Intel® 80331 I/O Processor

Specification Update

April 2006

The Intel® 80331 I/O Processor may contain design defects or errors known as errata that may cause the product to deviate from published specifications. Current characterized errata are documented in this specification update.

Order Number: 273930-021US

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

Revision History

Date	Version	Description
April 2006	021	Added: • D-1 stepping information to all Summary tables and to Table 1, "Intel® 80331 I/O Processor Die Details" on page 17 and Table 2, "Intel® 80331 I/O Processor Device ID Registers" on page 18 • Status change of Erratum 62 • Status change of Specification Change 22 • Status change of Specification Clarification 33
February 2006	020	Added: • Non-Core Errata 63 • Specification Clarification 33
January 2006	019	Added: • Non-Core Errata 62 • Specification Change 22
December 2005	018	Added: • Specification Clarification 31 and 32 • Documentation Change 17
October 2005	017	Added: • Specification Clarification 29 and 30
August 2005	016	Added: • Specification Clarification 27 and 28 • Documentation Change 16.
June 2005	015	Added: • Non-Core Errata 61
May 2005	014	Added: • Non-Core Errata 60 • Specification Change 21 Updated: • Specification Change 13, 15, and 16
April 2005	013	Added: • Non-Core Errata 58 and 59
March 2005	012	Added: • Specification Change 19 and 20 • Documentation Change 15 • PB-free Markings to Table 1, "Intel® 80331 I/O Processor Die Details" on page 17 Updated: • Non-Core Errata 4 • Updated Non-Core Errata 50 Changed: • Status of Non-Core Errata 20 to Fixed

Date	Version	Description
January 2005	011	Added: • Non-Core Errata 57 • Specification Clarification 26 • Documentation Change 13 and 14
November 2004	010	Added: • Non-Core Errata 56 • Specification Clarification 25
September 2004	009	Added: • Specification Clarification 23 and 24. • Specification Change 18. • Documentation Change 12. Updated: • Specification Change 11 and Specification Clarifications 13 and 22.
August 2004	008	Added: • D-0 stepping information. • Non-Core Errata 55. • Specification Clarification 21 and 22. • Specification Change 17. Updated: • Non-Core Errata 51 and 54.
June 2004	007	Added: • Non-Core Errata 52 through 54. • Specification Clarification 19 and 20. • Documentation Change 11.
May 2004	006	Added: • Non-Core Errata 49 through 51 • Specification Clarification 17 and 18 • Specification Change 14 through 16 • Documentation Change 10 Updated: • Spec Change 12; Spec Clarification 13
April 2004	005	Added: • Non-Core Errata 46 through 48. • Specification Clarification 13 through 16. • Documentation Change 8 and 9. • Added C-1 to errata table Updated: • Non-Core Errata 45. • Specification Change 11. • Specification Clarification 11 and 12. • Documentation Change 3. • Status information for Non-Core Errata.

Date	Version	Description
March 2004	004	Added: • Errata 37 through 45. • Specification Change 10 through 13. • Specification Clarification 9 through 12. • Documentation Change 5 through 7. Updated: • Erratum 35.
January 2004	003	Added: • Errata 33 through 36. • Specification Change 9. • Specification Clarification 8. • Documentation Change 3 and 4. Updated: • Erratum 31.
November 2003	002	Added: • Errata 27 through 32. • Specification Changes 7 and 8. • Specification Clarification 6 and 7. • Documentation Changes 1 and 2. Updated: • Errata 6, 21 and 26.
September 2003	001	Initial Release.

Preface

This document is an update to the specifications contained in the Affected Documents/Related Documents table below. This document is a compilation of device and documentation errata, specification clarifications and changes. It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools.

Information types defined in Nomenclature are consolidated into the specification update and are no longer published in other documents.

This document may also contain information that was not previously published.

Affected Documents/Related Documents

Title	Order
Intel® 80331 I/O Processor Developer's Manual	273942
Intel® 80331 I/O Processor Datasheet	273943
Intel® 80331 I/O Processor Design Guide	273823

Nomenclature

Errata are design defects or errors. These may cause the Intel® 80331 I/O Processor[1] behavior to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.

Specification Changes are modifications to the current published specifications. These changes will be incorporated in any new release of the specification.

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 any new release of the specification.

Documentation Changes include typos, errors, or omissions from the current published specifications. These will be incorporated in any new release of the specification.

Note: Errata remain in the specification update throughout the product's lifecycle, or until a particular stepping is no longer commercially available. Under these circumstances, errata removed from the specification update are archived and available upon request. Specification changes, specification clarifications and documentation changes are removed from the specification update when the appropriate changes are made to the appropriate product specification or user documentation (datasheets, manuals, etc.).

1. ARM* architecture compliant.

Summary Table of Changes

The following table indicates the errata, specification changes, specification clarifications, or documentation changes which apply to the Intel® 80331 I/O Processor. Intel may fix some of the errata in a future stepping of the component, and account for the other outstanding issues through documentation or specification changes as noted. This table uses the following notations:

Codes Used in Summary Table

Stepping

X:	Errata exists in the stepping indicated. Specification Change or Clarification that applies to this stepping.
(No mark) or (Blank box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.

Page

(Page):	Page location of item in this document.

Status

Doc:	Document change or update will be implemented.
Plan Fix:	This erratum may be fixed in a future stepping of the product.
Fixed:	This erratum has been previously fixed.
No Fix:	There are no plans to fix this erratum.

Row

	Change bar to left of table row indicates this erratum is either new or modified from the previous version of the document.

Non-Core Errata (Sheet 1 of 3)

No.	Steppings						Page	Status	Errata
	A-1	B-0	C-0	C-1	D-0	D-1			
1	X	X	X	X	X	X	19	No Fix	CAS latency of three not supported for DDR-II On-Die Termination (ODT)
2	X						19	Fixed	Upper PCI signals on Secondary PCI bus are not driven low during reset
3	X						19	Fixed	Memory Controller Unit does not properly support 32-bit memory configurations
4	X	X	X	X	X	X	19	No Fix	Legacy Power Fail Mechanism does not work
5	X						20	Fixed	Boundary Scan data gets inverted
6	X						20	Fixed	BIU interrupt does not occur on Internal Bus Write Master Abort
7	X						20	Fixed	CRC value calculated by DMA unit is not compliant with iSCSI
8	X						20	Fixed	ATU Outbound Direct Window overlaps with PBI exception vectors
9	X	X	X	X	X	X	20	No Fix	S_REQ64# Initialization Pattern Timing Violation in PCI-33 Mode
10	X						21	Fixed	PCI-66 Mode violates PCI AC Timings
11	X						21	Fixed	Reserved bits in the Modem Status Register incorrectly generate interrupts
12	X						21	Fixed	P_REQ# not de-asserted during idle
13	X						21	Fixed	Chassis/Slot PCI Extended Capability is not valid
14	X						22	Fixed	SDCR0.2 implemented as ‘Reserved’
15	X						22	Fixed	32-bit region missing proper address decode
16	X						22	Fixed	S_GNT[3:2]# outputs are not pulled high when Bridge is disabled.
17	X	X	X	X	X	X	22	No Fix	Split Transaction Commitment limit register mechanism, in the PCI-X bridge, does not operate as implied by the PCI-X Addendum to the PCI Local Bus Specification, Revision 1.0a
18	X	X	X	X	X	X	23	No Fix	Watchdog time-outs by the PCI-X bridge may cause data corruption
19	X						23	Fixed	Discard timer expiration on delayed read can cause data corruption or deadlock when data is still being received on target bus
20	X						23	Fixed	Configuration cycle attribute parity error signaled incorrectly by PCI-X bridge
21	X	X					23	Fixed	Transactions are buffered during Secondary Reset
22	X						24	Fixed	Primary bus pin mode behavior incorrect during reset in the 80331 no bridge mode
23	X						24	Fixed	P_REQ# pin mode behavior when in the 80331 no bridge mode
24	X	X					24	Fixed	Master abort after data transfer within a single ADB on PCI-X to PCI-X read block transactions may cause data corruption or deadlock

Non-Core Errata (Sheet 2 of 3)

No.	Steppings						Page	Status	Errata
	A-1	B-0	C-0	C-1	D-0	D-1			
25	X	X	X	X	X	X	25	No Fix	Boundary scan multi-chip module implementation
26	X	X	X	X	X	X	25	No Fix	Auto refresh command also generates a Precharge All command on DDR bus
27	X						26	Fixed	SERR# set to incorrect voltage
28	X						26	Fixed	M66EN set to incorrect voltage
29	X						26	Fixed	P_INT[D:A]# not operating correctly
30	X						26	Fixed	Secondary bus may not initialize correctly at 100 MHz PCI-X or 133 MHz PCI-X
31	X	X	X	X	X	X	26	No Fix	Coalesced writes to 32-bit memory can cause data corruption
32	X	X	X	X	X	X	27	No Fix	ATU passing rules operation in PCI mode
33	X	X					27	Fixed	S_INT[D:A]# pull-ups disabled by Internal Bus Reset
34	X	X					27	Fixed	MCU Preemption Control does not properly manage the byte count
35	X	X					28	Fixed	I^2C unit hang condition
36	X	X	X	X	X	X	28	No Fix	VPD Data Register bit 19 is not read/write
37	X	X					29	Fixed	Intel XScale® Core lockup condition
38	X	X					30	Fixed	32-bit region write corrupts ECC immediately after 64-bit Read-Modify-Write
39	X						30	Fixed	Reset straps incorrectly sampled on the secondary reset
40	X	X					31	Fixed	PCI-X to PCI Memory Read double-word near 1MB boundary may cause the system to hang
41	X	X					31	Fixed	PCI-X to PCI Memory Read Block across 1MB boundary may cause data corruption
42	X	X	X	X	X	X	32	No Fix	DMA CRC result is byte reversed
43	X	X	X	X	X	X	32	No Fix	CRC corruption on PCI-to-local DMA transfers
44	X	X	X	X			32	Fixed	Byte Count Modified bit set to 1
45	X	X	X	X			33	Fixed	Corrupted byte count and data when crossing 1 MByte boundary in PCI-X to PCI mode
46	X	X	X	X			33	Fixed	PCI-X to PCI Memory Read with 4 K byte count and unaligned starting address
47	X	X	X				34	Fixed	VCCDDR (VCC25/VCC18) Current Spike
48	X	X	X	X			35	Fixed	Bridge PCI ordering rule violation
49	X	X	X	X	X	X	35	No Fix	ATU claims PCI commands 8 and 9 when issued as Dual Address Cycle (DAC)
50	X	X	X	X	X	X	36	No Fix	Secondary bus PCI RST# pulse prior to the rising edge of P_RST#
51	X	X					36	Fixed	Slow edge rates observed when 80331 is driving the primary bus
52	X	X	X	X			37	Fixed	Enabling the core-to-memory port can cause a stall condition

Non-Core Errata (Sheet 3 of 3)

No.	Steppings						Page	Status	Errata
	A-1	B-0	C-0	C-1	D-0	D-1			
53	X	X	X	X	X	X	38	No Fix	PCI-to-PCI read flow-through with destination TRDY# stalls can cause data corruption
54	X	X	X	X			38	Fixed	S_PCIXCAP PCI mode threshold is too high
55	X	X	X	X	X	X	39	No Fix	No support for burst I/O and configuration read/writes
56	X	X	X	X	X	X	39	No Fix	Read flow through hangs due to disconnect without data at a buffer boundary.
57	X	X	X	X	X	X	39	No Fix	P_SERR# not asserted for parity error on AD[63:32]
58	X	X	X	X	X	X	39	No Fix	Bus Interface Unit (BIU) claims DAC addresses in the range of the Memory Mapped Registers (MMR
59	X	X	X	X	X	X	40	No Fix	PCIX-to-PCI Memory Read issued as 32-bit, then retried as 64-bit
60	X	X	X	X	X	X	41	No Fix	Tc1(min) of the PCI-X clock observed to be marginally less than the requirement specified for the PCI-X (Mode 1, class1) clock jitter.
61	X	X	X	X	X	X	41	No Fix	I2C Control Register reset bit does not function
62					X		41	Fixed	Internal Clock Misalignment Can Cause Processor Hang.
63	X	X	X	X	X	X	41	No Fix	Spurious DMA0 End-Of-Transfer Interrupt

Core Errata

No.	Steppings					Page	Status	Errata
	A-1	B-0	C-0	D-0	D-1			
1	X	X	X	X	X	43	No Fix	Abort is missed when lock command is outstanding
2	X	X	X	X	X	43	No Fix	Aborted Store that Hits the Data Cache May Mark Write-Back Data as Dirty
3	X	X	X	X	X	44	No Fix	Performance Monitor Unit event 0x1 can be incremented erroneously by unrelated events
4	X	X	X	X	X	44	No Fix	In Special Debug State, back-to-back memory operations — where the first instruction aborts — may cause a hang
5	X	X	X	X	X	45	No Fix	Accesses to the CP15 ID register with opcode2 > 0b001 returns unpredictable values
6	X	X	X	X	X	45	No Fix	Disabling and re-enabling the MMU can hang the core or cause it to execute the wrong code
7	X	X	X	X	X	46	No Fix	Updating the JTAG parallel registers requires an extra TCK rising edge
8	X	X	X	X	X	46	No Fix	Non-branch instruction in vector table may execute twice after a thumb mode exception

Specification Changes

No.	Steppings					Page	Status	Specification Changes
	A-1	B-0	C-x	D-0	D-1			
1	X	X	X	X	X	47	Doc	HPI# (High Priority Interrupt) is a maskable interrupt
2	X	X	X	X	X	47	Doc	PCIODT_EN Reset Strap Signal
3	X	X	X	X	X	47	Doc	LOCK# functionality has been de-featured
4	X	X	X	X	X	47	Doc	Watchdog Timer and Retry Timer has been de-featured
5		X	X	X	X	47	Doc	P_GNT# and P_REQ# signals have new ball locations on B-0
6	X	X	X	X	X	47	Doc	Peripheral Performance Monitor Unit has been de-featured
7	X	X	X	X	X	48	Doc	Processor Device ID has been removed
8	X	X	X	X	X	48	Doc	1.5K pull-down required on AD[15] of the PBI bus
9	X	X	X	X	X	48	Doc	OCD and Receive Enable calibration de-featured
10		X	X	X	X	48	Doc	New Watchdog Timer (WDT) functionality in B-0 stepping
11	X	X	X	X	X	48	Doc	PWRDELAY needs only a pull-up for battery back-up mode
12	X	X	X	X	X	48	Doc	ARB_EN signal has been de-featured
13	X	X	X	X	X	48	Doc	Intel® 80331 I/O Processor Design Guide change for Unbuffered DDR-I dual-banked DIMMs
14	X	X	X	X	X	48	Doc	DDRRES2 can be pulled down to reduce current during self-refresh
15	X	X	X	X	X	48	Doc	Intel® 80331 I/O Processor Design Guide change for Peripheral Bus Interface (PBI)
16	X	X	X	X	X	49	Doc	Intel® 80331 I/O Processor Design Guide change for PCI/-X busses
17				X	X	49	Doc	Internal bus operates at 333 MHz for D-0 stepping
18				X	X	49	Doc	Application Accelerator Unit enhanced for D-0 stepping
19	X	X	X	X	X	49	Doc	Recommended DLL register values
20	X	X	X	X	X	49	Doc	DDR-II JEDEC initialization sequence includes writes to EMRS2 and EMRS3
21	X	X	X	X	X	50	Doc	Case temperature (Tcase) change
22				X		50	Fixed	Internal Clock Misalignment.

Specification Clarifications

No.	Steppings					Page	Status	Specification Clarifications
	A-1	B-0	C-x	D-0	D-1			
1	X	X	X	X	X	51	No Fix	64 MB and 2 GB DDR333 capacities not to be tested in post-silicon validation
2	X	X	X	X	X	51	No Fix	DDR-II 400 Unbuffered DIMMs are not supported
3	X	X	X	X	X	51	No Fix	Interrupt behavior in the 80331 no bridge mode
4	X	X	X	X	X	51	No Fix	Memory map for 2 GByte of DDR memory
5	X	X	X	X	X	51	No Fix	Back to back MCU MMR reads
6	X	X	X	X	X	52	No Fix	Write requirements for the Peripheral Bus Interface
7	X	X	X	X	X	52	No Fix	PCI-X Status Register during PCI mode
8	X	X	X	X	X	52	No Fix	M_RST# driven to DDR-II or DDR-I voltage levels
9	X	X	X	X	X	52	No Fix	BIU master abort causes two interrupts on reads.
10	X	X	X	X	X	53	No Fix	Reset Internal Bus (PCSR.5) usage
11	X	X	X	X	X	53	No Fix	No Bridge Mode (BRG_EN) validation
12	X	X	X	X	X	53	No Fix	Potential race condition with Interrupt Controller Unit status bits
13	X	X	X	X	X	54	No Fix	Bus Interface Unit follows PCI ordering rules
14	X	X	X	X	X	55	No Fix	UART, I^2C and GPIO memory mapped registers should be addressed with 32-bit accesses
15	X	X	X	X	X	55	No Fix	Flash Wait States
16	X	X	X	X	X	55	No Fix	UART Interrupt Identification Register
17	X	X	X	X	X	55	No Fix	Reads on 16-bit PBI bus operate as 32-bit
18	X	X	X	X	X	56	No Fix	Embedded Design Usage Model - Secondary PCI bus only
19	X	X	X	X	X	56	No Fix	3.3 V to 1.5 V leakage
20	X	X	X	X	X	56	No Fix	Accessing “reserved” registers in “no bridge” mode
21	X	X	X	X	X	56	No Fix	Power plane isolation for Battery Back-Up (BBU) mode
22	X	X	X	X	X	57	No Fix	AAU result can be written directly to PCI host memory
23	X	X	X	X	X	57	No Fix	PWRDELAY functionality during power sequencing
24	X	X	X	X	X	57	No Fix	PBI lockout condition
25				X	X	57	No Fix	Interleaving descriptors with D-0 AAU
26	X	X	X	X	X	58	No Fix	RCVDLY setting for DDR-I memory
27	X	X	X	X	X	58	No Fix	ATUBAR3 Functionality
28	X	X	X	X	X	58	No Fix	VREF isolation for Battery Back-up (BBU) mode
29	X	X	X	X	X	58	No Fix	I2C Unit Enabling
30	X	X	X	X	X	59	No Fix	DMA transactions from local memory to a conventional PCI target can complete out of order
31	X	X	X	X	X	59	No Fix	SBR1 Programming with Bank 1 Unpopulated
32	X	X	X	X	X	59	No Fix	32-bit Writes to Unaligned 64-bit Addresses are Promoted to 64-bit Aligned Writes
33	X	X	X	X		60	Fixed	Case Temperature Clarification.

Documentation Changes

No.	Document Revision	Page	Status	Documentation Changes
1	273942-002	61	Doc	Table 236 shows incorrect description for tWDL field
2	273942-002	61	Doc	Processor Device ID should be removed
3	273942-002	61	Doc	Core Performance Monitoring Unit (CPMON) section is incorrect
4	273942-002	61	Doc	UART FIFO Occupancy Register is read only
5	273942-002	61	Doc	Refresh Frequency Register table is incorrect
6	273942-002	61	Doc	PBI interrupt should be removed
7	273942-002	62	Doc	ARB_EN has been de-featured
8	273942-002	62	Doc	Flash wait state information is incorrect
9	273942-002	63	Doc	SDCR0 and SDCR1 register updates
10	273823-001	66	Doc	Power sequence timing
11	273943-001	66	Doc	Updated Peripheral Bus Interface (PBI) timings
12	273942-002	66	Doc	ATU PM_CAP[5] text must be updated
13	273942-002	67	Doc	WDTCR also affected by TMR1[3]
14	273942-002	67	Doc	Split Completion Message clarification
15	273823-002	67	Doc	Figure 49 in Design Guide shows incorrect trace length
16	273943-001	67	Doc	Wrong Voltage Values in Table 21
17	273942-002	67	Doc	SBR1 Programming When Bank 1 is Unpopulated

Identification Information

Die Details

Table 1. Intel® 80331 I/O Processor Die Details

Stepping	Part Number	QDF (Q)/ Specification Number (SL)	Processor Speed (MHz)	Notes
A-1	NQ80331M500	Q623	500	Engineering Samples
B-0	NQ80331M500	Q651	500	Engineering Samples
C-0	NQ80331M500	Q723	500	Engineering Samples
C-0	NQ80331M667	Q726	667	Engineering Samples
C-0	NQ80331M800	Q729	800	Engineering Samples
C-1	NQ80331M500	Q870	500	Engineering Samples
C-1	NQ80331M667	Q871	667	Engineering Samples
C-1	NQ80331M800	Q872	800	Engineering Samples
C-1	NQ80331M500	SL7NM, SL82R	500	Production
C-1	NQ80331M667	SL7NN, SL82S	667	Production
C-1	NQ80331M800	SL7NP, SL82T	800	Production
D-0	NQ80331M500 QG80331M500	Q006 Q105	500	Engineering Samples PB-free Eng Samples
D-0	NQ80331M667 QG80331M667	Q007 Q106	667	Engineering Samples PB-free Eng Samples
D-0	NQ80331M800 QG80331M800	Q008 Q107	800	Engineering Samples PB-free Eng Samples
D-0	NQ80331M500 QG80331M500	SL823 (Tray), SL827 (T&R) SL8C5 (Tray), SL8C9 (T&R)	500	Production PB-free Production
D-0	NQ80331M667 QG80331M667	SL824 (Tray), SL828 (T&R) SL8C6 (Tray), SL8CA (T&R)	667	Production PB-free Production
D-0	NQ80331M800 QG80331M800	SL825 (Tray), SL829 (T&R) SL8C7 (Tray), SL8CB (T&R)	800	Production PB-free Production
D-1	NQ80331M500 QG80331M500	Q604 Q608	500	Engineering Samples PB-free Eng Samples
D-1	NQ80331M667 QG80331M667	Q605 Q609	667	Engineering Samples PB-free Eng Samples
D-1	NQ80331M800 QG80331M800	Q606 Q610	800	Engineering Samples PB-free Eng Samples

Table 1. Intel® 80331 I/O Processor Die Details

Stepping	Part Number	QDF (Q)/ Specification Number (SL)	Processor Speed (MHz)	Notes
D-1	NQ80331M500 QG80331M500	SL9B7 SL9BE	500	Production PB-free Production
D-1	NQ80331M667 QG80331M667	SL9B8 SL9BF	667	Production PB-free Production
D-1	NQ80331M800 QG80331M800	SL9B9 SL9BG	800	Production PB-free Production

Table 2. Intel® 80331 I/O Processor Device ID Registers

Device and Stepping	Processor Device ID (CP15, Register 0 – opcode_2=0)	Revision ID	JTAG Device ID and PDIR
A-1	0x69054090	0x00	0x09279013
B-0	0x69054094	0x04	0x49279013
C-0	0x69054096	0x06	0x69279013
C-1	0x69054097	0x07	0x69279013
D-0	0x6905409A	0x0A	0xA9279013
D-1	0x6905409A	0x0A	0xA9279013

NOTE: Processor core speed can be identified by reading CCLKCFG[3:0] (CP14, register 6)
500 MHz = 0x0, 667 MHz = 0x2, 800 MHz = 0x4

Non-Core Errata

1. CAS latency of three not supported for DDR-II On-Die Termination (ODT)

Problem: For DDR-II memory with a CAS Latency (CL) of three, the memory controller unit (MCU) does not provide the proper timing for the On-Die Termination signals (ODT[1:0]). The *JEDEC DDR-II SDRAM Specification*, September 2002, states that ODT must be driven one cycle prior to the write command, but the MCU does not meet this timing.

Implication: CAS latency of 3 is not supported in the 80331, therefore minimal performance impact as compared to CAS latency of 4.

Workaround: Use CAS latency = 4 or do not use ODT feature.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

2. Upper PCI signals on Secondary PCI bus are not driven low during reset

Problem: PCI requires the central resource to provide valid logic states on AD[63:32], C/BE[7:4]# and PAR64 during reset, and that they may only be driven to zero. The S_AD[63:32], S_C/BE[7:4]#, S_PAR64 on the secondary PCI bus are not driven to zero during secondary bridge PCI-X initialization; instead, these pins are tri-stated, “Z”. This is not a violation of the *PCI Local Bus Specification*, Revision 2.3.

Implication: Some PCI targets may not operate correctly, when they require these signals to be zero during reset.

Workaround: No workaround.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9.

3. Memory Controller Unit does not properly support 32-bit memory configurations

Problem: The memory controller unit (MCU) incorrectly stores the row address used in the page miss/page hit look up table. The result of this can cause the MCU to incorrectly determine a page hit or miss. When this occurs, it causes aliasing to an incorrect row of DDR SDRAM.

Implication: 32-bit memory cannot be used with A-1 stepping.

Workaround: No workaround. Use a 64-bit memory subsystem.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9.

4. Legacy Power Fail Mechanism does not work

Problem: For previous I/O processor generations, an external clock was required to maintain the incoming PCI clock (P_CLK) long enough for the power fail sequence to be sent to the memory. This is what is referred to as ‘legacy power fail’. The internal control circuit that enables the legacy power fail method is broken.

Implication: Legacy power fail cannot be used.

Workaround: For 80331, a new feature was added which keeps internal clocks running on power fail.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

5. Boundary Scan data gets inverted

Problem: Data driven in during boundary scan gets inverted during the capture phase. During parallel loading on a pin, an external zero on data in gets inverted to one for data out. This is a violation of the *IEEE 1149.1 Specification*.

Implication: Boundary scan cannot be used with A-1 stepping.

Workaround: No workaround. Do not use boundary scan.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9.

6. BIU interrupt does not occur on Internal Bus Write Master Abort

Issue: Software developers be aware that an interrupt is not generated when the Bus Interface Unit gets master aborted on an internal bus write.

Implication: When left undetected, it could cause data corruption.

Workaround: The status bit needs to be polled by software.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9

7. CRC value calculated by DMA unit is not compliant with iSCSI

Problem: Several issues with the CRC generation engine have caused it to not be compliant with iSCSI in A-1 step.

Implication: CRC cannot be used with A-1 stepping.

Workaround: No workaround.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9.

8. ATU Outbound Direct Window overlaps with PBI exception vectors

Problem: When the PBI exception vectors are enabled (PBCR.3), it claims addresses 00h-3fh. When the ATU outbound direct window is enabled, it claims addresses 28h-7ffffffh. There is an overlap between 28h-3fh, where both devices claim the internal bus transaction. For writes, this may not be an issue, since the only transaction here is a posted write to the ATU. For reads, the PBI handles the transaction as a single data phase disconnect, and the ATU internally handles it as a split transaction.

Implication: The agent which issued the internal bus transaction does not claim the returning data, since the transaction has been completed by the PBI. This causes the data from PCI to not be returned to the issuing internal bus agent.

Workaround: Consider address range of 28h-3Fh reserved.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9.

9. S_REQ64# Initialization Pattern Timing Violation in PCI-33 Mode

Problem: The *PCI Local Bus Specification*, Revision 2.3 states RST# to REQ64# Hold time is 0 ns minimum and 50 ns maximum. The 80331 drives the REQ64# signal for three cycles after RST# deasserts. Therefore, in PCI-33 mode, this is 90 ns, which violates the 50 ns maximum.

All other PCI and PCI-X modes are within the specification (i.e., PCI-66 is 45 ns).

Implication: Some PCI targets may not interpret the REQ64# signal correctly, causing it to operate in 32-bit mode.

Workaround: When running in PCI-33 mode, make sure PCI target can handle the extra 40 ns.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

10. PCI-66 Mode violates PCI AC Timings

Problem: The *PCI Local Bus Specification*, Revision 2.3 states a setup time of 3 ns for PCI-66 mode. The 80331 A-1 step has estimated the setup time to be in the range of 3.5 ns to 4 ns, which violates the *PCI Local Bus Specification*, Revision 2.3.

Implication: Some PCI 66 MHz targets may not function correctly with A-1 stepping.

Workaround: Use PCI-33 or PCI-X modes for initial development on A-1 stepping.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9.

11. Reserved bits in the Modem Status Register incorrectly generate interrupts

Problem: Bits[3:1] in the UART Modem Status Register are reserved bits. These reserved bits are incorrectly tied to the asserted state therefore generate false interrupts when the Modem Interrupt Enable bit (UxIER.3) is set. However, the reserved bits only generate the false interrupt once. This happens the cycle after the UART unit is enabled (UxIER.6) and the Modem Status Interrupts are enabled (UxIER.3). Once the Modem Interrupt Register is read no more false interrupts occur.

Implication: A false interrupt can occur.

Workaround: Software needs to read the Modem Status Register to determine proper interrupt sources.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9.

12. P_REQ# not de-asserted during idle

Problem: The P_REQ# signal stays asserted during bus idle instead of de-asserting as required by *PCI Local Bus Specification*, Revision 2.3. For example, when a PCI target asserts STOP#, the P_REQ# signal de-asserts one clock later than it should. The P_REQ# should have been de-asserted during the idle cycle.

Implication: May cause conflict with other PCI agents.

Workaround: Use a system arbiter that does not care when this condition occurs.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9.

13. Chassis/Slot PCI Extended Capability is not valid

Problem: The Chassis/Slot PCI extended capability cannot be configured, and therefore the capability is not valid and needs to be removed from the capability chain. The Chassis/Slot capability registers at D4h - D7h needs to be listed as reserved.

Implication: Invalid capability information provided to the host.

Workaround: Do not use Chassis/Slot PCI Extended Capability.

Status: Fixed. Fixed in B-0 stepping. The fix is to change the capability pointer at 34h to point to the next capability pointer (Power Management) at DCh, instead of D4h. D4h is to return zero when read. See the Table , “Summary Table of Changes” on page 9.

14. SDCR0.2 implemented as 'Reserved'

Problem: The attribute for bit 2 in the MCU SDRAM Control Register 0 (SDCR0) is incorrectly implemented as 'Reserved', instead of 'Read Only'. Since it is 'reserved', software cannot rely on reading this bit to determine when DDR or DDR-II memory type is selected. The external state of MEM_TYPE can not be correctly identified by reading this bit.

Implication: This bit cannot be used to determine memory type.

Workaround: Since hardware design is specific to either DDR or DDR-II, software engineers know what memory is being used in their application and therefore need not rely on this bit. Also, software can read DIMM information via Serial Presence Detect (SPD).

Status: Fixed. Fixed in B-0 stepping. See the Table , "Summary Table of Changes" on page 9.

15. 32-bit region missing proper address decode

Problem: When connected to 64-bit SDRAM, a 32-bit region can be defined by the S32SR register. There is an issue with the calculation that is performed to determine whether a transaction is in the 32-bit window or not. The MCU address decoder is missing the MSB for the 32-bit region hit/miss. This issue only applies to systems that intend to use the 32-bit region. There are no other base address restrictions.

Implication: 32-bit addresses can be decoded improperly causing unstable code execution.

Workaround: Do not allow the base address, defined by SDBR, to be aligned on an odd 1G boundary (e.g., bit 30 cannot be set to a '1':

For example, allowable base addresses: 0x8000_0000, 0xA000_0000, 0x2000_0000

Non-allowable base addresses: 0x4000_0000, 0xC000_0000, 0x4100_0000

Status: Fixed. Fixed in B-0. See the Table , "Summary Table of Changes" on page 9.

16. S_GNT[3:2]# outputs are not pulled high when Bridge is disabled.

Problem: When the bridge is disabled (BRG_EN = 0), the S_GNT[3:2]# signals float (Z). These signals need to not float, but be pulled high (H).

Implication: No negative impact expected.

Workaround: External pull-ups are required.

Status: Fixed. Fixed in B-0 stepping. See the Table , "Summary Table of Changes" on page 9.

17. Split Transaction Commitment limit register mechanism, in the PCI-X bridge, does not operate as implied by the *PCI-X Addendum to the PCI Local Bus Specification,* Revision 1.0a

Problem: The *PCI-X Addendum to the PCI Local Bus Specification,* Revision 1.0a requires that the bridge operate in a fully buffered mode. The 80331 bridge allows software to write the Split Transaction limit register, to allow compatibility with existing software, however, it behaves as though the register is programmed with a value FFFFh. The bridge forwards all split transactions without regard to the sequence size or the amount of available buffer space.

Implication: This behavior could result in the bridge split completion buffers becoming full, while additional split completions are being received. When this situation occurs, the bridge either issues a retry or disconnects on the next ADB, until buffer space is available.

Workaround: No workaround.

Status: No Fix. Not to be fixed. See the Table , "Summary Table of Changes" on page 9.

18. Watchdog time-outs by the PCI-X bridge may cause data corruption

Problem: A watchdog time-out by the PCI-X bridge may cause data loss or corruption in an unrelated transaction.

Implication: Watchdog time-outs can cause data corruption.

Workaround: Do not enable the watchdog time-out counter (default setting). The Watchdog Timer has been de-featured from the 80331. See Specification Change #4.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

19. Discard timer expiration on delayed read can cause data corruption or deadlock when data is still being received on target bus

Problem: The discard timer in the PCI-X bridge starts counting when read data is first received. When data comes in very slowly (for example, when the target bus speed is much slower than the requesting bus) the discard timer on the requesting bus may elapse before the read data is complete, and cause state machine errors.

Implication: When a delayed read in PCI-to-PCI or PCI-to-PCIX mode receives immediate data on the destination bus, and the time required to receive that data exceeds the value of the discard timer, data corruption or deadlock may occur.

Workaround: Set discard time-out value to 2^15 (default setting) or reduce prefetch settings when bus speeds are very different.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9.

20. Configuration cycle attribute parity error signaled incorrectly by PCI-X bridge

Problem: A configuration cycle with an attribute parity error gets a retry response by the PCI-X bridge, instead of a target abort.

Implication: When the SERR# Enable bit (PCR.8) is not set, transient parity errors may go undetected.

Workaround: Set PCR.8 to enable primary bus SERR# assertions for parity error detection.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9.

21. Transactions are buffered during Secondary Reset

Problem: When the secondary bus is reset by software (i.e., setting the secondary bus reset bit - BCR.6), the buffer control needs to retry all transactions (except type 0 configuration cycles). The problem is that the buffer controller is not looking at the secondary reset signal, so when the secondary bus reset bit is set (S_RST# asserted) the buffers enqueue a transaction that was retried. Instead, it needs to remain idle until reset is complete, and not enqueue the retried transactions.

Implication: May cause data corruption.

Workaround: After de-asserting S_RST# by clearing (i.e., writing a '0') the Secondary Bus Reset bit (BCR.6), wait 155 µs before sending a new transaction to the bridge.

Status: Fixed. Fixed in C-0 stepping. See the Table , “Summary Table of Changes” on page 9.

22. Primary bus pin mode behavior incorrect during reset in the 80331 no bridge mode

Problem: In the 80331 no bridge mode (BRG_EN = 0), several unused primary PCI signals provide the wrong behavior during reset. The following signals float (‘Z’ = output disabled) during reset: P_AD[63:0], P_PAR, P_PAR64, P_C/BE[7:0]#.

Implication: No negative impact expected.

Workaround: No workaround.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9. These signals are ‘H’ (pulled up to Vcc).

23. P_REQ# pin mode behavior when in the 80331 no bridge mode

Problem: In the 80331 No Bridge Mode (BRG_EN = 0), P_REQ# (ball T6) drives a 1 in A-1, needs to be “H” (pulled up to Vcc).

Implication: No negative impact expected.

Workaround: No workaround.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9. Note: P_GNT# and P_REQ# signals have new ball locations on B-0, see Specification Change #5.

24. Master abort after data transfer within a single ADB on PCI-X to PCI-X read block transactions may cause data corruption or deadlock

Problem: When a PCI-X to PCI-X Memory Read Block request is terminated with a single data phase disconnect (SDPD) by the target after the target has provided some data, and a subsequent request in the same allowable disconnect boundary (ADB) is not claimed (Master Abort), the bridge may deallocate memory buffers not involved in the transaction.

Implication: This may cause data corruption of another transaction or result in deadlock. This erratum only occurs in PCI-X to PCI-X mode, the read byte count must be greater than 4 bytes and must not cross an ADB boundary. The target must respond with a single data phase disconnect and when the bridge attempts to get the remainder of the data, the target does not claim the transaction.

Workaround: No workaround. The likelihood of this occurrence is extremely small and would require 'bad' behavior by the target.

Status: Fixed. Fixed in C-0 stepping. See the Table , “Summary Table of Changes” on page 9.

25. Boundary scan multi-chip module implementation

Problem: The 80331 is not BSDL-compliant for the SAMPLE and BYPASS instructions specified by the JTAG specification 1149.1. It is compliant for most board-level testing. When boards are tested for opens and shorts, the 80331 BSDL can define the boundary scan length as +1 to encompass the BYPASS register in the Intel XScale® core (therefore not visible).

Implication: When doing ID, SAMPLE, or BYPASS, the Intel XScale® core BYPASS register makes the path from TDI to TDO one flop longer than the specification requires, which could cause canned software to error.

Note: When neither of these are used by the board vendors during manufacturing testing, there is no issue.

Workaround: Intel can provide two BSDL files which allow opens and shorts testing, as long as it does not test the ID and BYPASS instructions. One covers the Intel XScale® core unit and the other one for the I/O processor, with the exception that both instruction sets are reduced from 14 to 7, since they are operating independently.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

26. Auto refresh command also generates a Precharge All command on DDR bus

Problem: When an auto refresh command is issued to the MCU SDRAM Initialization Register (write 0x6 to SDIR) the hardware state machine executes a precharge all command and then an auto refresh command.

Implication: Some DIMMs may fail to initialize.

Workaround: There is no way to decouple the precharge all and auto refresh commands in the MCU. However, the DCAL can issue an auto refresh command, which can be used instead to issue the auto refresh for initialization.

For both DDRI and DDRII initialization sequences, there are two back-to-back AutoRefresh (ARF) commands issued to the DIMM (accomplished by writing 0x6 to MCU_SDIR 0xFFFF_E500 twice). This sequence is replaced by writing to the DCAL Control and Status Register (DCALSR) 0xFFFF_F500 to issue an AutoRefresh to each Chip Select, thus there are four writes to the DCALSR as opposed to the previous two writes to MCU_SDIR. The exact address and data values are as follows:

```
ADDRESS      WRITE_DATA     NOTE
0xFFFF_F500 0x8000_0001 -- ARF to CS[0]
0xFFFF_F500 0x8000_0001 -- ARF to CS[0]
0xFFFF_F500 0x8010_0001 -- ARF to CS[1]
0xFFFF_F500 0x8010_0001 -- ARF to CS[1]
```

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

27. SERR# set to incorrect voltage

Problem: SERR# is being driven to 1 V. The 80331 might detect SERR# being asserted during boot-up and does not detect SERR# assertion when a device on the PCI-X bus asserts SERR#.

Implication: The 80331 falsely detects and logs SERR# assertion during boot-up, due to the SERR# pin being held at 1 V, which the 80331 detects as a logic low (asserted). Devices assert SERR# for a minimum of one clock when an SERR# occurs. Since SERR# is being held low, devices asserting SERR# are not detected by the 80331.

Workaround: Putting a 34–39 Ω pull-up resistor on SERR# might work; however, some add-in cards do not have the drive strength to pull-down against this pull-up, and therefore cannot reliably generate SERR# in the system. An alternative workaround is to have software mask SERR# assertion in the ATU or bridge.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9.

28. M66EN set to incorrect voltage

Problem: The M66EN signals do not get set to the appropriate level with or without an add-in card present. M66EN should be at 3.3 V without an add-in card present.

Implication: Prevents auto-detection of 66 MHz PCI devices. Forces the bus speed to 33 MHz when not in PCI-X mode and only impacts busses that should be set to 66 MHz PCI.

Workaround: Use a 34-39 ohm pull-up resistor on the M66EN signals.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9.

29. P_INT[D:A]# not operating correctly

Problem: The P_INT[D:A]# signals only rise to ~1 V at power-on.

Implication: P_INT[D:A]# interrupt signals can cause false interrupts to the host.

Workaround: Use a 34-39 ohm pull-up resistor on the P_INT[D:A]# signals.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9.

30. Secondary bus may not initialize correctly at 100 MHz PCI-X or 133 MHz PCI-X

Problem: When a 133 MHz PCI-X capable device is plugged into the secondary PCI-X bus supporting >= 100 MHz PCI-X, the 80331 might not correctly initialize the bus at 100 MHz PCI-X or 133 MHz PCI-X.

Implication: The bus may initialize at 66 MHz PCI-X.

Workaround: Replace the 3.3 Kohm pull-up resistor on S_PCIXCAP with a 1.5 Kohm resistor.

Status: Fixed. Fixed in B-0 stepping. See the Table , “Summary Table of Changes” on page 9.

31. Coalesced writes to 32-bit memory can cause data corruption

Problem: The Bus Interface Unit (BIU) can cause data corruption within the memory controller unit when either an 8-byte write with a starting address offset of four (i.e., DWord alignment) or a 12-byte write is done to 32-bit memory (or the 32-bit memory region).

Implication: Can cause data corruption. This is not an issue with 64-bit memory subsystems.

Workaround: When the use of 32-bit memory or the 32-bit memory region is required, write coalescing must be turned off.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

32. ATU passing rules operation in PCI mode

Problem: When the secondary PCI bus is in PCI bus mode, the PCI passing rule enforcement logic within the ATU, allows a read completion to pass write data; until at least four inbound delayed reads, inbound configuration writes, inbound configuration reads, or any combination of these have occurred from the PCI bus.

Implication: This issue causes a deadlock condition in legacy devices, which contain shared read and write data queues, where the device allocates the data buffer for the requested delayed read data and is also being addressed by the outbound ATU write data. This ATU functionality does not exist in PCI-X mode.

Workaround: Use the secondary PCI bus in PCI-X mode.

When the secondary PCI bus is in PCI mode, configuration retry is enabled, and the legacy buffer allocating device is present in the system, the ATU must not issue writes to that device until the configuration cycles or reads have completed. When this situation cannot be achieved by the host, the ATU can be momentarily programmed in loopback mode and issue delayed reads to itself. This workaround is only required when the ATU is sending upstream traffic at the same time as the host is configuring it. This should not happen since the ATU needs to wait for the driver to be enabled after enumeration completes.

This erratum does not occur when configuration retry is deasserted at power on and the host meets the above configuration cycles.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

33. S_INT[D:A]# pull-ups disabled by Internal Bus Reset

Problem: During power-on or anytime the internal bus is reset, the S_INT[D:A]# interrupt inputs are asserted as the on-die termination (ODT) gets disabled.

Implication: This situation causes floating/asserted interrupts to the 80331, during power-on or anytime the internal bus is reset.

Workaround: Need external pull-ups on the S_INT[D:A]# pins.

Status: Fixed. Fixed in C-0 stepping. See the Table , “Summary Table of Changes” on page 9.

34. MCU Preemption Control does not properly manage the byte count

Problem: During a transfer, the MCU tracks the byte count to determine when a transfer can be preempted. When preemption is enabled, a preempted transfer that is resumed does not transfer the full remaining byte count of data.

Implication: Can cause data corruption when preemption control is enabled.

Workaround: Do not enable preemption control. Keep the default setting of the MCU Preemption Control Register (MPCR, FFFF E540h) bits 3-0 as 0h.

Status: Fixed. Fixed in C-0 stepping. See the Table , “Summary Table of Changes” on page 9.

35. I²C unit hang condition

Problem: When a processor reset occurs, the 80331 does not properly detect an idle condition on the I²C bus, potentially causing the I²C unit to hang indefinitely. This occurs only when an I²C slave device is writing a 0 on the SDA signal when reset is asserted. The SCL signal goes high, but SDA remains low, signaling that the bus is still busy. Warm reset does not clear up this condition; only a cold reset (power-on) resolves the I²C bus hang.

Implication: The I²C bus stays in a hang condition indefinitely

Workaround: Since this is fixed in the C-0 stepping, this workaround must be implemented only when I²C hang issues are preventing proper usage of A-2 and B-0 parts. The recommended workaround for A-1 and B-0 parts involves both a hardware and software change. In hardware, tie GPIO pins to the SCL signals; then in the initialization software, the GPIO pins must toggle the SCL signals to bring the slave I²C devices out of the locked condition. This can be done by using one GPIO signal and an external tristate buffer on SCL lines, or by using two GPIO signals and no external logic to allow unlocking the I²C bus. Also, the external PWRDELAY circuit must use S_RST#, instead of PWRGD.

Status: Fixed. Fixed in C-0 stepping. See the Table , “Summary Table of Changes” on page 9.

In the C-0 stepping, the I²C bus lock condition can be cleared by software doing a toggle of the GPOD[11:10] to toggle SCL[1:0]. Bit[11] and bit[10] of the GPOD register are writable. When GPOD[11] is high, SCL[1] is driven low. When GPOD[10] is driven high, SCL[0] is driven low. I²C is no longer in the processor reset equation for the internal bus (that it, MRST), so the I²C bus can hang when a reset happens during the reading of a 0 from an external I²C slave. Therefore, the software is required to toggle the I²C clock signals (at least eight times) before I²C is enabled. GPOD bit[11] and bit[10] can be used to unhang the bus by creating a 100 K (5 ms high and low time) or 400 K (1.25 ms high and low time) clock pulse.

Also, since the I²C bus has been removed from the processor reset equation, the PWRDELAY circuit is no longer needed. Only a 1.5 KΩ pull-up to 3.3 V is required on PWRDELAY. See Specification Change 11, “PWRDELAY needs only a pull-up for battery back-up mode” on page 48.

36. VPD Data Register bit 19 is not read/write

Problem: VPD (Vital Product Data) is an extended capability of the ATU. Bit 19 of the VPD Data Register (FFFF_E1BCh) was implemented as RC (read clear) instead of RW (read/write).

Implication: This is application dependant as VPD provides the system with information that uniquely identifies hardware and, potentially, software elements of a system.

Workaround: When the VPD feature is needed, care must be taken to mask bit 19.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

37. Intel XScale® Core lockup condition

Problem: The Intel XScale® core fetches code and data through an internal switch called the Bus Interface Unit (BIU), which manages traffic in two directions; a general path to the Flash and PCI-X interfaces (as well as internal units) and a private path to DDR memory controller. There is a boundary condition for returning data, specifically for code fetches from Flash or host memory colliding with private DDR read data. At the wrong clock alignment, the BIU corrupts the returning code with the DDR data. This results in arbitrary code returned to the core, resulting in lockup or other unpredictable core behavior. The following describes one scenario of how this condition can occur:

- The Intel XScale® core fetches an instruction cache line, which is executed through the internal bus to the Flash interface unit and out to Flash memory.
- While this is happening, the Intel XScale® core is scrubbing the DDR clean with a series of data transactions through a second private path, directly to the DDR Memory Controller. These data transactions are a series of cache line writes to DDR and reads from DDR.
- The original instruction fetch is returned from the Flash interface unit back to the core with the critical instruction first (critical word first).
- One clock after the instruction cache line is being returned to the Intel XScale® core, one of the data transactions returns from the DDR memory controller.
- The BIU switches to handle the higher priority DDR return.
- The instruction cache line that was being returned from the Flash interface is only partially returned to the Intel XScale® core (five of the eight instructions).
- The data cache line from the DDR memory controller is then returned to the Intel XScale® core in its entirety.
- The Intel XScale® core then starts executing the instructions that were fetched in the cache line (since the critical instruction was returned first, the core can continue).
- The five instructions that were returned are executed by the Intel XScale® core and the core then stalls waiting for the sixth, seventh and eighth instruction to be returned from the initial cache line fill.
- The BIU continues to return any outstanding transactions unaware that the instruction cache fetch was not fully returned.

Note: 1) The case where this happens is where the core memory port is enabled and there is a cache line fill (instruction or data) that executes through the internal bus (Flash/PCI/etc., transactions, not DDR transactions) and any size fetch to DDR through the direct port to DDR memory controller. The timing has to be such that the cache line fill from the internal bus must beat the DDR return by 1 clock cycle.
2) The scenario given above is just one example of what excites this condition and is by no means the only scenario that excites this condition.

Implication: This results in a lockup or other unpredictable core behavior.

Workaround: For A-x and B-0 silicon the only reliable workaround is for software to turn off the core to the DDR Memory Controller port, redirecting DDR accesses through the Internal Bus interface with other transactions. This prevents the core lockup condition but at a cost to performance.

Status: Fixed. Fixed in C-0 stepping. See the Table , “Summary Table of Changes” on page 9.

38. 32-bit region write corrupts ECC immediately after 64-bit Read-Modify-Write

Problem: ECC can be corrupted when the following scenario occurs:

- All of memory is initialized to 0s
- A 32-bit region is enabled.
- Agent A fills the 64-bit memory region with pattern A.
- Agent A does a DDR write to same 64-bit memory region, causing an Read-Modify-Write (RMW) at the end of the transfer where pattern A is returned as part of the RMW process.
- Immediately thereafter, Agent B does a DDR write to an open region of 32-bit memory. On DQ[63:32] data is driven to pattern A. The wrong ECC is generated (where the ECC is that of DQ[63:0] where pattern A is present at the upper half of the bus).

Implication: The wrong ECC is generated (where the ECC is that of DQ[63:0] where pattern A is present at the upper half of the bus).

Workaround: Turn ECC off or don't use the 32-bit memory region.

Status: Fixed. Fixed in C-0 stepping. See the Table , "Summary Table of Changes" on page 9.

39. Reset straps incorrectly sampled on the secondary reset

Problem: All reset straps are sampled on the rising edge of P_RST#, except the following: MEMTYPE, P_BOOT16# and CORE_RST#.

Implication: When external circuitry is used to dynamically control the reset straps, the timing may cause incorrect values to be latched.

Workaround: By default all reset straps have internal pull-ups. When non-default state is required, then an external 1.5K pull-down resistor should be used. This implementation has no timing requirements. Otherwise, when external circuitry is used to dynamically control, then make sure the proper level is provided at the rising edge of S_RST#.

Status: Fixed. Fixed in B-0 stepping. See the Table , "Summary Table of Changes" on page 9.

40. PCI-X to PCI Memory Read double-word near 1MB boundary may cause the system to hang

Problem: PCI-X to PCI Memory Read DWORD near a 1 MB boundary may cause the system to hang.

Implication: In the following scenario:

- A master on the PCI-X bus initiates a Memory Read DWORD through the bridge, with a starting address that is not DW aligned and is within the last 4 bytes of a 1 MB address boundary.
- On the PCI side, the bridge initiates the request and disconnects after a single data phase. The transaction should end at this point, but instead, the transaction is then issued a second time with the starting address at the 1 MB boundary before being retired.
- Back on the PCI-X side, the completion is issued with the correct data for the initial 4-byte request, followed by a second “unexpected” completion. As a result of the second completion, the internal resource counter gets decremented twice, causing the resource checking logic to assume there is no more room on the chip for split transactions.
- At this point the bridge can no longer process upstream non-posted requests and terminates them with retry.

Note: The above failure only occurs on a Memory Read DWORD from PCI-X to PCI when the starting address is not DW aligned. A Memory Read DWORD to 0xffffc does not fail, whereas a read to 0xfffff does fail.

Workaround: This issue only affects PCI-X to PCI transactions. PCI to PCI-X, PCI to PCI and PCI-X to PCI-X transactions are not affected. PCI-X to PCI implementations should not attempt to do a Memory Read DWORD that is not DW aligned and is within the last 4 bytes of a 1 MB address boundary.

Status: Fixed. Fixed in C-0 stepping. See the Table , “Summary Table of Changes” on page 9.

41. PCI-X to PCI Memory Read Block across 1MB boundary may cause data corruption

Problem: PCI-X to PCI Memory Read Block across 1MB boundary may cause data corruption.

Implication: In the following scenario:

- A master on the PCI-X bus initiates a Memory Read Block through the bridge, with a starting address + byte count that crosses a 1 MB boundary. The request is not DWORD aligned and starts less than three data phases from the 1 MB boundary.
- On the PCI side, the bridge initiates the request by setting the low two address bits to 00b as required by the *PCI Local Bus Specification*, Revision 2.3.
- The bridge receives data up to the boundary and disconnects, then requests the remainder of the data in a second transaction.
- On the source bus, the bridge should deliver a small completion up to the 1MB boundary with the BCM bit set. However, in this case the bridge plays the request without BCM and uses the full remaining byte count. Since the bridge can not disconnect, it winds up satisfying the byte count with garbage for data after the boundary.

Note: For failure to occur, starting address and byte count must be set so the request crosses the 1 MB boundary, but bytes requested after the boundary are less than 4 bytes. Requests that are DWORD aligned do not fail, nor do requests that do not cross the 1MB boundary (even when not aligned).

Workaround: This issue only affects PCI-X to PCI transactions. PCI to PCI-X, PCI to PCI and PCI-X to PCI-X transactions are not affected. PCI-X to PCI implementations that do not cross a 1 MB boundary does not experience this issue.

Status: Fixed. Fixed in C-0 stepping. See the Table , “Summary Table of Changes” on page 9.

42. DMA CRC result is byte reversed

Problem: The DMA CRC result value is byte reversed.
Example CRC operation with the DMA:
CRC Seed: 0x0000 0000
Data Pattern (16 bytes): 0x1234 5678, 0x1457 9098, 0x1234 5678 0x1457 9098
Expected CRC Value: 0x6791 25DA
Actual CRC Value: 0xDA25 9167

CRC Seed: 0x1122 3344
Data Pattern(16 bytes): 0x1234 5678 0x1457 9098, 0x1234 5678 0x1457 9098
Expected CRC Value: 0x1D2C B941
Actual CRC Value: 0x41B9 2C1D

Implication: Data corruption occurs when software does not take care of the byte reversal.

Workaround: Software must byte reverse the CRC result after the DMA completes.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

43. CRC corruption on PCI-to-local DMA transfers

Problem: CRC corruption can occur when polling DMA registers during PCI-to-local transfers. CRC corruption happens regardless of whether the transfer data is written to memory (DMA Transfer disable bit DCR.7, is set). Both DMA channels are affected.

Implication: CRC data corruption occurs. The DMA transfer itself is not affected by this erratum.

Workaround: Use local address sources only when calculating CRC.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

44. Byte Count Modified bit set to 1

Problem: During a memory read DWORD which crosses the 1 MB boundary in PCI-X to PCI mode, the split completion has the correct data and byte count but the BCM (Byte Count Modified) bit in the attribute field is set to 1. In the *PCI-X Addendum to the PCI Local Bus Specification*, Revision 1.0a description of the BCM bit it states, “BCM is used only for split completions resulting from burst transaction and is set to 0 for split completions resulting from all other commands.”

Implication: The *PCI-X Addendum to the PCI Local Bus Specification*, Revision 1.0a states that for non-burst transactions that the BCM bit is not to be used and should read 0, this bit is used for diagnostic purposes, and targets (bridges and requesters) are permitted to ignore this bit.

Workaround: BCM is a non-error status bit, only used for “flow control” in split completions and should be ignored.

Status: Fixed. Fixed in D-0 stepping. See the Table , “Summary Table of Changes” on page 9.

45. Corrupted byte count and data when crossing 1 MByte boundary in PCI-X to PCI mode

Problem: Scenario - Memory Read Block PCI-X to PCI mode beginning at address 9, 10 or 11 bytes from the 1 MB boundary, with a byte count that crosses the boundary.

In this failing scenario, the check for two DWords away from the boundary relies on the aligned DWord address of the third DWord. Since the starting address is in the middle of the third DWord, the case is missed. The result is the PCI read satisfies the byte count by reading across the 1 MB boundary.

The first split completion on the PCI-X bus is correct data, modified byte count and BCM set. The second split completion on the PCI-X bus has a corrupted byte count, therefore, the correct data is sent and additional garbage data is then sent until the corrupt byte count is satisfied.

The corrupted byte count is a result of crossing the 1 MB boundary. The original byte count minus the byte count captured is less than zero.

The failing case is very similar to the previous memory read block case, but the impact to the requester is different.

Implication: Corrupted byte count and additional garbage data is sent to the requester to satisfy the corrupted byte count. In some cases, other transactions (on the bridge) following the bad split completion can be corrupted. When the bridge completes sending the corrupted byte count, it frees the buffer space used by that data. That buffer may contain data for other transactions and that data is lost.

Workaround: Do not allow PCI-X to PCI memory reads with DWORD unaligned starting address to cross the 1 MByte boundary.

The starting address must have bits [1:0] equal to 01, 10 or 11 for this erratum to occur. When the starting address[1:0] = 00, this erratum does not occur.

Status: Fixed. Fixed in D-0 stepping. See the Table , “Summary Table of Changes” on page 9.

46. PCI-X to PCI Memory Read with 4 K byte count and unaligned starting address

Problem: The initiator on the PCI-X side issues a Memory Read Block with 4 K request byte count (BC=000h), which the bridge terminates with Split Response. The address in this case is not DWORD aligned (A[1:0] does not = 00b). On the PCI side, the bridge executes the transaction as Memory Read Multiple, but disconnects after only a single dataphase and does not re-issue the transaction. No split completion is returned to the PCI-X bus and the transaction permanently consumes transaction resources.

Implication: This can permanently consume transaction resources, and therefore could result in a system hang.

Workaround: Use addresses that are DWORD aligned or any byte count other than 4 K.

Status: Fixed. Fixed in D-0 stepping. See the Table , “Summary Table of Changes” on page 9

47. VCCDDR (VCC25/VCC18) Current Spike

Problem: An internal DDR signal gets routed through logic that is powered by the VCC15 rail. This signal gets driven to the wrong level when VCC15 rail is not powered up. This signal controls the input and output enable of all DDR buffers, so when VCC15 is off and VCC25/18 is on, it asserts high and cannot tristate the DDR outputs. The issue can occur at power-up, power down and power fail when battery back-up continues to power the 80331.

Implication: Excessive VCC25/18 power supply rail current to the VSS rail is experienced until VCC15 core supply is up.

Also, when using DDR-I mode (VCC25), the buffers can be overstressed when VCC15 = off and VCC25 = on. This can cause a reliability problem since the processor may become damaged over time.

When in DDR-II mode (VCC18), the buffers do not overstress since VCC18 maximum is 1.9 V. Therefore, no reliability problems exist.

Workaround: A new power sequencing requirement should be followed for B-0 and C-0 steppings. This problem does not affect A-1 stepping, and is being fixed in C-1 stepping.

1. VCC33 power up
2. VCC15 power up
3. VCC25/18 power up

The VCC15 minimum supply voltage has to be within 25% of the VCC25 supply voltage, until the VCC15 supply reaches 1.425 V (VCC15 minimum), during power-up and power-down sequences.

This is the minimum requirement to ensure no gate overstress when using DDR-I memory.

For power-fail condition, the battery should power only the DIMM. The 80331 should be isolated from the battery power.

Status: Fixed. Fixed in C-1 stepping. See the Table , “Summary Table of Changes” on page 9.

48. Bridge PCI ordering rule violation

Problem: When the bridge is in PCI-to-PCI mode, it receives a downstream memory write which is claimed by the bridge, followed by a subsequent memory read request which is enqueued immediately afterwards. At the same time, the upstream buffers on the chip are filled with upstream Memory Writes and read completion data. On the destination bus, the memory write is issued in small fragments due to either target disconnect or a low setting of the secondary MLT, and the upstream buffers begin to free up due to delivery of upstream Memory Write data. In between delivery of downstream Memory Write fragments, the bridge issues the subsequent Memory Read, in violation of PCI ordering rules.

Summary of the four failure modes:

1. PCI-to-PCI: A Memory Read transaction may pass a previously enqueued Memory Write.
2. PCI-to-PCI: A Configuration or I/O Read transaction may pass a previously enqueued Memory Write.
3. PCI-to-PCI: A Configuration or I/O Write transaction (including type1 configure-to-special cycle) may pass a previously enqueued Memory Write.
4. PCI-X-to-PCI: A Configuration or I/O Write transaction (including type1 configure-to-special cycle) may pass a previously enqueued Memory Write.

Implication: In the above scenario, a PCI read/write transaction can pass a PCI memory write transaction which violates PCI ordering rules.

When the primary bus is PCI and secondary bus is PCI-X, this would only be seen as described in item 4 listed above. No impact when the bridge is configured in PCI-X to PCI-X mode.

Workaround: Use in PCI-X to PCI-X mode. When the primary bus is PCI and secondary bus is PCI-X, do not use upstream DWORD I/O or type1 configure-to-special cycle writes that must be ordered against previously issued Memory Writes.

Status: Fixed. Fixed in D-0 stepping. See the Table , “Summary Table of Changes” on page 9.

49. ATU claims PCI commands 8 and 9 when issued as Dual Address Cycle (DAC)

Problem: In PCI mode, commands 8 and 9 are reserved. The appropriate PCI response to these commands is to master abort. When these commands are issued as a Dual Address Cycle (DAC), the Address Translation Unit (ATU) claims them and they are executed as Memory Read (command 8) and Memory Write (command 9) on the internal bus. The ATU properly master aborts SAC PCI commands 8 and 9.

This issue does not occur in PCI-X mode.

Implication: No negative impact expected, since these PCI commands are ‘reserved’ and should not be issued to the ATU.

Workaround: No workaround.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

50. Secondary bus PCI RST# pulse prior to the rising edge of P_RST#

Problem: During system power on and prior to 80331 receiving the rising edge of P_RST#, a pulse may be observed on the secondary bus PCI RST# signal (S_RST#).

This functionality is a result of the 3.3v to 1.5v leakage described in Specification Clarification 19. Other signals that may see a pulse during power-on include the following: all Peripheral Bus Interface (PBI), PCI, GPIO, UART, JTAG and PWRDELAY signals. Refer to Specification Clarification 23 for specifics on PWRDELAY. Signals not included are DDR and I2C.

Implication: PCI/PCI-X controllers on the secondary bus segment could interpret this S_RST# pulse as a true rising edge and initialize into an undetermined state. Pulses on PWE# and PCEx# may cause data corruption for memory devices connected to the PBI bus.

Workaround: A hardware workaround has been identified. The P_RST# signal that is received by the 80331 can be used to gate the secondary bus PCI RST# signal. For example, use P_RST# and S_RST# as inputs to an AND gate and connect the output to the secondary device RST# pin. The gate delay should be kept down to a couple nanoseconds, so as to not interfere with the PCI initialization pattern.

Another workaround is to bring up 3.3 V and 1.5 V power rails simultaneously, but continue to maintain the power sequencing requirement (as specified in the design guide) for these two power rails.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

51. Slow edge rates observed when 80331 is driving the primary bus

Problem: Signal integrity issues might occur when the 80331 is driving the primary PCI/PCI-X bus in conventional PCI or PCI-X modes.

Implication: Parity errors and system hangs may occur.

Workaround: A software workaround has been identified. The workaround is required for reliable conventional PCI and PCI-X mode operation. The workaround is to write all 1s to the PCI drive strength overrides with the Intel XScale® core.

Bit 31 should be set in the following two registers to enable drive strength override:

- 3.3 V Drive Strength Control Register at FFFF_F5D0h
- 1.5 V Drive Strength Control Register at FFFF_F5D4h

  The default value is 0000_3F3Fh for both registers.

Status: Fixed. Fixed in C-0 stepping. See the Table , “Summary Table of Changes” on page 9.

52. Enabling the core-to-memory port can cause a stall condition

Problem: A stall condition can occur during high data throughput to the Memory Controller Unit (MCU), due to a non-empty Bus Interface Unit (BIU) data queue that might allow no further traffic to be submitted from the Intel XScale® core. The stall is due to an invalid state in the queue control registers and the associated pointers. The pointers and the queue status bits get out of sync, allowing part of the BIU to conclude that it is full and therefore not to accept any more transactions. The other part of the BIU concludes that it is empty, and therefore does not flush the non-empty condition.

The stall condition can be generated in the following scenario:

1. Within the MCU, accept a read from one port (core-to-memory or IB-MCU). This triggers an automatic coherency check, which requires writes within the same 1 K address "page" to be pushed ahead of the read.
2. An "incoherent" write (in other words, a write that must precede the read from step 1) must exist in the opposite port. This write is executed to DDR memory as normal.
3. A read must also exist in the opposite port; destination (coherent or incoherent) is unimportant. This has the effect of maintaining the port's request to the MCU.
4. An incoherent write arrives in the original port during the exact cycle the opposite write is completed.
5. Due to the combination of pending read request and last-second incoherent write, the MCU gets out of sync and attempts to acknowledge a write from the wrong port. This causes the offending queue pointer/status bit mismatch. This situation locks the opposite port to the original read transaction.

Implication: Enabling the core-to-memory port (BIUCR.0 = 1, IB address = FFFF_E608h) can cause a stall condition or data corruption.

Workaround: Do not enable the core-to-memory port in the BIU Control Register (BIUCR.0). When BIUCR.0 = 0 (default condition), the core-to-memory port is disabled and forces all Intel XScale® core memory transactions to be issued out the core internal bus (IB) port, therefore avoiding the stall condition.

Note: Disabling the core-to-memory port requires a review of specification clarification 13, "Bus Interface Unit follows PCI ordering rules" on page 54.

Status: Fixed. Fixed in D-0 stepping. See the Table , "Summary Table of Changes" on page 9.

53. PCI-to-PCI read flow-through with destination TRDY# stalls can cause data corruption

Problem: PCI-to-PCI read flow-through with destination TRDY# stalls can cause data corruption.

Implication: The scenario is a PCI-to-PCI memory read with flow-through enabled (source bus bandwidth is less than or equal to destination bus bandwidth). Requestor initiates the read on the source bus, and it is retried by the bridge (executed as a delayed transaction). The bridge plays the transaction on the destination bus and starts receiving data from the target. After getting two or three cache lines, the transaction is requested again on the source bus, and the bridge enters flow-through. The target device on the destination bus begins to insert target wait-states, and eventually ends the transaction by stalling TRDY# for a number of clocks then signaling disconnect without data at the cache line boundary. On the source bus, the bridge delivers an extra garbage dataphase beyond what was received on the destination side, causing data corruption.

The affected mode is PCI-to-PCI only. Neither PCI-X-to-PCI-X nor PCI-to-PCI-X modes allow TRDY# stalls on the destination side.

Workaround: Do not use in PCI-to-PCI configuration. Expected usage model for the 80331 is that the secondary PCI bus operates in PCI-X mode only.

For an alternate workaround, when in a system that has devices that might terminate transactions in the above manner, the prefetch policy registers must be set so that the target is not given the opportunity to significantly stall TRDY#.

For example: When a device can provide 256 bytes in a timely manner, but delays a request for 384 bytes, the first read and reread factors must not be set higher than 01h (giving an MRM prefetch of 256 bytes, assuming a 64-byte cache-line size).

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

54. S_PCIXCAP PCI mode threshold is too high

Problem: The “PCI mode” threshold on the S_PCIXCAP signal on the secondary PCI bus is too high (1.87 V). This has the effect of detecting a PCI bus when more than two slots are populated with PCI-X 66 cards.

Implication: A heavily loaded secondary PCI bus (three or four slots), might operate in PCI mode instead of PCI-X mode.

Workaround: When a board is configured with more than two PCI-X slots, a circuit can be added to adjust the PCIXCAP voltage.

Status: Fixed. Fixed in D-0 stepping. See the Table , “Summary Table of Changes” on page 9.

55. No support for burst I/O and configuration read/writes

Problem: The 80331 bridge does not support burst I/O read, I/O write, configuration read, or configuration write in PCI mode.

Implication: In conventional PCI mode, it is legal for a requester to attempt a burst I/O read, I/O write, configuration read, or configuration write. The 80331 bridge supports only single data phase configuration and I/O transactions.

When a master attempts to issue a burst I/O or configuration read, the bridge does not assert STOP# after the first data phase, resulting in data corruption of all DWords beyond the first.

For burst I/O or configuration writes, the bridge asserts STOP#, but does not de-assert TRDY# on the final data phase, resulting in loss of data (the second data phase is dropped by the bridge).

PCI-X does not permit these modes of operation.

Workaround: X86 processors do not allow this condition to occur. In non-X86 systems that do not preclude these burst mode operations, the application must limit the transfer to a single aligned DWORD (32 bit) operation.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

56. Read flow through hangs due to disconnect without data at a buffer boundary.

Problem: Read flow through hangs due to disconnect without data at a buffer boundary.

Implication: In PCI to PCI mode an upstream read is issued to the bridge and forwarded to the primary bus, using a calculated prefetch size of at least 3 cachelines. On the primary bus, the target claims the request, and delivers data (with intermittent TRDY# slips) up to a 128-byte boundary, then deasserts TRDY# and waits a number of clocks before disconnecting without data. After the 2nd chatelaine gets on chip, the bridge enters flow-through and begins delivering data to the initiator. If the time difference between the end of the primary bus read and the end of the secondary bus read is approximately 10 clocks or less, the bridge may hang after the data is delivered.

Workaround: PCI-X to PCI-X and PCI to PCI-X implementations are not affected by this issue.
For PCI to PCI systems in which targets might exhibit the above behavior, set all prefetch factors to 000b. This will set the maximum prefetch size to 2 cachelines (for MRM) and effectively disable read flow-through. There may be a performance impact due to these settings.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

57. P_SERR# not asserted for parity error on AD[63:32]

Problem: Primary side P_SERR# is not asserted for a parity error on AD[63:32] during the data phase of a split response to a read request issued by the bridge.

Implication: Since no valid data is driven onto the upper A/D bus during a split response for a read transaction, there are no negative side affects of not asserting P_SERR# in this case.
This behavior does not comply with the PCI-X specification v1.0a requirements in section 5.4.1.3, however it will have no impact in the application.

Workaround: No workaround needed.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

58. Bus Interface Unit (BIU) claims DAC addresses in the range of the Memory Mapped Registers (MMR

Problem: The BIU incorrectly decodes and claims Dual Address Cycle (DAC) addresses in the xxxx_xxxx_FFFF_E000h to xxxx_xxxx_FFFF_FFFFh range (e.g. - ‘x’ represents any bit being set

to ‘1’). The 32-bit address range of FFFF_E000h to FFFF_FFFFh on the internal bus, represents MMR and reserved space. When a 64-bit address in this range is presented on the internal bus, multiple internal bus units, one of them being the BIU, will claim the transaction.

Note: Only the DMA can generate a 64-bit address (DAC) on the internal bus.

Implication: Using DAC addresses in the xxxx_xxxx_FFFF_E000h to xxxx_xxxx_FFFF_FFFFh range on the internal bus will cause an internal bus conflict that may result in the reception of undesired data and setting of error flags.

Workaround: Avoid using the DMA with DAC addresses in the xxxx_xxxx_FFFF_E000h to xxxx_xxxx_FFFF_FFFFh range. This can be done by utilizing one of the two 64MB ATU outbound memory windows (8000 0000H or 8400 0000H) and its corresponding outbound translation registers (OMWTVR0/OUMWTR0 or OMWTVR1/OUMWTR1) in order to present a 32 bit address on the internal bus and generate a 64 bit address on the PCI bus.

The upper translate value register should be programmed with the upper 32 bits of the desired PCI address. The lower translate value register would then be configured to OR in the appropriate value such that the desired lower 32 bits appear on the PCI bus after translation. Refer to Section 3.2.2 “Outbound Transactions – Single Address Cycle (SAC) Internal Bus Transactions” in The *Intel® 80331 I/O Processor Developer’s Manual* (274065) for more information on how the windowing and translation scheme works.

It is possible to now generate the 32 bit internal bus transaction either using the core or the DMA. Both options are viable and one may be preferable over the other depending on the application.

Note: Use of the DMA to generate the transaction would require not only the modification of the descriptors in question and setting of the memory-memory transfer enable bit (DCRx) for those descriptors, but care must also be taken not to allow any individual transfer to overrun the size of the outbound window (64MB).

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

59. PCIX-to-PCI Memory Read issued as 32-bit, then retried as 64-bit

Problem: A 32-bit memory read transaction from PCI-X to PCI may be retried with REQ64# asserted in violation of the PCI specification.

Implication: A memory read with multiple dataphases requested is issued from the PCI-X side across the bridge and played on the PCI-side of the bridge. This read gets some data and is then disconnected by the target. At the same time, a memory read with starting address less than 4DW from the next ADB is enqueued on the PCI-X side and then issued on the PCI immediately after the first read is disconnected.

This read is issued the first time with REQ64# deasserted and is retried by the target (delayed transaction). Some time later, the bridge re-issues this read with REQ64# asserted, violating PCI requirements that REQ64# remain the same for every retry.

Note: This can only occur in PCI-X to PCI transactions. This behavior is fairly common in many PCI devices and generally has minimal side effects on overall system performance. In some systems, depending on the setting of the discard timer, a delay may occur before normal system throughput is restored.

Workaround: Limit the PCI-side to 32-bit only in X-to-I configurations.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

60. Tc1(min) of the PCI-X clock observed to be marginally less than the requirement specified for the PCI-X (Mode 1, class1) clock jitter.

Problem: The 80331 generates the PCI-X clock with a nominal frequency of 133 MHz (Tc1 of 7.5 ns). After considering the clock jitter, the minimum clock period (Tc1(min)) observed at the pin may be less than 7.5 ns. The PCI-X class 1 clock jitter specification for mode 1 requires the Tcyc-min to be 7.5 ns with jitter consideration. The same applies for the PCI-X clock generated @ 66 MHz.

Implication: No negative impact is expected, when compliant with the routing guidelines.

Workaround: There is no workaround to adjust the minimum clock period (Tc1(min)) of the PCI-X clocks. However, the routing guidelines for the PCI-X clock signal take into consideration the effect of the jitter on the minimum clock period (Tc1(min)). Conforming to the routing guidelines in the 80331 design guide will offset the effect of the marginally reduced minimum clock period (Tc1(min)) towards the setup and hold times. Therefore, for system boards that are compliant with the routing guidelines, the risk of violating the setup and hold time requirements and any resulting functional impact, is low. Please refer to the 80331 I/O processor design guide (273823) for further information on the PCI-X clock routing guidelines.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

61. I2C Control Register reset bit does not function

Problem: The I2C Control Register (ICR0 and ICR1) bit 14 is supposed to be used for resetting the I2C unit, but writing a '1' to this bit does not reset the I2C unit. Writing a '1' to bit 14 has no effect.

Implication: The I2C unit cannot be reset by using ICRx.14.

Workaround: Depends on what needs to be accomplished. Asserting P_RST# or setting BCR.6 will reset the I2C unit but will also reset the entire chip or the secondary bus/ATU. For an I2C bus lock condition, it may be cleared by software doing a toggle of the GPOD[11:10] to toggle SCL[1:0] (see non-core erratum 35). If SDA[1:0] need to be toggled, then an external device or unused GPIO will need to be used to control this sequence.

Status: No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

62. Internal Clock Misalignment Can Cause Processor Hang

Problem: After a reset, the 80331 can hang during initial accesses to SDRAM, due to a possible clock misalignment, aggravated by a race condition in the clock divider clear circuit.

Implication: A failure will manifest itself as a hang of the I/O processor after reset, during initial accesses to SDRAM. Subsequent warm or cold resets may clear the condition and allow the 80331 to continue operation.

Workaround: In most cases, doing a cold or warm reset will clear this condition. Increasing the 1.5v power supply will reduce the probability of a processor hang. Intel is screening parts to eliminate the probability of occurrence (refer to Specification Change #22 “Internal Clock Misalignment” on page 50).

Status: Fixed. This issue was fixed in the D-1 stepping of the product (this is also related to Specification Change #22 “Internal Clock Misalignment” on page 50).

63. Spurious DMA0 End-Of-Transfer Interrupt

Problem: When the interrupt controller goes from having no interrupts asserted to one or more asserted, there is a 1-clock cycle window in which the IINTVEC (IRQ Interrupt Vector register; FFFF_E7C8h or CP6, register 14) or FINTVEC (FIQ Interrupt Vector register; FFFF_E7CCh or CP6, register 15) may report the value of the INTBASE register (Interrupt Base register; FFFF_E7C0h or CP6, register 12), which is the vector address for interrupt 0, DMA0 End-of-Transfer.
This condition can occur even if the DMA0 EOT interrupt is masked, INTCTL0.0 = 0.

Implication:	No negative impact expected. If the routine that reads the IINTVEC/FINTVEC qualifies the return value against IINTSRC0.0/FINTSRC0.0, it will either see there is nothing to do or it will validly call the DMA0 End-of-Transfer handler.
Workaround:	If IINTVEC/FINTVEC equals INTBASE, then re-read the IINTVEC/FINTVEC register.
Status:	No Fix. Not to be fixed. See the Table , “Summary Table of Changes” on page 9.

Core Errata

1. Abort is missed when lock command is outstanding

Problem: A bus abort occurs on a code fetch, while an instruction TLB or I-Cache lock MCR, *Move to Coprocessor from ARM Register*, command is outstanding. The core fails to abort, instead executing the instruction returned on the aborting transaction. Parity errors are not affected. The bus abort may be due to either an abort pin assertion or a multi-bit ECC error.

Workaround: Branch flush after every I-TLB or I-Cache lock. For example, the following instruction does this: SUB PC, PC #4;flush the pipe.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

2. Aborted Store that Hits the Data Cache May Mark Write-Back Data as Dirty

Problem: When there is an aborted store that hits clean data in the data cache (data in an aligned four word range that has not been modified from the core since it was last loaded in from memory or cleaned), the data in the array is not modified (the store is blocked), but the dirty bit is set. When the line is then aged out of the data cache or explicitly cleaned, the data in that four word range is evicted to external memory, even though it has never been changed. In normal operation this is nothing more than an extra store on the bus that writes the same data to memory that is already there.

Here is the boundary condition where this might be visible:

1. A cache line is loaded into the cache at address A.
2. Another master externally modifies address A.
3. A core store instruction attempts to modify A, hits the cache, aborts because of MMU permissions, and is backed out of the cache. That line normally is not marked dirty, but because of this errata is marked as dirty.
4. The cache line at A then ages out or is explicitly cleaned. The original data from location A is evicted to external memory, overwriting the data written by the external master. This only happens when software is allowing an external master to modify memory that is, write-back or write-allocate in the core page tables, and depending on the fact that the data is not 'dirty' in the cache, to preclude the cached version from overwriting the external memory version. When there are any semaphores or any other handshaking to prevent collisions on shared memory, this is not a problem.

Workaround: For this shared memory region, mark it as write-through memory in the core page table. This prevents the data from ever being written out as dirty.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

3. Performance Monitor Unit event 0x1 can be incremented erroneously by unrelated events

Event 0x1 in the performance monitor unit (PMU) can be used to count cycles in which the instruction cache cannot deliver an instruction. The only cycles counted should be those due to an instruction cache miss or an instruction TLB miss. The following unrelated events in the core, also causes the corresponding count to increment when event number 0x1 is being monitored:

- Any architectural event (e.g., IRQ, data abort)
- MSR instructions which alter the CPSR control bits.
- Some branch instructions, including indirect branches and those mispredicted by the BTB.
- CP15 MCR instructions to registers 7, 8, 9, or 10 which involve the instruction cache or the instruction TLB.

Each of the preceding items may cause the performance monitoring count to increment several times. The resulting performance monitoring count may be higher than expected, when the preceding items occur, but should never be lower than expected.

Workaround: There is no way to obtain the correct number of cycles stalled due to instruction cache misses and instruction TLB misses. Extra counts, due to branch instructions mispredicted by the BTB, may be one component of the unwanted count that can be filtered out.

The number of mispredicted branches also can be monitored using performance monitoring event 0x6 during the same time period as event 0x1. The mispredicted branch number then can be subtracted from the instruction cache stall number generated by the performance monitor to get a value closer to the correct one. This workaround only addresses counts contributed by branches that the BTB is able to predict.

All the items in the preceding bulleted list still affect the count. Depending on the nature of the code being monitored, this workaround may have limited value.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

4. In Special Debug State, back-to-back memory operations — where the first instruction aborts — may cause a hang

Problem: When back-to-back memory operations occur in the Special Debug State (SDS, used by ICE and Debug vendors) and the first memory operation gets a precise data abort, the first memory operation is correctly cancelled and no abort occurs. Depending on the timing, however, the second memory operation may not work correctly. The data cache may internally cancel the second operation, but the register file may have scoreboarded registers for that second memory operation. The effect is that the core may hang (due to a permanently scoreboarded register) or that a store operation may be incorrectly cancelled.

Workaround: In Special Debug State, any memory operation that may cause a precise data abort should be followed by a write-buffer drain operation. This precludes further memory operations from being in the pipe when the abort occurs. Load Multiple/Store Multiple that may cause precise data aborts should not be used.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

5. Accesses to the CP15 ID register with opcode2 > 0b001 returns unpredictable values

Problem: The *ARM Architecture Reference Manual* (ARM DDI 0100E) states the following in chapter B-2, section 2.3:

When an <opcode2> value corresponding to an unimplemented or reserved ID register is encountered, the System Control processor returns the value of the main ID register. ID registers other than the main ID register are defined so that when implemented, their value cannot be equal to that of the main ID register. Software can therefore determine whether they exist by reading both the main ID register and the desired register and comparing their values. When the two values are not equal, the desired register exists.

The Intel XScale® core does not implement any CP15 ID code registers other than the Main ID register (opcode2 = 0b000) and the Cache Type register (opcode2 = 0b001). When any of the unimplemented registers are accessed by software (e.g., mrc p15, 0, r3, c15, c15, 2), the value of the Main ID register was to be returned. Instead, an unpredictable value is returned.

Workaround: No workaround.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

6. Disabling and re-enabling the MMU can hang the core or cause it to execute the wrong code

Problem: When the MMU is disabled via the CP15 control register (CP15, CR1, opcode_2 = 0, bit 0) after being enabled, certain timing cases can cause the processor to hang. In addition to this, re-enabling the MMU after disabling it can cause the processor to fetch and execute code from the wrong physical address. To avoid these issues, the code sequence below must be used whenever disabling the MMU or re-enabling it afterwards.

Workaround: The following code sequence can be used to disable and/or re-enable the MMU safely. The alignment of the mcr instruction that disables or re-enables the MMU must be controlled carefully, so that it resides in the first word of an instruction cache line.

```
@ The following code sequence takes r0 as a parameter. The value of r0 will be
@ written to the CP15 control register to either enable or disable the MMU.
mcr p15, 0, r0, c10, c4, 1 @ unlock I-TLB
mcr p15, 0, r0, c8, c5, 0 @ invalidate I-TLB
mrc p15, 0, r0, c2, c0, 0 @ CPWAIT
mov r0, r0
sub pc, pc, #4
b 1f @ branch to aligned code
.align 5
1:
mcr p15, 0, r0, c1, c0, 0 @ enable/disable MMU, caches
mrc p15, 0, r0, c2, c0, 0 @ CPWAIT
mov r0, r0
sub pc, pc, #4
```

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

7. Updating the JTAG parallel registers requires an extra TCK rising edge

Problem: The IEEE 1149.1 spec states that the effects of updating all parallel JTAG registers should be seen on the falling edge of TCK in the Update-DR state. The Intel XScale® core parallel JTAG registers, incorrectly require an extra TCK rising edge to make the update visible. Therefore, operations like hold-reset, JTAG break, and vector traps require either an extra TCK cycle by going to Run-Test-Idle or by cycling through the state machine again in order to trigger the expected hardware behavior.

Workaround: When the JTAG interface is polled continuously, this erratum has no effect. When not, an extra TCK cycle can be caused by going to Run-Test-Idle after writing a parallel JTAG register.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

8. Non-branch instruction in vector table may execute twice after a thumb mode exception

Problem: When an exception occurs in thumb mode and a non-branch instruction is executed at the corresponding exception vector, that instruction may execute twice. Typically instructions located at exception vectors must be branch instructions which go to the appropriate handler, but the ARM architecture allows the FIQ handler to be placed directly at the FIQ vector (0x0000001c/0xffff001c) without requiring a branch. Because of this bug, the first instruction of such an FIQ handler may be executed twice when it is not a branch instruction.

Workaround: When a ‘NOP’ is placed at the beginning of the FIQ handler, the ‘NOP’ executes twice and no incorrect behavior results. When a branch instruction is placed at the beginning of the handler, it does not executed twice.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

Specification Changes

1. HPI# (High Priority Interrupt) is a maskable interrupt

Issue: The HPI# interrupt input is both maskable and masked by default (as are all interrupts). It is controlled by INTCTL1.31. HPI# operates the same as the other external interrupt inputs (S_INT[D:A]#).

2. PCIODT_EN Reset Strap Signal

Issue: PCI Bus ODT Enable (PCIODT_EN) is a reset strap muxed onto signal A[20] and is latched on the rising (de-asserting) edge of P_RST#. It is used to determine when the secondary PCI(-X) bus uses internal pull-ups on the following signals:

S_AD[63:32], S_C/BE[7:4]#, S_PAR64, S_REQ64#, S_REQ[3:0]#, S_ACK64#, S_FRAME#, S_IRDY#, S_DEVSEL#, S_TRDY#, S_STOP#, S_PERR#, S_LOCK#, S_M66EN, S_SERR# and S_INT[D:A]#.

When disabled, then external pull-up resistors are required. PCI ODT enable is valid for the 80331 secondary PCI bus only.

Secondary PCI bus options:

1. PCIODTEN = 1 (default), enables 8.2 K internal pull-ups to 3.3 V.
2. PCIODTEN = 0, external 8.2 K pull-ups to 3.3 V are required.

3. LOCK# functionality has been de-featured

Issue: The LOCK# functionality of the PCI-X bridge has been de-featured from the 80331.

4. Watchdog Timer and Retry Timer has been de-featured

Issue: The Watchdog Timer and Retry Timer of the PCI-X bridge has been de-featured from the 80331.

5. P_GNT# and P_REQ# signals have new ball locations on B-0

Issue: The location of P_REQ# and P_GNT# on A-1 causes a violation of the maximum trace length required by the *PCI Local Bus Specification*, Revision 2.3:

P_REQ# signal is moving from T6 on A-1 to H11 on B-0.

P_GNT# signal is moving from R4 on A-1 to G12 on B-0.

The 80331 boards need to incorporate series resistors that can be populated/de-populated for connection to either the A-1 or B-0 P_REQ#/P_GNT# ball location. Route the P_REQ# net to the two ball locations as separate routes through two 0 ohm series resistor located near the PCI edge connector. Route the P_GNT# net to the two ball locations as separate routes through two 0 ohm series resistor located near the PCI edge connector. Populate the resistor that connects the net to the correct ball according to the silicon revision.

6. Peripheral Performance Monitor Unit has been de-featured

Issue: The Peripheral Performance Monitor Unit has been de-featured. The Intel XScale® core PMON is still functional.

7. Processor Device ID has been removed

Issue: The Processor Device ID (PDIR) at FFFF E780h is suppose to mirror the JTAG ID, but instead has a value of 0x0. This register is not to be fixed in future steppings, since there are other registers that can be read to determine processor type and revision information.

8. 1.5K pull-down required on AD[15] of the PBI bus

Issue: In order to prevent secondary PCI clock instabilities, make sure AD[15] of the PBI bus has a 1.5 K pull-down.

9. OCD and Receive Enable calibration de-featured

Issue: The ability to adjust the electrical interface to account for out of specification DDR-II DIMMs using OCD (off-chip driver) and receive enable calibration, is no longer a supported feature.

10. New Watchdog Timer (WDT) functionality in B-0 stepping

Issue: Watchdog timer (WDT) expiration can either generate a chip reset or a core interrupt. Added Watchdog timer interrupt control to the following register bits: INTCTL0.17, INTSTR0.17, IINTSRC0.17, FINTSRC0.17, IPR1.3:2

Control for WDT reset or interrupt generation is done by the interrupt mask bit, INTCTL0.17. When clear (default), the WDT generates a reset. When set, the WDT generates an interrupt to the core. These functions are mutually exclusive. The WDT is disabled by default see the developers manual for the enabling sequence.

11. PWRDELAY needs only a pull-up for battery back-up mode

Issue: The I^2C bus has been removed from the processor reset equation (see erratum #35). This allows the PWRDELAY circuit to be simplified to a single pull-up resistor, to enable battery back-up mode. PWRDELAY must be isolated from all other circuitry, and only a 1.5K pull-up to 3.3 V is required. When battery back-up is not required, PWRDELAY should have a 1.5 K pull-down resistor.

Note: In some applications PWRDELAY may control other board logic. Before making this change, make sure the other logic is not adversely impacted.

12. ARB_EN signal has been de-featured

Issue: The ARB_EN reset strap, muxed on AD[1], has been de-featured from the 80331. When using ‘no bridge’ mode (BRG_EN=0), ARB_EN needs to be pulled low.

13. *Intel® 80331 I/O Processor Design Guide* change for Unbuffered DDR-I dual-banked DIMMs

Issue: The latest routing guidelines for unbuffered DDR-I dual-banked DIMMs are available upon request and is included in the next revision of the design guide.
Update - These routing guidelines were added to the October 2004 release of the 80331 design guide (273823-002)

14. DDRRES2 can be pulled down to reduce current during self-refresh

Issue: DDRRES2 is used as compensation for DDR-II OCD. Since OCD is not supported in the 80331 (see Specification Change #9), then DDRRES2 can be pulled down to reduce current draw during self-refresh mode.

15. *Intel® 80331 I/O Processor Design Guide* change for Peripheral Bus Interface (PBI)

Issue: The latest routing guidelines for the Peripheral Bus Interface (PBI) are available upon request and included in the next revision of the *Intel® 80331 I/O Processor Design Guide*.

Update - These routing guidelines were added to the October 2004 release of the 80331 design guide (273823-002)

16. *Intel*® 80331 *I/O Processor Design Guide* change for PCI/-X busses

Issue: The latest routing guidelines for the PCI/-X busses are available upon request and included in the next revision of the *Intel® 80331 I/O Processor Design Guide*.
Update - These routing guidelines were added to the October 2004 release of the 80331 design guide (273823-002)

17. Internal bus operates at 333 MHz for D-0 stepping

Issue: For the D-0 stepping of the 80331, the internal bus operates at 333 MHz. For all previous steppings, the internal bus runs at 266 MHz.

18. Application Accelerator Unit enhanced for D-0 stepping

Issue: The Application Accelerator Unit (AAU) in the D-0 stepping has been enhanced to support RAID 6 functionality. The details of this new feature are described in an addendum to the AAU chapter.

19. Recommended DLL register values

Issue: Using the default DLL values in combination with low duty cycle DIMMs may result in low hold time margins on read transactions.

The default values for the DLL master and slave registers do not center the internal DQS with respect to the eye of the incoming data. Using the default values may result in a reduced hold time due to DQS being late in the data eye, which could lead to ECC errors. Errors have only been observed when using DIMMs that have a low DQS duty cycle.

Note: All of the Intel validation up to this point has been with the default, worst case DLL values and all DIMMs used in validation have passed.

Firmware should be updated and tested with these new DLL values, in order to add margin to the hold timing during memory reads.

DDR-II 400 settings
SLVLMIX0 - Address FFFF_F554h; Recommended value - 3333_3333h
SLVLMIX1 - Address FFFF_F558h; Recommended value - 0000_0003h
SLVHMIX0 - Address FFFF_F55Ch; Recommended value - 3333_3333h
SLVHMIX1 - Address FFFF_F560h; Recommended value - 0000_0003h
SLVLEN - Address FFFF_F564h; Recommended value - 0000_0003h
MASTMIX - Address FFFF_F568h; Recommended value - 0000_000Ah
MASTLEN - Address FFFF_F56Ch; Recommended value - 0000_0002h

DDR-I 333 settings
SLVLMIX0 - Address FFFF_F554h; Recommended value - 6666_6666h
SLVLMIX1 - Address FFFF_F558h; Recommended value - 0000_0006h
SLVHMIX0 - Address FFFF_F55Ch; Recommended value - 6666_6666h
SLVHMIX1 - Address FFFF_F560h; Recommended value - 0000_0006h
SLVLEN - Address FFFF_F564h; Recommended value - 0000_0003h
MASTMIX - Address FFFF_F568h; Recommended value - 0000_0000h
MASTLEN - Address FFFF_F56Ch; Recommended value - 0000_0002h

20. DDR-II JEDEC initialization sequence includes writes to EMRS2 and EMRS3

Issue: The JEDEC DDR-II specification includes a write to EMRS2 and EMRS3 (Extended Mode Register Set) during the initialization sequence. Step 5 is ‘Issue EMRS2 command’ and step 6 is

'Issue EMRS3 command'. In order to be JEDEC compliant, these steps should be added to the memory controller initialization sequence.

Note: Before implementing, check with your DIMM/memory manufacturer to determine if these steps are necessary. Software should always follow the initialization sequence provided by the DIMM/memory manufacturer guidelines.

The following pseudo code shows the EMRS initialization steps that are required to be compliant with the JEDEC DDR-II initialization sequence.

```
// Step 5 and 6 - EMRS(2) and EMRS(3) programming
if MemoryType is DDR-II
Write 0x2 to DCALADDR (Setting BA[1:0] for EMRS(2))
Write 0x81000003 to DCALCSR (issue EMRS command to CS0)
Wait for DCALCSR[31] to report 'Operation Completed'
Write 0x81100003 to DCALCSR (issue EMRS command to CS1)
Wait for DCALCSR[31] to report 'Operation Completed'

Write 0x3 to DCALADDR (Setting BA[1:0] for EMRS(3))
Write 0x81000003 to DCALCSR (issue EMRS command to CS0)
Wait for DCALCSR[31] to report 'Operation Completed'
Write 0x81100003 to DCALCSR (issue EMRS command to CS1)
Wait for DCALCSR[31] to report 'Operation Completed'
endif
```

21. Case temperature (Tcase) change

Issue: To be consistent with the production test environment, the case temperature (Tcase) for the 80331 I/O processor has been changed from 105° c to 95° c.

22. Internal Clock Misalignment

Issue: Due to non-core Erratum 62, Internal Clock Misalignment Can Cause Processor Hang, on page 41, Intel is screening parts to eliminate the probability of occurrence. Until this is fixed in a future stepping, a screen has been implemented which will screen out parts exhibiting this issue with a VCC15 greater than 1.46v.

With the screen at 1.46v, the on-board 1.5v power rail should not be allowed to go below 1.46v, as this would increase the risk of failure. The 1.5v rail minimum is currently specified in the datasheet as 1.425v, therefore for screened parts, the minimum is changed to 1.46v.

Status: Fixed. This issue was fixed in the D-1 stepping of the product.

Specification Clarifications

1. 64 MB and 2 GB DDR333 capacities not to be tested in post-silicon validation

Issue: Intel is not able to test 64 MB and 2 GB DDR333 DIMMs due to availability. Intel cannot guarantee proper functionality since validation cannot be completed.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

2. DDR-II 400 Unbuffered DIMMs are not supported

Issue: The Intel® 80331 I/O processor (80331) supports DDR333 buffered (registered) and unbuffered DIMMs, but only supports DDR-II 400 buffered (registered) DIMMs. DDR-II 400 unbuffered DIMMs are not supported.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

3. Interrupt behavior in the 80331 no bridge mode

Issue: When in the 80331 No Bridge Mode (BRG_EN = 0), the external interrupts behave just like when the bridge is enabled in the 80331 (BRG_EN = 1), in that the S_INT[D:A]# interrupt inputs can be forwarded to the P_INT[D:A]# interrupt outputs. The only difference between the two modes is how the internal Messaging Unit interrupt is forwarded. In the 80331 Mode, it is forwarded to P_INTC#, and in the 80331 No Bridge Mode it is forwarded on P_INTA#.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

4. Memory map for 2 GByte of DDR memory

Issue: The 80331 can support up to 2Gbytes of DDR SDRAM, but it cannot cross a 2GB boundary, therefore it must be mapped to either 0x00000000 - 0x7fffffff or 0x80000000 - 0xffffffff. Either range conflicts with one or more of the statically assigned regions. The recommendation is to disable the direct outbound ATU window, in order to use the larger 2GB memory, by clearing ATUCR.8 (default setting is ‘0’ - disabled).

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

5. Back to back MCU MMR reads

Issue: The memory controller unit (MCU) returns the wrong memory mapped register (MMR) read data, when two MMR read transactions are enqueued into the transaction queues at the same time. This cannot happen from the BIU port as mapping the MMRs to this space is illegal. The only way this can occur is for two internal bus devices to request info from the MCU MMRs at the same time (with different addresses). For example, the BIU (via the Intel XScale® core) and the ATU (via the host), which is a very unlikely usage model.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

6. Write requirements for the Peripheral Bus Interface

Issue: PBI write requirements are:

- must set up the Flash or memory to accept writes.
- must ensure the write has occurred before another one starts.
- it is illegal to burst writes to the PBI.

The data presented should not be larger than the PBI width.

Any PBI device should be mapped in the MMU the same. Making the device address space cachable can result in buffered/coalesced writes, which are burst to the PBI. XCB=000 and XCB=001 are the only cache policies that should be used for PBI. All other cache policies result in multi-byte transactions.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

7. PCI-X Status Register during PCI mode

Issue: The PCI-X Status Register (PX_SR, offset E4h-E5h) in the ATU has meaning only in PCI-X mode. The device number and bus number fields are always updated when a configuration write is detected. The ATU always grabs AD[15:11] for configuration writes, whether it is in PCI or PCI-X mode. When a PCI configuration transaction occurs, the Device Number bits (7:3) are updated to a value of “00000” from the default value of “11111”. The bus number bits (15:8) are only grabbed during the attribute phase (which does not exist for PCI).

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

8. M_RST# driven to DDR-II or DDR-I voltage levels

Issue: The de-asserted voltage level on M_RST# with DDR-II is 1.8 Volts and with DDR-I is 2.5 Volts. When M_RST# is needed for other devices (i.e,. - Flash), make sure these voltage levels are appropriate for the target device.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

9. BIU master abort causes two interrupts on reads.

Issue: The B-0 stepping added a Bus Interface Unit (BIU) interrupt when the BIU gets master aborted on an internal bus write (see erratum #8). The functionality is different between a read and write case. For a write, the BIU master abort asserts IINTSRC.29 (when enabled) and does not assert an error to the core. For a read, the BIU master abort asserts both IINTSRC[29] (when enabled) and an error to the core. Therefore, on a read case two interrupts is generated. When the interrupt source is masked, then only the master abort on reads is detected, and this is from the direct core error.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

10. Reset Internal Bus (PCSR.5) usage

Issue: PCI bus data corruption can occur when the reset internal bus bit is not used correctly. PCSR.5 can be used to reset the internal bus. This resets all memory mapped registers and the ATU configuration header space. This bit is read/write capable from both the internal bus and PCI bus. For both PCI and PCI-X modes, make sure the following steps occur when setting PCSR.5:

1. Clear Bus Master (ATUCMD.2) Enable and Memory Enable (ATUCMD.1).
2. Wait for both the outbound (PCSR.15) and inbound (PCSR.14) read transaction queue busy bits to clear.
3. Make sure no PCI configuration read/write cycles targeting the ATU are in progress, except the reset configuration write, when applicable.
4. Set the Reset Internal Bus bit.
5. Wait for 40 PCI clocks.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

11. No Bridge Mode (BRG_EN) validation

Issue: The BRG_EN signal (0 = no bridge, muxed on AD[0]) is used to disable the internal bridge, in order to operate as an I/O processor with a single PCI-X interface. This feature is not to be validated until after PTQ (prototype qualification).

Note: When central resource functionality is required, then the 80331 needs to be used with the bridge enabled.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

12. Potential race condition with Interrupt Controller Unit status bits

Issue: There is a slight lag in the time it takes between clearing a status bit inside the unit and the corresponding bit in the Interrupt Controller Unit Status Register getting cleared. This has the potential of generating a false interrupt, meaning that the Intel XScale® core is interrupted but the handler is not able to find any source reported in the ICU registers. This condition can be avoided by adding a read from any ICU register after the bit is cleared in the local unit before returning from the interrupt handler. The data from this read can be ignored, but the read itself creates enough latency to allow the updated status to propagate to the ICU

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

13. Bus Interface Unit follows PCI ordering rules

Issue: The Core Bus Interface Unit orders transactions based on PCI rules. This allows outgoing writes to pass incoming reads. For most devices on the internal bus, this does not cause problems since the devices function asynchronously with respect to each other. For transactions between the Intel XScale® core and memory via the internal bus, this can result in unexpected data. For example:

0: ldr r0, =0x40000

1: ldr r1, =0xaaaaaaaa

2: ldr r2, =0x55555555

3: str r1, [r0]

4: … <time-delay to allow the previous transactions to complete>

5: ldr r3, [r0]

6: str r2, [r0]

This code could potentially load the register r3 with the value 0x55555555 since the store in line 6 may pass the load in line 5. This only happens with transactions on the internal bus.

The MCU core port enforces strict ordering and does not exhibit this behavior. If the MCU core port is used, then this issue will not occur.

When caching is enabled, then the initial read will initiate a cache-line fill. The subsequent write is pended in the Intel XScale® core until the line fill and the ldr instruction complete. In this case, this issue does not occur.

When caching is disabled, and the caching policy is stall-until-complete (X=0, C=0, B=0), this issue does not occur. For other MMU settings with caching disabled, the issue can occur. Specifically regions with data cache and write buffer policies of bufferable (X=0, C=0, B=1) or coalescing-disabled-bufferable (X=1, C=0, B=1) are vulnerable to this issue. In addition, regions configured as write through (X=0, C=1, B=0) are also vulnerable to this issue.

In addition, if caching is enabled in the MMU page tables, but the DCache is disabled in the CP15 ARM Control Register, then the effective caching policy is bufferable and this reordering must be accounted for.

It is important to realize that any code which accesses memory spaces on the internal bus need to account for this possibility. Code which dynamically disables cache (i.e. - flash programming routines) needs to ensure that the caching policy for the appropriate memory region is set to “stall until complete” until the cache is re-enabled.

The simplest scenario to reproduce this is two back-to-back function calls, for example:

```
main:
        bl      fun1
        bl      fun2

        …
        …
        …
fun1:
        stmfd   sp!, {r4, r5, r6, r7, r8, r9, r10, lr}
        ldmfd   sp!, {r4, r5, r6, r7, r8, r9, r10, pc}
fun2:
        stmfd   sp!, {r4, fp, ip, lr}
        ldmfd   sp!, {r4, fp, ip, pc}
```

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

14. UART, I2C and GPIO memory mapped registers should be addressed with 32-bit accesses

Issue: The UART, I2C and GPIO units sit on a dedicated low speed internal bus that does not support byte enables. Due to this functionality, accessing any of these unit memory mapped registers (MMR) with any accesses less than 32-bits can result in corruption of the other bits in the 32-bit MMR. For example, beginning with C-0 stepping, the GPOD register (located at FFFF_F788h) added new bit functions to bits 10 and 11. When software does a byte access to GPOD, this could cause bits 10 and 11 to be written with incorrect data.

While most of these registers only implement the lower 8-bits (the upper three bytes are ‘reserved’), the recommendation is that all UART, I2C and GPIO MMRs should only be accessed as 32-bit registers. While it is desired that 32-bit accesses be performed, it is acceptable to access with less than 32-bits, as long as all non-reserved bits are accessed. For purposes of future expansion, 32-bit accesses are preferred.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

15. Flash Wait States

Issue: The Address-to-Data wait state and Recovery Cycle wait state fields in Table 60 (PBBAR0) and 62 (PBBAR1) are incorrect in the *Intel® Lindsay I/O Processor Component Specification*, Rev. 2.0, Vol. 2. The Address-to-Data wait state is actually one more than listed, and the Recovery Cycle wait state is actually two more than listed. See Documentation Change # 8 for more specific information.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

16. UART Interrupt Identification Register

Issue: The UART Interrupt Identification Register (UxIIR) is read by software to determine the type and source of UART interrupts. This register gathers and priority encodes the various sources of UART interrupts. The register is read after an interrupt occurs. Enabling and disabling of interrupts (via the Interrupt Enable Register - UxIER or Modem Control Register - UxMCR) effects whether or not the interrupt to the processor occurs. This does not effect the logging of the status of what is happening in the UART. The UART operates in interrupt or polling mode. In polling mode, all interrupts to the processor would be disabled.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

17. Reads on 16-bit PBI bus operate as 32-bit

Issue: 2-byte and 4-byte read transactions on the Peripheral Bus Interface (PBI) bus operate as burst reads (in other words, two 16-bit read cycles). All the read transactions from the Intel XScale® core to PBI devices (in other words, SRAM, Flash, etc.) are translated to burst reads with burst size of 2, even though there is no necessity to generate a burst transaction. Therefore, devices on the 16-bit PBI bus should be configured as pre-fetchable.

Note: 1-byte transactions on 16-bit PBI bus is not a supported case. Also, a PBI bus configured as 8-bit does not operate this way.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

18. Embedded Design Usage Model - Secondary PCI bus only

Issue: The 80331, with bridge enabled, can be used in embedded designs that require central resource functionality. The secondary PCI bus provides clockouts, arbitration and interrupt inputs for up to four devices. The ATU can generate configuration cycles to all devices on the secondary PCI bus.

In contrast, the primary PCI bus cannot be used in embedded designs, therefore no PCI devices can be placed on this bus. The primary PCI bus was designed to be driven by a central resource only. Clockouts, arbitration and interrupt inputs are not provided on the primary PCI bus. When the ATU generates configuration cycles they are not forwarded from the secondary PCI bus to the primary PCI bus. Also, the Intel XScale® core cannot write to the bridge configuration registers.

When using the 80331 in embedded designs, keep the bridge enabled (BRG_EN=1) and connect the primary PCI bus as follows:

1. P_CLK is still required. Clock frequency should be 66 MHz, since external pull-ups need to be on P_M66EN, P_DEVSEL#, P_STOP# and P_TRDY#. With these signals pulled high, the 80331 is put into conventional PCI mode expecting a 66 MHz input clock. If providing a 33MHz clock is easier/cheaper, then tie P_M66EN low.
2. P_RST# still required from an external reset source.
3. P_RCOMP needs 100 Ω to ground
4. With P_REQ64# tied high, P_AD[63:32], P_PAR64, and P_C/BE[7:4]# have internal pull-ups
5. Put pull-ups (8.2 K to 3.3V) on all other primary PCI signals.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

19. 3.3 V to 1.5 V leakage

Issue: There is a leakage path from 3.3 V rail to the 1.5 V rail. When the 3.3 V is powered on and the 1.5 V is not, then ~500 mV is seen on the 1.5 V rail. This leakage is expected and does not cause any long-term reliability issues.

For related issues, see Documentation Change 10 (“Power sequence timing” on page 66) and Non-Core Errata 50 (“Secondary bus PCI RST# pulse prior to the rising edge of P_RST#” on page 36).

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

20. Accessing “reserved” registers in “no bridge” mode

Issue: In general, accessing “reserved” registers or bit fields must never be done, as these “reserved” fields might be used for test information or might be implemented in future product revisions. Specifically, accessing peripheral memory mapped registers at FFFF_F5D0h to FFFF_F5DFh when in “no bridge” mode, causes the 80331 to hang. Do not access these registers.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

21. Power plane isolation for Battery Back-Up (BBU) mode

Issue: During battery back-up (BBU) mode, when the battery powers the DIMM and the VCC25/18 signals (1.8 V or 2.5 V, depending on the memory type being used) on a single power plane, the battery life is probably reduced, due to leakage.

To attain longer battery life, the DIMM and VCC25/18 power planes must be isolated. The power-plane isolation can be accomplished by using a FET.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

22. AAU result can be written directly to PCI host memory

Issue: The Application Accelerator Unit (AAU) can write results not only to local memory but also to the PCI bus host memory via the ATU.

This feature can be applied to degraded RAID-5 reads, where the AAU result is the reconstructed data for the host I/O read. The AAU can write its result to PCI; therefore, the degraded read XOR result can be written directly to host memory. This eliminates the need for a DMA operation to transfer the result from local memory to host memory via PCI.

Savings for the RAID application include the following:

- No DMA descriptor needs to be generated.
- No DMA interrupt needs to be serviced.
- Memory and internal bus bandwidth is saved (result write by AAU and read by DMA).
- Read I/O is serviced faster (eliminates latency of DMA operation).

Note: AAU source reads from PCI are ***not*** supported; only local memory can be used for this.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

23. PWRDELAY functionality during power sequencing

Issue: When the 3.3 V rail is powered on and the 1.5 V rail is powered off, the PWRDELAY input signal drives out until the 1.5 V rail powers up. This is important to understand if still using the legacy power-fail circuit, because it might cause other circuitry to function incorrectly. The proper usage of PWRDELAY is described in specification change 11 (“PWRDELAY needs only a pull-up for battery back-up mode”), which recommends that PWRDELAY be isolated from all circuitry and tied to a 1.5 K pull-up resistor.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

24. PBI lockout condition

Issue: When the core is in a tight loop writing to the PBI bus, while the DMA is doing a large block transfer (for example, from SRAM, located on the PBI, to DDR memory), the DMA can be locked out of accessing the PBI and the transaction will never complete.

If this condition occurs, use one of these workarounds:

1. Change the MTTR1 from 98h (default) to a lower value (such as 01h). A lower value allows the DMA (or ATU which can also master a transaction to the PBI) to gain access to the PBI, because the BIU is given shorter access for back-to-back BIU internal bus transactions.

2. Add a core read along with the core write, causing it to stall and preventing it from starving the DMA.

3. Add NOPs or dummy instructions to ensure the loop spans greater than two cache lines.

4. Modify the loop such that the write is not done on every iteration.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

25. Interleaving descriptors with D-0 AAU

Issue: The P+Q capability is enabled in the AAU globally (ACR.3), not on a descriptor by descriptor basis.

Therefore if enabled, all descriptors are processed as P+Q descriptor formats.
If disabled, all descriptors are processed as prior AAU definitions (ie - straight XOR).

In order to mix RAID-5 with P+Q RAID-6, enable P+Q RAID-6 GF Multiply for the AAU, and build all RAID-5 XOR descriptors as P+Q RAID-6 descriptors, where the GF Multiplier Byte values are all 0x01.

Status: No Fix. See the Table , "Summary Table of Changes" on page 9.

26. RCVDLY setting for DDR-I memory

Issue: The Receive Enable Delay Register (RCVDLY at FFFF_F550h) is used for DQS receive enable calibration. In other words, RCVDLY adjusts the memory controllers relationship of DQS to an internal M_CLK.
The RCVDLY value is highly dependant on the board layout and DIMM characteristics. Also, the memory controller only supports a non integer CAS latency (tCAS = 2.5, SDCR0.9:8) for DDR-I, which means that RCVDLY may need to be adjusted because DQS is no longer synchronized with M_CLK.
Therefore, when using DDR-I memory, the RCVDLY default setting of 5 may need to be changed to 6 or 7 to operate correctly with a specific DIMM based on the board layout. For example, the Redboot reference code provided by Intel uses a value to 7 to allow for a wider compatibility with various DIMMs.

Status: No Fix. See the Table , "Summary Table of Changes" on page 9.

27. ATUBAR3 Functionality

Issue: The private memory window of the secondary PCI segment defines an address range that the 80331 uses to map private devices to, and to locate local memory for private device access. This range is intended to be mapped to the ATUs private BAR window (ATUBAR3) and the private device BARs. Note that even when the private addressing is enabled, the normal 80331 behavior defined for BME, MSE, IOSE bits in the ATUCMD register are still true. Therefore, when the ATU Memory Space Enable bit is cleared, all ATU BARs including ATUBAR3 will be unable to claim any memory transactions. For example, this bit is typically cleared during a PCI bus scan / enumeration.

Status: No Fix. See the Table , "Summary Table of Changes" on page 9.

28. VREF isolation for Battery Back-up (BBU) mode

Issue: During battery back-up (BBU) mode, the DIMM power can be isolated from the 80331 IOP power. This isolation should also include the VREF signal for the DIMM interface. Due to leakage, the VREF signal for the DIMM should be isolated from the VREF signal for the IOP. This is to ensure that VREF for the DIMM is not disturbed as the IOP powers down when entering battery back-up (BBU) mode. The isolation can be provided by using separate voltage dividers or a FET.

For related issues, refer to Specification Clarification 21, "Power plane isolation for Battery Back-Up (BBU) mode" on page 56.

Status: No Fix. See the Table , "Summary Table of Changes" on page 9.

29. I2C Unit Enabling

Issue: Software must guarantee that the I2C bus is idle before enabling the I2C unit. Failure to do so could result in unstable behavior. The IBMR register can be used to monitor the state of the SCL and SDA pins in order to determine bus activity. The I2C Bus Busy bit in the I2C Status Register (ISR.3) can not be used for this purpose, as it is only valid when the I2C unit is enabled.

Status: No Fix. See the Table , "Summary Table of Changes" on page 9

30. DMA transactions from local memory to a conventional PCI target can complete out of order

Issue: It is possible for the Outbound ATU to get backed up and retry the DMA unit. When this occurs, the possibility exists for DMA transactions to complete out of order as each DMA has 3 separate data queues and there is no guarantee as to which one of the queues will drain on the retry.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9

31. SBR1 Programming with Bank 1 Unpopulated

Issue: When using single-banked DDR-II DIMMs and the SDRAM Boundary Register 1 (SBR1; FFFF_E514h) is not programmed to match the SDRAM Boundary Register 0 (SBR0; FFFF_E510h), the wrong ODT signal will become activated. The memory controller provides two ODT (On Die Termination) signals (ODT[1:0]) which are used with DDR-II DIMMs to turn on termination during writes.

Section 8.7.6 in the *Intel® 80331 I/O Processor Developer's Manual* (273942) states, “If bank 1 is unpopulated, SBR1[6:0] is programmed either with all zeroes or a value equal to SBR0[6:0].” To clarify this statement for single-banked DDR-II DIMMs, if bank 1 is unpopulated, then the entire SBR1 must be programmed the same as SBR0. This includes lower bits 06:00 and upper bits 31:30. Bits 30:07 are reserved and should not be written. The memory controller compares the entire range of the SBR0 and SBR1 to determine which ODT signal to enable. If the upper bits 31:30 in SBR0 and SBR1 do not match, then ODT1 will become active instead of ODT0.

In addition, when bank 1 is unpopulated, SBR1[6:0] should never be zero if SBR0[6:0] is non-zero.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

32. 32-bit Writes to Unaligned 64-bit Addresses are Promoted to 64-bit Aligned Writes

Issue: In applications that run the PCI bus segment in 32-bit PCI Mode or 64-bit PCI Mode with 32-bit targets, write transactions that are on unaligned 64-bit addresses are promoted to 64-bit aligned writes. The first half of the 64-bit write is on a 64-bit aligned address and has the BE# signals disabled. Therefore, the write is invalid. The second half on the 64-bit write is a valid write with the BE# enabled and the write is to the intended 32-bit address.

Per the PCI Local Bus Specification, Revision 2.2, PCI compliant devices should ignore the first half of the 64-bit write due to the BE# signals being disabled.

For devices that support using the I/O window, the 64-bit write does not occur when using the ATU I/O Window and only the expected 32-bit write occurs.

For memory mapped devices, the only option is to run in PCI-X mode, where the byte count and starting address are consistent with the actual number of bytes to be written (i.e., 4). When a 64-bit PCI-X request gets downshifted, the requester can use the starting address/byte count to recognize that the write request does not cross a DWORD address boundary and only perform a single 32-bit wide data cycle.

Status: No Fix. See the Table , “Summary Table of Changes” on page 9.

33. Case Temperature Clarification

Issue: Internal models indicate that certain elevated case temperatures (Tcase) of the 80331 may cause elevated field failures in later years of operation. This clarification is precautionary, as Intel has not seen any failures of the 80331 products in the field. Internal modeling data indicates that the degree of failure risk decreases with lower operating frequency (i.e. – 500, 667, or 800MHz) and temperature (i.e. – case temperature). If a failure should occur, it would manifest itself as a hard failure of the I/O processor which can cause a system failure.

A detailed analysis of the system operating conditions, specifically case temperature of the 80331 I/O processor, is required. The 80331 should be run with Tcase temperatures that, over the lifetime of the part, average less than the 95°C maximum Tcase that is currently specified in the *Intel® 80331 I/O Processor Datasheet* (273943).

Tcase recommendations for the 80331:

Frequency	5 year operating life	6 year operating life	7 year operating life
500MHz	95° C	93° C	91° C
667MHz	85° C	83° C	81° C
800MHz	79° C	76° C	74° C

Note: Tcase temperatures in the table reflect an average Tcase temperature over the lifetime of the product.

Status: Fixed. This issue was fixed in the D-1 stepping of the product. See the Table , “Summary Table of Changes” on page 9.

Documentation Changes

1. Table 236 shows incorrect description for tWDL field

Problem: tWDL field (bits 13:12) shows ‘00’ equals DDR-II Type only.

Workaround: Should state ‘DDR-I Type only’

Affected Docs: *Intel® 80331 I/O Processor Developer’s Manual*

2. Processor Device ID should be removed

Problem: Page 739, top of the table shows Processor Device ID Register.

Workaround: This register is no longer valid and should be removed, see specification clarification #7.

Affected Docs: *Intel® 80331 I/O Processor Developer’s Manual*

3. Core Performance Monitoring Unit (CPMON) section is incorrect

Problem: Incorrect and incomplete information regarding the Core Performance Monitoring Unit.

Workaround: Changes include additional control registers (Interrupt, Overflow, Event) and two more counter (PMN2 and PMN3). See the *Intel XScale® Core Developer’s Manual* (273473), referred to in the manual as ‘XSC2’ core.

Affected Docs: *Intel® 80331 I/O Processor Developer’s Manual*

4. UART FIFO Occupancy Register is read only

Problem: Table 312 shows UART FIFO Occupancy register (UxFOR) as ‘read/write’.

Workaround: Needs to be changed to ‘read-only’.

Affected Docs: *Intel® 80331 I/O Processor Developer’s Manual*

5. Refresh Frequency Register table is incorrect

Problem: The values listed in Table 230 are incorrect.

Workaround: Should be as follows:
333 MHz: 500h (7.8 µs value); A00h (15.6 µs value)
400 MHz: 600h (7.8 µs value); C00h (15.6 µs value)

Affected Docs: *Intel® 80331 I/O Processor Developer’s Manual*

6. PBI interrupt should be removed

Problem: PBI interrupt should not be included in the 80331 documentation.

Workaround: All references to the Peripheral Bus Unit Error Interrupt should be removed. This can be found in Figure 116, INTCTL1.21, INTSTR1.21, IINTSRC1.21, FINTSRC1.21 and IPR3.

Affected Docs: *Intel® 80331 I/O Processor Developer’s Manual*

7. ARB_EN has been de-featured

Problem: The ARB_EN signal should not be included in the 80331 documentation.

Workaround: Remove all references to the ARB_EN signal.

Affected Docs: *Intel® 80331 I/O Processor Developer's Manual*
Intel® 80331 I/O Processor Datasheet
Intel® 80331 I/O Processor Design Guide

8. Flash wait state information is incorrect

Problem: The Address-to-Data wait state and Recovery Cycle wait state fields in Table 275 (PBBAR0) and Table 276 (PBBAR1) are incorrect.

Workaround: The Address-to-Data wait states are defined in bits 4:2. The wait states are actually one more than listed. Therefore the following change applies:
000 - 5 Address-to-Data wait states
001 - 9 Address-to-Data wait states
010 - 13 Address-to-Data wait states
011 - 17 Address-to-Data wait states
Others (default) - 21 Address-to-Data wait states

The Recovery Cycle wait states are defined in bits 8:6. The wait states are actually two more than listed. Therefore the following change applies:
000 - 3 Recovery wait states
001 - 6 Recovery wait states
010 - 10 Recovery wait states
011 - 14 Recovery wait states
100 - 18 Recovery wait states
Others (default) - 22 Recovery wait states

This change also affects Table 271, Flash Wait State Profile Programming.

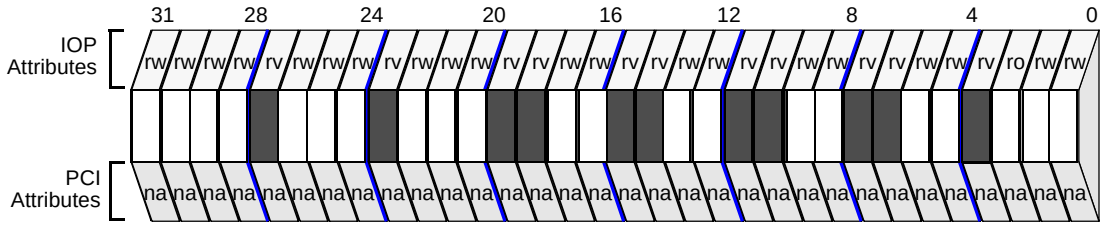
Affected Docs: *Intel® 80331 I/O Processor Developer's Manual*

9. SDCR0 and SDCR1 register updates

Problem: The SDCR0 (Table 236) and SDCR1 (Table 237) register description needs to be updated.

Workaround:

Table 236. DDR SDRAM Control Register 0 - SDCR0 (Sheet 1 of 2)

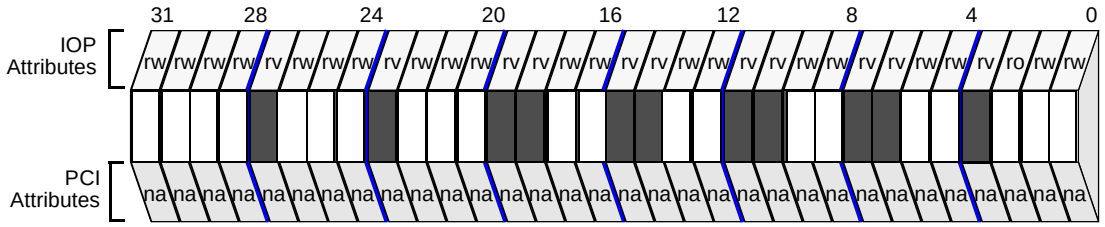


Intel XScale® core Local Bus Address
FFFF E504H

Attribute Legend:
RV = Reserved
PR = Preserved
RS = Read/Set
RW = Read/Write
RC = Read Clear
RO = Read Only
NA = Not Accessible

Bit	Default	Description
31:28	0H	**RAS:** Active to Precharge duration in MCLK periods **Equation 7: RAS = tRAS - 1** where tRAS is from SPD
27	0_2	Reserved
26:24	000_2	**RP:** Precharge Command Period in MCLK periods **Equation 8: RP = tRP - 1** where tRP is from SPD
23	0_2	Reserved
22:20	000_2	**RCD:** Active to Read, Active to Write Period in MCLK periods **Equation 9: RCD = tRCD - 1** where tRCD is from SPD
19:18	00_2	Reserved
17:16	01_2	**EDP:** Data Path Latency in MCLK periods This field should be programmed to 10_2.
15:14	00_2	Reserved
13:12	00_2	WDL: Write Data Latency in MCLK periods used by the Memory Controller state machine. See Equation 11. • 00 = 0 MCLK periods (tCAS = 2.5) - (DDR-I Type only) • 01 = 1 MCLK period (tCAS = 3) - (DDR-II Type only) - *not supported* • 10 = 2 MCLK periods (tCAS = 4) - (DDR-II Type only) 11 = reserved where tCAS is from SPD
11:10	00_2	Reserved
09:08	00_2	**CAS: CAS Latency:** Indicates the CAS Latency used by the Memory Controller state machine. Encoded value based on tCAS from SPD • 00 = reserved • 01 = 2.5 MCLK periods (DDR-I Type only) • 10 = 3 MCLK periods (DDR-II Type only) - *not supported* • 11 = 4 MCLK periods (DDR-II Type only)

Table 236. DDR SDRAM Control Register 0 - SDCR0 (Sheet 2 of 2)

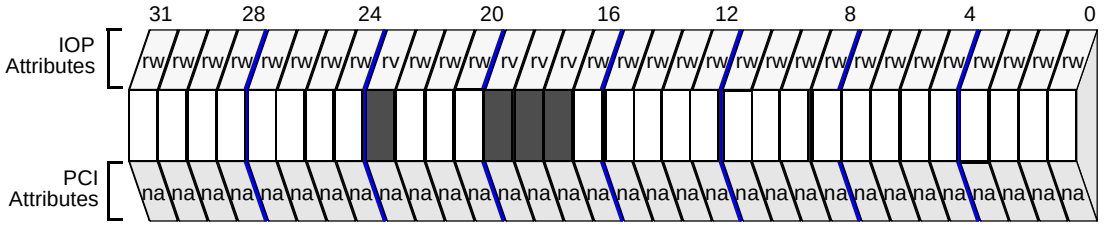


Intel XScale® core Local Bus Address
FFFF E504H

Attribute Legend:
RV = Reserved
PR = Preserved
RS = Read/Set
RW = Read/Write
RC = Read Clear
RO = Read Only
NA = Not Accessible

Bit	Default	Description
07:06	00_2	Reserved.
05:04	00_2	ODT Termination Value: Determines the termination value of the On Die Termination for both Banks (controlled by **ODT[1:0]**). Applies to DDR-II SDRAM memory type only. • 00 Disabled • 01 75 ohm • 10 150 ohm • 11 reserved
03	0_2	Reserved
02	Varies with external state of **MEM_TYPE** at PCI bus reset	**DDR Type:** Identifies the selected DDR generation of SDRAM based on the **MEM_TYPE** reset strap. 0 = DDR-II (supported speed of 400 MHz) - **MEM_TYPE** Deasserted. 1 = DDR (supported speed of 333 MHz) - **MEM_TYPE** Asserted.
01	0_2	**Data Bus Width:** Indicates the width of the data bus. See Section 8.3.3.4, “32-bit Data Bus Width” on page 458. 0 = 64 bits 1 = 32 bits
00	0_2	**DIMM Type:** Selects unbuffered or registered DIMM operating modes for the MCU. 0 = Unbuffered* 1 = Registered **NOTE:** Unbuffered DDR SDRAM memory subsystems will use the Unbuffered mode.

Table 237. DDR SDRAM Control Register 1 - SDCR1



Bit	Default	Description
31	0_2	**DQS# Disable:** Controls the behavior of the strobes as well as the configuration of the DIMM. 0 = DQS# Enabled for Differential operation, EMRS bit 10 will be programmed as zero. 1 = DQS# Disabled for Singled-ended operation, EMRS bit 10 will be programmed as one.
30:28	000_2	**RTCMD:** Read-to-Command (non-Read) turnaround period in MCLK periods. • 010 = 2 MCLK periods (DDR-I Type only) • 011 = 3 MCLK periods (DDR-II Type only) all other values reserved
27:24	0H	**WTCMD:** Write-to-Command (non-Read) turnaround period in MCLK periods. **Equation 10: WTCMD = tCAS + tWR + tREG** where tREG = 1 for registered DIMM and 0 for unbuffered DIMM, tCAS and tWR are from SPD.
23	0_2	Reserved
22:20	000_2	**RTW:** Read-to-Write turnaround period in MCLK periods. **Equation 11: RTW = tCAS + (BL/2) + tREG + (tEDP-1) = tCAS + tREG + 3** where tREG = 1 for registered DIMM and 0 for unbuffered DIMM, BL=4, tCAS is from SPD, and EDP=2. **NOTE:** a programmed value of 000_2 represents a decimal value of eight (8).
19:17	000_2	Reserved
16:12	$0\ 0000_2$	**RFC:** Refresh-to-Active and Refresh-to-Refresh period in MCLK periods **Equation 12: RFC = tRFC - 1** where tRFC is from SPD
11:09	000_2	**WR:** Write Recovery time in MCLK periods. • 000 = 0 - for DDR333 • 010 = 3 - for DDR-II 400 (encoding per JEDEC spec) all other values reserved
08:04	$0\ 0000_2$	**RC:** Active-to-Active and Active-to-Refresh period in MCLK periods. **Equation 13: RC = tRC - 1** where tRC is from SPD
03:00	0H	**WTRD:** Write-to-Read turnaround period in MCLK periods. **Equation 14: WTRD = tCAS + tWTR + tREG** where tREG = 1 for registered DIMM and 0 for unbuffered DIMM, tCAS and tWTR are from SPD.

Affected Docs: *Intel® 80331 I/O Processor Developer's Manual*

10. Power sequence timing

Problem: Section 9.1 of the *Intel® 80331 I/O Processor Design Guide* describes the power sequence requirement between VCC33 and VCC15, but does not mention any timing parameters.

Workaround: The following are the power sequencing requirements that must be followed:

1. The 80331 requires that the VCC33 voltage rail be no less than 0.5 V below VCC15 (absolute voltage value) at all times during operations, including during system power-up and power-down. In other words, the following must always be true:
   — VCC33 >= (VCC15 - 0.5 V). This can be accomplished by placing a diode (with a voltage drop < 0.5 V) between VCC15 and VCC33. The Anode is connected to VCC15 and the Cathode is connected to VCC33.
2. When a voltage regulator solution is used, which shunts VCC15 to ground while VCC33 is powered, the maximum allowable time that VCC15 can be shunted to ground while VCC33 is fully powered is 20 ms.
3. The maximum allowed time between VCC33 and VCC15 ramping is 525 ms. There is no minimum sequencing time requirement.

Affected Docs: *Intel® 80331 I/O Processor Design Guide*

11. Updated Peripheral Bus Interface (PBI) timings

Problem: Table 27 in the 80331 datasheet must be updated.

Workaround: Table 27 must have the following values:

- Tah1 – ALE high time = 15 ns minimum
- Tav1 – ALE high to address valid = 0 ns maximum
- Tah2 – ALE low to address invalid = 15 ns maximum
- Tas1 – Address valid to ALE low = 15 ns minimum
- Tao1 – ALE low to POE# low = 0 ns minimum
- Taw1 – ALE low to PWE# low = 15 ns minimum
- Tah3 – PWE# high to data invalid = 15 ns minimum
- Tas2 – Data valid to PWE# high = 6 0ns minimum
- Tac1 – ALE low to PCE[1:0]# low = 15 ns minimum

Affected Docs: *Intel® 80331 I/O Processor Datasheet*

12. ATU PM_CAP[5] text must be updated

Problem: Table 140, bit[5] reads as follows: DSI – This field is set to 0_2 meaning that this function requires a device specific initialization sequence following the transition to the D0 uninitialized state.

Workaround: It must be changed to the following: DSI – This field is set to 0_2 meaning that this function does not require a device specific initialization sequence following the transition to the D0 uninitialized state.

Affected Docs: *Intel® 80331 I/O Processor Developer's Manual*

13. WDTCR also affected by TMR1[3]

Problem: The developer's manual specifies in Section 14.4.2.4 that TMRx, bit[3], (where x = 0 or 1) enables or disables user-mode writes to the timer registers (TMRx, TCRx, TRRx). However, when TMR1[3] is set, the Watchdog timer register (WDTCR) is also disabled from user-mode writes.

Workaround: TMR1[3] also controls the Watchdog timer.

Affected Docs: *Intel® 80331 I/O Processor Developer's Manual*

14. Split Completion Message clarification

Issue: Clarification is required for bridge functionality with split completion message.

Workaround: Add this to section 2.2.13.3.3 between the second and third paragraph:

"If the split completion message indicates the occurrence of a write-data parity error (i.e. - PCI-X bridge class and write data parity error index), the bridge asserts PERR# and sets the appropriate bits in the status register when the transaction completes on the conventional interface. The 80331 bridge exhibits this behavior for both the PCI-X bridge class and completer class transactions.

For all other cases in which a non-posted write transaction completes with a split completion message, the bridge terminates the transaction on the conventional interface with target abort."

Affected Docs: *Intel® 80331 I/O Processor Developer's Manual*

15. Figure 49 in Design Guide shows incorrect trace length

Issue: Figure 49 in the 80331 Design Guide shows incorrect trace length

Workaround: Figure 49 incorrectly shows the trace length for DQ/DQS/DM as 2.0"-5.0". The correct trace length is 2.0"-8.0", as stated in Tables 58, 59, and 60.

Affected Docs: *Intel® 80331 I/O Processor Design Guide*

16. Wrong Voltage Values in Table 21

Issue: Table 21 shows wrong voltage values.

Workaround: Replaced two rows in Table 21. The two rows now appear as follows:

V_{OL1}	Output Low Voltage (DDR SDRAM)		0.35	V	I_{OL} = 12.5 mA (1, 2)
V_{OH1}	Output High Voltage (DDR SDRAM)	1.95		V	I_{OH} = -12.5 mA (1, 2)

Affected Docs: *Intel® 80331 I/O Processor Datasheet* (273943-003).

17. SBR1 Programming When Bank 1 is Unpopulated

Issue: Section 8.7.6 incorrectly states: "If bank 1 is unpopulated, SBR1[6:0] is programmed either with all zeroes or a value equal to SBR0[6:0]."

Workaround: The sentence should be changed to "If bank 1 is unpopulated, SBR1[6:0] and SBR1[31:30] should be programmed with a value equal to SBR0[6:0] and SBR0[31:30]."

Affected Docs: *Intel® 80331 I/O Processor Developer's Manual*