ# to system bus external arbiter
12	X				NoFix	Zero-length requests to third party agents may cause buffer conflict
13	X				NoFix	Response optimization for zero-length read requests causes MIOC hang
14		X	X		Fixed	PXB PCI PLLs show sensitivity to Analog V <sub>CC</sub> levels

NO.	MIOC B1	PXB B0	PXB B1	MUX B0	Plans	ERRATA
15	X				NoFix	DBSY# causes hang condition on inbound transaction with HITM#
16				X	NoFix	MUX originates glitches on DSTRB# signals during memory data bus simultaneous-switching
17	X				NoFix	MIOC noise sensitivity on DSTRB# inputs
18					Fix	Erroneous checking of expander bus parity by the MIOC
19		X			Fix	PIIX4E controller and PXB may livelock when using certain DMA transfer modes
20		X	X		NoFix	PXB arbiter can cause deadlock condition in bus lock mode
21	X				NoFix	ERR#[1:0] pins do not reflect ERRSTS, HEL, MEL state during BINIT# initiated resets
22		X	X		NoFix	Device drivers that violate the producer/consumer model may provide invalid data
23	X				NoFix	Nonascending cacheline evictions to write combining PCI targets may result in data corruption
24		X	X		NoFix	Fifth byte of PXB Performance Monitoring Data registers aliases third byte

NO.	MIOC B1	PXB B0	PXB B1	MUX B0	Plans	DOCUMENT CHANGES
1	X				Doc	Description of MIOC CVCR/CVDR Registers Bit 10
2		X	X		Doc	Description of PXB Configuration register bit 1
3	X	X	X	X	Doc	Description of MIOC/PXB boundary scan functionality

## ERRATA

### 1. *Performance Counter Master Enable (PCME) Bit in MIOC and PXB CONFIG Register Does Not Function as Specified*

**PROBLEM:** If the PCME bit is used to stop the MIOC's performance counters, and a value is written to restart the counting, the counters will be overwritten by the internal (stopped) count value in the very next clock.

**IMPLICATION:** The performance counter access procedure does not function as expected, and incorrect performance counter values may be used if the workaround is not implemented.

**WORKAROUND:** This erratum can be worked around using the following steps:

1. Stop all the counters by clearing the PCME.
2. Write the Count Mode field of the MIOC PMR (3:2) register to 0 (stop counting).
3. Access the PMD counters.
4. Write the Count mode field in the PMR to resume counting in the manner desired.

Start the counters by writing a 1 to the PCME.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

### 2. *Accesses to SMM Space Might Be Erroneously Handled If Retried*

**PROBLEM:** If a third party agent issues a read access to SMM space with the SMMEM# signal deasserted and the access is retried with DEN# deasserted, the MIOC will accept the transaction and send it to the PXB. If the next read to an address in the same cache line has the SMMEM# signal asserted, the data returned will be the one from I/O space instead of from memory. The MIOC will return the data it obtained from the previous access to SMM space (which was directed to the compatibility bus because the SMMEM# signal was deasserted).

**IMPLICATION:** Third party agent SMM accesses may not return the expected data if the third party device does not serialize all SMM accesses like the P6 family processors.

**WORKAROUND:** None.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

### 3. *Packets Still Recognized By MIOC When Live Port Flag for F16 is Zero*

**PROBLEM:** The Live Port flags do not act as specified. The specification states that when a Live Port flag is cleared, all packets arriving from the corresponding port should be ignored by the MIOC, and no requests, responses, or data should be forwarded to the internal logic. This erratum causes the status of the flags to be ignored, and packets from the PXB whose Live Port flag are cleared are accepted and processed internally.

**IMPLICATION:** If a system is implemented with a single PXB and the second F16 port is not properly terminated, the MIOC may hang.

**WORKAROUND:** If a second PXB is not used, the XRTS# pin of the unused F16 bus must be pulled up. After power up either PXB can be disabled by writing to the Reset Control Register bit 4 for Port 0 and bit 5 for Port 1.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## 4. *Inverted Sampling of Output Data Cells in Boundary Scan*

**PROBLEM:** The IEEE 1149.1 standard for boundary scan architecture indicates that the value captured in the data register of an output or bi-directional pin during a “sample/preload” instruction should be a 1 if the pin would be driven high if the output enable were active. A number of pins violate this specification, because the sampling of output data cells is inverted.

**IMPLICATION:** It is not expected that any board level tests depend on the value returned by Sample/Preload. It is expected that Sample/Preload is only used to initialize the boundary scan chain and expected values will be sampled using EXTEST.

The ITP pins command will not work as specified.

**WORKAROUND:** The BSDL cell definition of the affected pins has been modified. Affected pins expect an unknown on Sample/Preload. Older versions of some boundary Scan test generation software does not correctly interpret this description.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## 5. *“Check Connection” Packets to PXB May Not Be Recognized*

**PROBLEM:** A “check connection” packet to the PXB will only be acknowledged if no other outbound transaction immediately precedes it.

**IMPLICATION:** To enable the second F16 port, the first transaction must be a “check connection.” Checking for either PXB’s connection post-boot (e.g., for diagnostic purposes) may not return data as expected.

**WORKAROUND:** None identified.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## 6. *Deadlock Due to Simultaneous Inbound and Outbound PXB Locks with PHOLD# Asserted*

**PROBLEM:** An architectural condition exists in the PXB whereby an inbound locked transaction from a PCI master, followed by a PHOLD# request from the south bridge and an outbound LOCK from the processor, will result in a deadlock.

**IMPLICATION:** Any PCI bus master that can generate an inbound lock to the PXB, if placed on the compatibility bus, may cause a deadlock.

**WORKAROUND:** Do not place a PCI card that can generate inbound locks to the PXB on the compatibility bus. Such devices should be placed on the other three PCI buses in a fully configured Intel 450NX PCIsset-based system.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## 7. *64-bit PXB PCI Bus Nonconformance with 32-bit Target*

**PROBLEM:** When a PXB configured to use a 64-bit PCI bus initiates a transaction to a PCI target that does not respond with ACK64#, the PCI specification, revision 2.1, states that the master may stop driving the upper AD lines or could continue to present the full 64-bits of data on each even DWORD address boundary. On the odd

DWORD boundary, the master is expected to drive the same data on the upper and lower halves of the AD bus. The PXB presently does neither, but instead drives random data on the upper half of the AD bus.

**IMPLICATION:** If a device requires correct data on the upper 32 bits of the AD bus in this situation, it will receive incorrect data. Intel is not aware of any PCI devices that are affected by this erratum. However, individual systems should be checked to determine if this erratum presents a problem

**WORKAROUND:** None identified.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## **8. *MIOC Does ECC Checking on Data That is Not Targeted to Itself***

**PROBLEM:** When single or multi-bit error logging is enabled on the system during transactions targeted to an agent other than the MIOC via the ECCCMD register. The MIOC checks the data and the ECC bits on the system bus for every DRDY# assertion, except when DRDY# is caused by the MIOC driving data to the system.

The MIOC will detect data corruption errors between agents on the system bus.

**IMPLICATION:** If there is data corruption on the system bus during transactions targeted to an agent other than the MIOC, an error will be logged in the HEL[1:0] registers, and the INTREQ# pin or the processor BERR# pins will be asserted (and possibly BINIT# as well).

The same events may occur if the MIOC reads the ECC bits on a PCI to TPA time-out condition and must drive the time-out data.

Data corruption on a transaction which is not directed at the MIOC may cause the MIOC to behave as if the error had occurred on a transaction which it claimed.

**WORKAROUND:** None identified. However, an unrecoverable ECC error is a catastrophic event that should cause re-initialization of the system. No workaround may be necessary for a given system.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## **9. *Hard Reset Implementation Using CF9h Register Causes Lockup Condition***

**PROBLEM:** The Intel 450NX PCIsset does not intercept writes to the register at address CF9h to implement reset; the PCIsset supports these via a new RC (Reset Control) register. The specification recommends using the PIIX4e, which does not support PCIsset hard reset via the register at address CF9h.

**IMPLICATION:** Utilities that rely on the register at CF9h to perform a PCIsset hard reset will only stop the PXB from responding to PCI requests without the knowledge of the MIOC or the CPU which may result in a system hang condition.

**WORKAROUND:** Besides using the PIIX4e output CPURST as recommended in the Intel 450NX PCIsset specification, external logic should implemented such that the initial assertion of the signal (CPURST = 1, PCIRST= 0) results in a pulse to the Intel 450NX PCIsset PWRGOOD signal. A single pulse is required to avoid functional feedback loops between PWRGOOD, PCIRST, and CPURST.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## 10. *PXB Will Not Disable I/O Address Masking For I/O Addresses 64K + 1, 2, 3*

**PROBLEM:** The Intel 450NX PCISet specification describes a PXB feature in its configuration register (address 40-41h) bit 9 which controls an "I/O Address Mask Enable" for processor accesses to I/O addresses above 64 Kbytes. However, the MIOC always zeros address bits [31:16] when sending I/O transactions to the PXB, rendering the "mask disabled" mode nonfunctional.

**IMPLICATION:** This feature cannot be used to expand I/O address space.

**WORKAROUND:** System designers should not attempt to use any I/O space beyond the 64 Kbyte boundary. Do not use a processor's legacy ability to access I/O addresses between 64 Kbytes and 64 Kbytes + 3.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## 11. *MIOC Prematurely Asserts IO\_REQ# to System Bus External Arbiter*

**PROBLEM:** The Intel 450NX PCISet specification states that a device must not assert IO\_REQ until its IO\_GNT# from a previous request has been deasserted. However, if an external arbiter holds IO\_GNT# more than two clock cycles after IO\_REQ# is observed deasserted, then the MIOC will assert IO\_REQ# again.

**IMPLICATION:** The MIOC may start the next transaction assuming that it does not need to rearbiter for the system bus. This assumption conflicts with the third-party agent system bus ownership, causing a contention condition.

**WORKAROUND:** Third-party agent designs where the IO\_GNT# signal is held asserted after the IO\_REQ# signal from the MIOC has been deasserted need to be aware of this condition. However, it is unlikely that an external arbiter would hold IO\_GNT# long enough to cause a concern.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## 12. *Zero-Length Requests to Third Party Agents May Cause Buffer Conflict*

**PROBLEM:** When the MIOC issues a read transaction with Length = 0 and BE = 0 on the system bus to a third party agent (TPA) and the transaction reaches top of the queue, if the TPA delays the response phase, there is a window of opportunity for a buffer conflict between two transactions. This erratum can only occur if there is a PCI-to-TPA zero-length read transaction followed by a PCI to memory read transaction under high V<sub>CC</sub> operation conditions (typically above 3.3 V).

**IMPLICATIONS:** Zero-length requests from PCI to a TPA may result in a buffer conflict and data corruption.

**WORKAROUND:** Avoid zero-length requests to the TPA. Enable the PWE (Pre-fetch Word Enable) bit in the PXB to override any zero-length PCI reads to a TPA. Alternatively, do not use PCI devices which issue zero-length requests.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

### 13. Response Optimization for Zero-Length Read Requests Causes MIOC Hang

**PROBLEM:** The system bus protocol allows zero-length responses to be given when DBSY# is active and the previous transaction has not completed its data phase. If a third party agent (TPA) implements this response optimization for zero-length requests from the MIOC, and the following sequence of events occurs:

1. A read request with data (or read with HITM#) from the MIOC targets a TPA,
2. One or more zero-length reads from the MIOC target the TPA and do not result in an assertion of HITM#, and
3. A read request with data (or read with HITM# asserting) from the MIOC targets the TPA,

the TPA will then drive a data response to the transaction that followed the zero-length transactions without a wait cycle. This will cause the MIOC to lose the data for the transaction in step 3, above.

**IMPLICATION:** If this erratum occurs, the system will fail unpredictably. However, Intel has only observed this erratum in a focused testing environment.

**WORKAROUND:** Avoid zero-length requests to the TPA. Enable the PWE (Pre-fetch Word Enable) bit in the PXB to override any zero-length PCI reads to a TPA. Alternatively, do not use PCI devices which issue zero-length requests. Do not implement response optimization in TPA.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

### 14. PXB PCI PLLs Show Sensitivity To Analog Vcc Levels

**PROBLEM:** The two PLLs that provide PCI clocks fail to lock properly and to provide stable clocks at an analog supply voltage of greater than 2.5 V. The PLLs require an external filter which is used to block high frequencies. The nominal voltage applied to the filter is 3.3 V. The PLLs associated with the analog supply pins  $V_{CCA0}$  (A27) and  $V_{CCA1}$  (C27) will not be stable at this voltage. This condition does not affect the PLL associated with  $V_{CCA2}$  which operates properly at 3.3 V.

**IMPLICATIONS:** PCI transactions will not be processed correctly if the PLL does not lock or becomes unstable at 33 MHz, resulting in unpredictable behavior.

**WORKAROUND:** The following circuit, (which includes the required low-pass filter) should be used to lower the supply voltage at the analog supply pins  $V_{CCA0}$  (A27) and  $V_{CCA1}$  (C27) of the component.

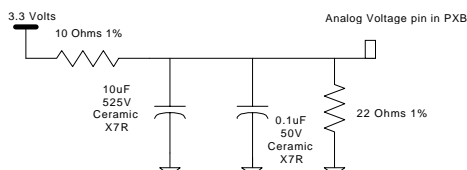


Figure 1. Workaround Circuit for Erratum 14

The circuit is a 10Ω resistor connected to a 3.3 V source. The other side of the resistor connects to a 22Ω resistor, a 10 μF capacitor and a 0.1 μF capacitor to ground. It also connects to the PXB analog supply pin ( $V_{CCA0}$ ) at this point. A similar circuit should be used for the second PCI PLL connected to pin  $V_{CCA1}$ . Note that the supply voltage provided to the third PLL pin  $V_{CCA2}$  (E27) **must** remain at 3.3 V.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.



## 15. *DBSY# Causes Hang Condition on Inbound Transaction With HITM#*

**PROBLEM:** The Intel 450NX PCIsset is not able to handle relaxed DBSY# deassertion (DBSY# kept asserted after DRDY# deassertion) on inbound transactions which receive an implicit writeback from a third party agent (TPA).

**IMPLICATIONS:** Systems that contain a TPA may hang if the TPA performs relaxed DBSY# deassertion following an implicit writeback. Relaxed DBSY# deassertion affects only one transaction case where inbound reads targeted to memory receive HITM# from the processor or a TPA. Note that this occurs only during implicit writeback data transfer.

**WORKAROUND:** None identified.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## 16. *MUX Originates Glitches on DSTRB# Signals During Memory Data Bus Simultaneous-Switching*

**PROBLEM:** The MUX component of the Intel 450NX PCIsset may originate glitches of up to 200 mV in magnitude upon the memory data bus DSTRB# outputs. These glitches occur at a point in time corresponding to memory-bus data-bit switching, and are due to parasitic coupling between signals and inductively-generated  $V_{SS}$  noise. The depth and direction of the glitches are dependent on the number of data bits switching and the direction of the data transitions. Glitch depth is expected to be worst-case at lower operating temperatures and/or elevated bus  $V_{TT}$  voltage due to the elevated  $V_{SS}$  currents.

**IMPLICATIONS:** Induced glitches on the memory data bus strobe signals, especially the DSTRBP# signals, could cause double-strobing of the MIOC memory data-bus receivers, resulting in detection of uncorrectable memory-bus ECC errors. These errors will be recurring (“sticky”) until the next time the Intel 450NX PCIsset is reset, if additional system noise (such as termination noise, transmission-line reflections, and/or coupling noise) sources are also present.

**WORKAROUND:** To reduce noise sensitivity, follow good system-design practices to help eliminate or reduce noise sources. The following design practices are recommended:

- If possible, use a board plane or “island” for VTT. If VTT trace routing is used, it should be short, wide (not minimal trace width) and decoupled as closely as possible.
- Decouple VTT at both the termination resistors and the component to keep AC noise to a minimum. Specifically, use at least one 0.1 •F or 1.0 •F capacitor for each of the four RTT resistors, plus ten 100 •F bulk termination capacitors at each end of the bus. Use four 0.1 •F capacitors (in 0805 or 1206 packages) plus one additional 22 •F capacitor for the MUX component.
- Maintain the specified voltages for VTT (1.5 V  $\pm$ 3% DC,  $\pm$ 9% overall).
- Ensure that signal return-paths (reference planes) do not have large loop areas or significant supply-plane noise due to distribution across connectors.
- Use connectors that have been demonstrated to work at 100 MHz signaling frequency, with supply-pins interleaved frequently to provide the best possible high-frequency response.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## 17. *MIOC Noise Sensitivity on DSTRB# Inputs*

**PROBLEM:** The MIOC component has less-than-anticipated input-high noise margins on the memory bus DSTRB# signals. In particular, if glitches of up to or exceeding 300 mV downwards from high levels are

presented on the DSTRBP# inputs, the MIOC may observe these glitches as additional strobe-low transitions, incorrectly clocking portions of the memory data-bus source-synchronous logic. However, testing components to these input-high levels, even under heavy simultaneous-switching traffic scenarios, may not trigger such failures.

**IMPLICATIONS:** If this erratum occurs due to glitches on the DSTRB# signals, the resulting misalignment in the MIOC's source-synchronous buffer pointers may result in memory reads returning corrupted data and/or flagging ECC errors.

**WORKAROUND:** To minimize additional system noise sources on these signals, follow good system-design practices. The following practices are recommended:

- Maintain the specific voltages for VREF ( $(2/3 \cdot V_{TT}) \pm 2\%$ ). The use of three matched resistors to minimize divider variance is encouraged.
- Create a local copy of the VREF net for each component.
- Decouple VTT at both the termination resistors and the component to keep AC noise to a minimum. Specifically, use at least one 0.1  $\mu$ F or 1.0  $\mu$ F capacitor for each of the four RTT resistors, plus ten 100  $\mu$ F bulk termination capacitors at each end of the bus. Use twelve to fifteen 0.1  $\mu$ F capacitors (in 0805 or 1206 packages) plus four additional 22  $\mu$ F capacitors for the MIOC component.
- Minimize additional system coupling on the memory-bus DSTRB# lines by following good signal routing; reference signals to planes which provide good current-return paths with a small loop area.
- Memory-bus strobe lengths should be designed so that any driver SSO-induced glitches will not arrive at the MIOC coincident with any transmission-line reflection.
- Additional noise margin may be gained with respect to the MIOC VREF level by slightly reducing the DSTRB# termination values and/or slightly increasing the termination-voltage for the DSTRB# signals only.
- Assess margins by using bus exerciser tests and determining the highest and lowest levels of VREF at which tests do not exhibit "sticky" ECC failures. From 100 to 120 mV of additional VREF headroom above the nominal level is desirable.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## 18. *Erroneous Checking of Expander Bus Parity By the MIOC*

**PROBLEM:** The MIOC component erroneously checks parity on the expander (F16) bus during the turn-around cycle immediately following a PXB-driven packet and the start of an MIOC-driven packet. If a very specific timing relationship exists between the MIOC's clock and the arriving XSTRB# signals exists (XSTRBN# strobe falling-edge arrives at the MIOC right coincident with the start of the data bus being driven by the MIOC), data and parity sampled by the strobes may be inconsistent due to bit-skew and/or metastability. An F16 Bus Parity Error may be erroneously flagged.

**IMPLICATIONS:** An Expander (F16) Bus error, when detected, will cause an F16 Bus Parity Error to be flagged in the MIOC's ERRSTS register and displayed on the ERR# [1:0] monitor pins. While no data corruption would result, the erroneous parity check and resulting failure may result in a BINIT# assertion by the chipset when BINIT# assertion is enabled.

**WORKAROUND:** The failure band is narrow enough that slightly varying XSTBN# falling-edge arrival with respect to MIOC's data-output timing can prevent the failure from occurring. Delaying XnCLK from the MIOC to the PXB can vary this timing and can be accomplished by adding extra loading or trace-routing to the XnCLK signal routing (if adjusting the entire F16-bus routing is not feasible due to existing board layout). To assist in designing board layout for avoidance of this problem, consider calculating the sum of the following system timing parameters and comparing against the limits of the failure band or timing "window."

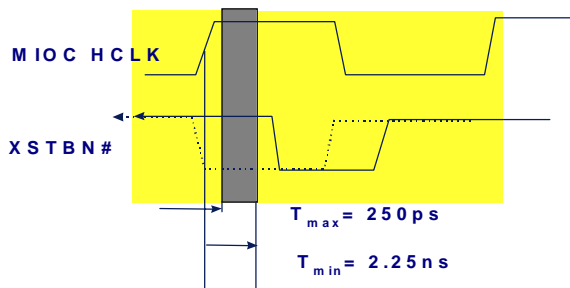
$$HCLK / XpClkFB \text{ Skew} + XpClkFB / XpClk \text{ Skew} + T_{66} + XpXSTBN\# \text{ Flight time} \leq (T_{PERIOD} + 250 \text{ ps}), \text{ or } > (T_{PERIOD} + 2.25 \text{ ns})$$

Where:

HCLK/XpClkFB Skew	is the rising edge skew value ( $\pm 250$ ps typical)
XpCLKFB/ XpCLK Skew	is a value currently specified to be $\pm 125$ ps
T66*	is the PXB XCLK-to-XpXSTBN# delay value
XpXSTBN# flight time	is a function of the OEM's Expander (F16) Bus trace length

\*Use a typical value of 8.6 ns for T66 in the above calculations.

Implementing this workaround may require deviating from the clock-skew specification. The skew between XpClkFB at the MIOC and XpCLK at the PXB may need to increase to account for a delay to the XpCLK line. Note that due to min/ max parameter variances, it may not be possible to design a workaround which can guarantee avoidance of the failing timing scenario, at least not without violating other timings. If this is the case, calculating the above timing and comparing min/ typ/ max values may still provide some assistance in designing a system workaround which could minimize exposure to this problem.



**Figure 2. Timing Window Where Strobe Falling Edge Must Be Avoided**

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## 19. PIIX4E Controller and PXB May Livelock When Using Certain DMA Transfer Modes

**PROBLEM:** A livelock condition may occur in systems using the PIIX4E controller and the Intel 450NX PCIsset when using bus-mastering IDE, Ultra DMA/33 and Type-F ISA DMA transfer types. If these transfer modes are used, the PIIX4E will not accept any outbound I/O read transactions until receiving a PHLDA# assertion from the Intel 450NX PCIsset and confirming completion of its DMA transfer across the PCI bus to system memory.

The current Intel 450NX PCIsset does not complete the PHLD#-PHLDA# protocol (by asserting PHLDA#) to allow the PIIX4E to complete its DMA transfer if outbound I/O transactions are pending in the PXB. This results in a livelock condition.

**IMPLICATIONS:** If devices such as IDE hard drives, CD-ROM drives, or ISA cards use any of the above DMA transfer modes, the system may hang due to this livelock condition. Agents attached to the ISA bus will not induce the problem unless the ISA DMA used is Type-F. Also, using the PIIX4E onboard IDE in PIO mode does not demonstrate the problem, since the data in this case is processed by the PIIX4E in a way which does not affect the outbound retry mechanism.

**WORKAROUND:** Do not use devices which utilize the affected DMA modes in systems configured with the affected PXB components. Affected IDE devices could be supported by using an appropriate PCI plug in card.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## 20. *PXB Arbiter Can Cause Deadlock Condition in Bus Lock Mode*

**PROBLEM:** If bus lock mode in the PXB is enabled, the PXB may cause a system deadlock. This condition may occur if there is one (32/64 Bit Mode) outbound read that is continuously retried while the PIIX4E is making a request to the PXB.

**IMPLICATIONS:** If PCI bus locked mode is enabled on the PXB, contention between two arbitration states of the PXB and the PIIX4E device may occur under retry activity with an outbound read and a request made by the PIIX4E.

**WORKAROUND:** Disabling bus lock mode in the PXB configuration space (CONFIG register, bit 14) works around this erratum even if there are no locked operations involved in the deadlock. In addition, the PXB should allow the PIIX4E to be granted the bus even if outbound reads are pending in its buffers. The bus lock mode is defined in the Intel 450NX PCISSET solely to provide compatibility with a legacy PIIX4E south bridge requirement. However, this mode is not necessary in systems based on the Intel 450NX PCISSET, since the architecture guarantees that the PIIX4E will maintain bus ownership once it has acquired it. The MIOC prevents any outbound transaction which might require use of the PCI bus from even being accepted once it decides to send an SSBR-acknowledge back to the PIIX4E.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## 21. *ERR#[1:0] Pins Do Not Reflect ERRSTS, HEL, MEL State During BINIT# Initiated Resets*

**PROBLEM:** The ERR#[1:0] pins should reflect the state of the ERRSTS, MEL, and HEL internal configuration register values which in turn should be “sticky” during BINIT#-initiated chipset reset. However, the internal “sticky” error register bits on the ERRSTS register are reset by the MIOC reset signal, which is asserted during BINIT#-initiated resets. The assertion of the reset signal from the MIOC causes the ERR# state to be masked during the internal reset. After the internal reset occurs, the output of ERR#[1:0] shows the correct values.

**IMPLICATIONS:** Systems that sample the status of the ERR#[1:0] during BINIT# initiated resets may get incorrect data from the ERRSTS register value.

**WORKAROUND:** None identified.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## 22. *Device Drivers That Violate the Producer/Consumer Model May Provide Invalid Data*

**PROBLEM:** The re-streaming feature of the PXB PCI bridge substantially improves the sustainable PCI Bandwidth for long bursts in situations where either the PXB or PCI master disconnects before the burst is complete. However, if there is not strict adherence to the “Producer-Consumer” programming model, the re-streaming algorithm may result in invalid data being provided to a PCI master.

**IMPLICATIONS:** The Producer-Consumer model uses a flag or semaphore to communicate between the producer of data (for example a processor writing to memory) and a consumer of data such as a PCI DMA device. The data to be consumed is first placed in a designated location, then a flag is set in another designated location. The consumer must not begin to consume the data until the flag is set, and the producer must not set the flag until the data is ready. The producer also must not alter the data once it has set the flag. When the consumer observes the flag set, it is free to consume the data. The consumer resets the flag when it has finished consuming the data. The producer is then free to modify the data in preparation for the next transfer. The re-streaming algorithm allows the data and flag to be located anywhere in the system. For example, they

can both be in memory or they may be split between memory and the I/O system. The only requirement is that there must be a new flag for each transfer of new data. The flag is observed by the PCI master either by reading from memory, from another I/O location, or from an internal register. The producer must set the flag in memory, in an I/O location, or directly onto a register in the PCI master.

**WORKAROUND:** Any device that does not follow the Producer-Consumer programming model (as specified above) may supply invalid data to the PCI master. Drivers that modify data while a transfer to a PCI master is in progress must ensure that PXB re-streaming is not enabled.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## ***23. Nonascending Cacheline Evictions to Write Combining PCI Targets May Result in Data Corruption***

**PROBLEM:** The Intel 450NX PCIsset assumes full cache line writes from write combining buffers (WCBs) in the processor to PCI memory begin with critical chunk zero. This assumption by the 450NX PCIsset does not affect partial writes out of the WCBs that are less than a cache line.

**IMPLICATIONS:** Devices that do not use WC memory, or whose WC memory is located in system memory space, are unaffected by this issue. Data corruption may occur if the WCBs are written in nonascending chunk order, the data is destined for PCI, and the entire cache line is evicted from the WCB. If only part of a cache line is evicted, data corruption will not occur. In a typical graphics environment, data is written to the WCB in ascending order and therefore there is no impact with respect to this issue. If graphics devices are written to in nonascending order, one possible result is pixel corruption. The impact of this issue on other devices depends upon the particular usage and adapters involved.

**WORKAROUND:** A device driver can avoid this issue by disabling the use of WC memory in the driver and/or operating system (this may have performance implications depending upon the device). Device drivers that have exclusive access to their adapter memory (i.e., it is not mapped into the address space of applications) can ensure that writes to the WC memory occur in ascending order only.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## ***24. Fifth Byte of PXB Performance Monitoring Data Registers Aliases Third Byte***

**PROBLEM:** The Intel 450NX PCIsset PXB Performance Monitoring Data (PMD) registers, which store the current count of events monitored on an associated PCI bus, do not count accurately beyond the fourth byte.

**IMPLICATIONS:** The PMD register is a five-byte register. The failure is observed when the counter increments from 000000FFFFh to 0000010000h (when the third byte of the counter becomes significant). In this instance, the erratum results in the five-byte counter value to be 0100010000h, where the fifth byte is incorrect.

**WORKAROUND:** It is possible for software to implement a workaround in which only the lower four bytes of the PMD registers are used. This results in a minor change to software where only the lower four bytes are read from the PCI configuration space of the PXB's. The upper byte must be ignored.

**STATUS:** For the steppings affected see the Summary Table of Changes at the beginning of this section.

## DOCUMENTATION CHANGES

### 1. *Description of MIOC CVCR/CVDR Registers Bit 10*

The *Intel 450NX PCIsset* datasheet (Order Number 243771-004) incorrectly documents the CVCR and CVDR registers, bit 10:

- Section 3.4.6 of the datasheet describes MIOC CVCR register bit 10 as follows:

Bit 10 Enable BINIT# Input.

Captured from A#[10]. If set, the MIOC will observe the assertion of the BINIT# input. Further details on BINIT# processing may be found in the ERRCMD register.

In actuality, this bit has no effect, and the MIOC will always observe BINIT# assertion regardless of the state of it.

- Section 3.4.7 of the datasheet describes MIOC CVDR register bit 10 as “reserved (0).” The correct definition is:

Bit 10 Enable BINIT# Input.

If set, A#[10] will be asserted during RESET#, and all system bus agents will enable BINIT# observation. This bit should be set under normal operation. Default = 0.

### 2. *Description of PXB Configuration Register Bit 1*

The *Intel 450NX PCIsset* datasheet (Order Number 243771-004) incorrectly documents the PXB Configuration register (40-41h) Bit 1

Section 3.4.4 of the datasheet describes PXB Configuration register bit 1 as follows:

Bit 1 Host/PCI Bus gearing ratio:

This is a read-Only bit that selects the system clock to PCI clock gearing ratio. This bit reflects the state of the GEAR4# strapping pin. This bit should be cleared (i.e., GEAR4# is high, or deasserted), resulting in a system clock/ PCI clock gearing ratio of 3:1.

Default =[GEAR4# pin].

The current Intel 450NX definition specifies PXB configuration register Bit 1 as a reserved bit, there is no GEAR4# pin functionality defined and accessible in the present specification.

### 3. *Description of MIOC/PXB Boundary Scan Functionality*

The *Intel 450NX PCIsset External Design Specification, Revision 3.1* (OR-1272) incorrectly documents the IEEE 1149.1 compliance of the PXB and MIOC.

Section 12.1.7.3 should include the following:

Due to the differential nature of the F16 expander strobes, XpXSTBP# and XpXSTBN#, setting of these signals via boundary scan is limited to valid, opposite values. Setting both the positive and negative strobes to the same value will result in unpredictable system behavior.