



Intel[®] IXP2800 Network Processor

Specification Update

February 2005

The IXP2800 Network Processor may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are documented in this specification update.

Order Number: 278756-016



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

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

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

Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the express written consent of Intel Corporation.

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

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

Intel and XScale are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2005, Intel Corporation

Contents

Revision History	4
Preface.....	5
Summary Table of Changes.....	6
Identification Information.....	13
Errata	15
Specification Changes	53
Specification Clarifications	54
Documentation Changes	55

Revision History

Date	Version	Description
February 2003	001	A0 Stepping.
March 2003	002	A1 Stepping.
April 2003	003	Corrected typo in errata table for 45, 46, 47, 49, and 50 (should have indicated Fixed in A1). Updated errata 5, 6, 7, and 56 with miscellaneous changes. Left change bars in from -002 release.
July 2003	004	A2 Stepping changes; added Errata 54 through 72 . Added Documentation Changes 13 through 17.
September 2003	005	Updated all cross-references. Updated Erratum 68 . Added Errata 73 through 77 . Added Documentation Changes 18 through 24. Revised Specification Change Table by removing previously identified specification changes that belong as Documentation Changes.
November 2003	006	Updated all cross-references. Added Errata 78 through 88 . Added Documentation Change 25. Added B0 Stepping.
January 2004	007	Updated all cross-references. Updated Erratum 86 . Added Errata 89 through 95 .
February 2004	008	Updated all cross-references. Updated Errata 69 , 89 , 90 , 91 , and 93 . Added Errata 96 and 97 . Removed Documentation Changes 22 through 25, which have all been added to the <i>IXP2800 and IXP2850 Network Processors Datasheet</i> .
March 2004	009	Added Errata 98 , 99 , 100 , 101 , and 102 .
April 2004	010	Added Errata 103 and 104 .
May 2004	011	Added information to <i>all</i> errata for B1 Stepping; added Erratum 105 . Updated Errata 92 , 93 , 94 , 97 , 100 , 102 , 103 , and 104 .
June 2004	012	Added Errata 106 , 107 , 108 , 109 , and 110 . Deleted all entries in Documentation Changes section as all have been incorporated in the documentation set. Table in Markings section has been updated with changes to production marketing numbers.
August 2004	013	Added Errata 111 , 112 , and 113 .
November 2004	014	Modified Errata 99a and 99b ; added Errata 114 .
January 2005	015	Added Errata 115 and 116 .
February 2005	016	Added Errata 117 .

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.

This document indicates the pre-silicon errata, specification changes, specification clarifications, or documentation changes that apply to the IXP2800 Network 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.

Affected Documents/Related Documents

Title	Order
<i>IXP2800 and IXP2850 Network Processors Datasheet</i>	278547
<i>IXP2800 Network Processor Hardware Reference Manual</i>	278737
<i>IXP2400 and IXP2800 Network Processor Programmer's Reference Manual</i>	278746

Nomenclature

Errata are design defects or errors. These may cause the Product Name's 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 life cycle, 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 and 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.).

Summary Table of Changes

The following table indicates the errata, specification changes, specification clarifications, or documentation changes that apply to the Product Name product. 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 component.
- Fixed: This erratum has been previously fixed.
- No Fix: There are no plans to fix this erratum.

Row

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

Errata (Sheet 1 of 5)

No.	Steppings					Page	Status	ERRATA
	A0	A1	A2	B0	B1			
1	X	X	X			14	Fixed	DRAM/SRAM Memory Reference with Indirect_ref Qualifier
2	X	X	X	X	X	14	Doc	ME_TIMESTAMP_HIGH Register
3	X	X	X			14	Fixed	Thread Summary Registers
4	X	X	X			14	Doc	Scratch Ring
5	X	X	X	X	X	14	No fix	LVDS I/O
6	X	X	X	X	X	15	No fix	QDR I/O
7	X	X	X	X	X	15	No fix	RDRAM* I/O
8	X	X	X			16	Fixed	MSF_INTERRUPT_STATUS Register
9	X	X	X			16	Fixed	RDRAM Controller Configuration Restriction
10	X	X	X			16	Fixed	MSF write64 Command with Indirect_ref Qualifier Greater than Eight
11	X	X	X	X	X	16	Doc	RDRAM Controller Programming Restriction for Three-way Interleave
12	X	X	X			17	Fixed	Simultaneous fast_wr and cap[read] to SELF_DESTRUCT_n Register
13	X	X	X			17	Fixed	Watchdog Interrupt with Watchdog History Enabled Fails to Perform Soft Reset
14	X	X	X			17	Fixed	Watchdog Timer Interrupt when the IXP2800 is Configured as Central Function
15	X	X	X			17	Fixed	SPI-4 Packet is Autopushed with Length Error when EOP is Lost Due to Overflow
16	X	X	X			18	Fixed	RDRAM_ERROR_STATUS2 Register
17	X	X	X			18	Fixed	Updating Microengine Indirect Local Memory Address CSR Also Overwrites Working Copy
18	X	X	X			18	Fixed	CAP (Enumerated CSR Addressing) Commands Modify T_INDEX and NN_GET for Certain CSR Addresses
19	X	X	X			21	Fixed	Performance Monitor Unit
20	X	X	X			21	Fixed	Writes to Slowport Interface when Intel XScale® Core is Configured in Big Endian Mode
21	X	X	X			22	Fixed	MSF Overflow Count Not Incremented Correctly
22	X	X	X			22	Fixed	PCI DMA Channel Removed
23	X	X	X			22	Fixed	Enqueue and Dequeue Operations from Intel XScale® Core to SRAM Channel 3
24	X	X	X			23	Fixed	Flushing MSF TBUF Entries
25	X	X	X			23	Fixed	Intel XScale® Core Timers Removed
26	X	X	X			23	Fixed	Intel XScale® Core UART FIFO Depth Reduced
27	X	X	X			23	Fixed	XPI Timer Control Register
28	X	X	X			24	Fixed	Microengine Initiated PCI Burst Reads

Errata (Sheet 2 of 5)

No.	Steppings					Page	Status	ERRATA
	A0	A1	A2	B0	B1			
29	X	X	X			24	Fixed	Simultaneous fast_wr and cap[read] to THD_MSG Register
30	X	X	X			24	Fixed	MSF TX Flush Bits
31	X	X	X			25	Fixed	MSF Training
32	X	X	X			25	Fixed	Dequeue Read Passes an Enqueue Write Command
33	X	X	X			25	Fixed	PCI Hold Time
34	X	X	X			25	Fixed	SRAM PUT May Pass an SRAM GET Command
35	X	X	X			26	Fixed	Atomicity of SRAM Burst Reads and Burst Writes
36	X	X	X	X	X	26	Doc	FCEFIFO Overflow
37	X	X	X			26	Fixed	MSF Dynamic Jitter Compensation
38	X	X	X			27	Fixed	SPI-4 LVDS FIFO Status
39	X	X	X			27	Fixed	UART Transmit Interrupt
40	X	X	X			27	Fixed	MSF Train Flow Control Register
41	X	X	X			27	Fixed	MSF Flow Control RX Interface Training
42	X	X	X			28	Fixed	MSF Train Flow Control Register
43	X	X	X			28	Fixed	MSF Transmit Restart
44	X	X	X	X	X	28	Doc	MSF Checksum
45	X					28	Fixed	Multiple TBUF Partitions
46	X					28	Fixed	PLL Lock/Ring Oscillator
47	X					29	Fixed	Interrupt Control Registers Cannot Set/Reset All Bits
48	X	X	X			30	Fixed	Intel XScale® Core Gasket SRAM Burst Write Followed by Local CSR Read
49	X					30	Fixed	MSF RCLK DLL Fails To Lock
50	X					30	Fixed	LVDS Receive Buffers Invert Incoming Data
51	X	X				31	Fixed	PCI Signals Being Driven During Reset when Not the Central Function
52	X	X				32	Fixed	LVDS RComp Circuits are Driven by the Receive Clock
53	X	X				32	Fixed	LVDS Static Alignment Does Not Function Properly
54	X	X				32	Fixed	Incorrect QDR and MSF Clocks
55	X	X	X			32	Fixed	PCI Memory and I/O Byte Reads from Intel XScale® Core Do Not Assert Byte-enables Correctly
56	X	X	X			33	Fixed	Master Abort on PCI Controller Causes Wrong Data on Next PCI Burst
57		X				33	Fixed	MSF, Interleaved SPI-4/CSIX, Three-Partition Mode
58	X	X				33	Fixed	PCI_PAR and PCI_AD are Not Driven Correctly when Soft Reset is Asserted
59	X	X	X			33	Fixed	PCI DMA Read FIFO Full Causes DMA to Hang
60	X	X				33	Fixed	PCI RCOMP Control

Errata (Sheet 3 of 5)

No.	Steppings					Page	Status	ERRATA
	A0	A1	A2	B0	B1			
61	X	X	X			34	Fixed	FCEFIFO Full Bit Asserts Late
62		X				34	No fix	Product_ID Register Not Updated for the A1 Stepping
63	X	X	X			34	Fixed	PCI Impedance Compensation Pin Wiring
64	X	X	X			34	Fixed	RDRAM Data Corruption
65	X	X				34	Fixed	LVDS I/O are Reset by Software Reset
66	X	X				35	Fixed	ESD Failure on VREFHI_CLK and VREFLO_CLK Input Pins
67	X	X				35	Fixed	ESD Failures on Slow Port/GPIO Port/JTAG/Serial Port Pins
68	X	X	X			35	Fixed	QDR Read Data Slips 1/2 or Full Cycle
69	X	X	X			35	Fixed	Intel XScale® Core/Microengine Initiated and Target PCI Access Anomalies
70	X	X	X			36	Fixed	MSF TCLK Duty Cycle when Sourced from TCLK_REF Pins
71	X	X	X			36	Fixed	SPI-4 DIP-4 Errors when the Clock Edge of Training Changes
72	X	X	X			36	Fixed	QDR Dequeue Performance
73			X			37	Fixed	JTAG_RST Pin Input Receiver Sensitivity to 2.5 V Supply
74	X	X	X	X	X	37	No Fix	Parity Errors Detected During RMW Operations are Written with Good Parity
75	X	X	X	X	X	37	No Fix	MSF SPI-4 Parity Error Reported on Valid Packets
76	X	X	X			38	Fixed	LVDS TSTAT Port Incorrectly Recognizes Data Patterns as Training
77	X	X	X			38	Fixed	Slowport Mode 1, Mode 3, and Mode 4 May Read Data Incorrectly
78	X	X	X			38	Fixed	FCIFIFO Errors Reported when Using LVDS TSTAT Mode
79	X	X	X			38	Fixed	PCI Drives REQ64# During Soft Reset
80	X	X	X	X	X	39	No Fix	TXCSRB Disabled in Duplex Mode
81	X	X	X			39	Fixed	Flow Control Vertical Parity Error Not Detected
82	X	X	X	X	X	39	No Fix	Training Sequence Mistaken for Type 3 CFrame
83	X	X	X	X	X	39	No Fix	MSF FCEFIFO Overflow Interrupt Does Not Increment Correctly
84	X	X	X			40	Fixed	RDRAM Read Failures in Three-channel Mode
85	X	X	X	X	X	40	No Fix	Reflect in Four Context Mode Sets Incorrect Event Signal
86	X	X	X	X	X	40	No Fix	Slowport I/O May Be Repeated Once for Every Access or Loop Indefinitely
87	X	X	X	X	X	40	No Fix	DIP-4 Error Reported when Receive Enable for SPI-4 is Disabled
88	X	X	X	X	X	41	No Fix	DIP-4 Error Reported on Receive of CIDLE After Continuous CDEAD
89				X	X	41	Doc	High Power Consumption on the 1.3 V Power Supply

Errata (Sheet 4 of 5)

No.	Steppings					Page	Status	ERRATA
	A0	A1	A2	B0	B1			
90				X	X	41	Doc	Random RDRAM Initialization Failures
91				X	X	41	No Fix	PCI DMA with Microengine PCI Read Transactions Fail Intermittently
92				X		42	Fixed	MSF Interleave TX Arbitration is Not Fair in Three-Partition Mode
93				X		42	Fixed	SRAM Controller Stall
94				X		43	Fixed	Invalid Duty Cycle on Internal Clocks
95				X	X	43	No Fix	Intel XScale® Core JTAG Debug Instructions May Be Misinterpreted
96				X	X	44	No Fix	Ingress and Egress Switch Fabric Ready Bits Not Cleared on Continuous Dead
97				X		44	Fixed	Internal Divided Clocks Have Incorrect Duty Cycle After Reset
98				X	X	44	No Fix	RDRAM I/O JTAG SAMPLE Command
99a				X	X	44	No Fix	Write Commands with Indirect Reference in Four Context Mode
99b				X	X	45	No Fix	Read Commands with Indirect Reference in Four Context Mode
100	X	X	X	X		45	Fixed	Small CFrames Should Be Followed by Dead Cycles
101	X	X	X	X	X	45	No Fix	SRAM Read Latency
102	X	X	X	X		46	Fixed	MSF Transmit Arbiter is Unfair in Multiple Partition Mode
103	X	X	X	X	X	46	No Fix	Incorrect Reading from Thermal Diode
104				X	X	46	No Fix	Incorrect Status Data on RSTAT in RSTAT Override Mode
105				X	X	47	No Fix	Extra SP_CP Pulse in Slowport Mode 1
106	X	X	X	X	X	47	No Fix	PCI Retry followed by Master Abort May Stall Outbound PCI Transactions
107				X	X	47	No Fix	MSF FCI_Full Signal Asserts Early
108	X	X	X	X	X	48	No Fix	Back-to-Back Premature SOF Error Not Reported
109	X	X	X	X	X	48	No Fix	Premature SOF on CFrames Mapped to FCEFIFO
110	X	X	X	X	X	48	Doc	QDRI 100 MHz SRAM Unsupported
111	X	X	X			48	No Fix	A Stepping Command Bus Arbitration
112				X	X	49	No Fix	B Stepping Command Bus Arbitration
113	X	X	X	X	X	49	No Fix	RDRAM Current Control Failure
114				X	X	49	No Fix	Back-to-Back CRC is not supported even if one of the CRCs is not executed due to a branching code stream

Errata (Sheet 5 of 5)

No.	Steppings					Page	Status	ERRATA
	A0	A1	A2	B0	B1			
115	X	X	X	X	X	50	No Fix	When Training is received on the RXC Bus in Duplex Mode, the FC status bits are not cleared automatically.
116				X	X	50	No Fix	Performance Monitor Unit CHAP counters will not handle underflow correctly
117				X	X	50	No Fix	When Microengine is operating at 1.4 GHz, the Performance Monitor Unit may not monitor events from the DRAM properly.

Specification Changes

No.	Steppings							Page	Status	SPECIFICATION CHANGES
	A0	A1	A2	B0	B1	#				
										None for this revision of this Specification Update.

Specification Clarifications

No.	Steppings							Page	Status	SPECIFICATION CLARIFICATIONS
	A0	A1	A2	B0	B1	#				
										None for this revision of this Specification Update.

Documentation Changes

No.	Document Revision	Page	DOCUMENTATION CHANGES
			None for this revision of this Specification Update.



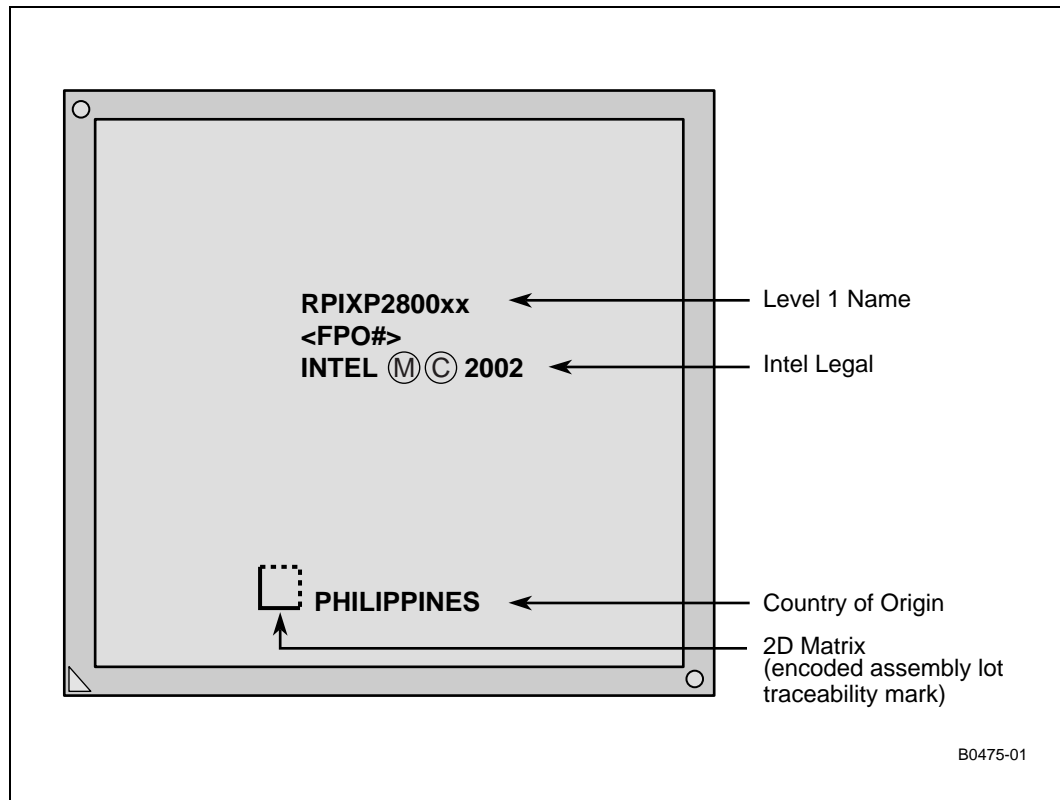
Identification Information

Markings

Product Name	Stepping	QDF Number	Marketing Part Number	Version
RPIXP2800AA	A0	Q424	MM# 848953	1.0 GHz
RPIXP2800AB	A0	Q425	MM# 848954	1.4 GHz
RPIXP2800AA	A1	Q480	MM# 851637	1.0 GHz
RPIXP2800AB	A1	Q481	MM# 851649	1.4 GHz
RPIXP2800AA	A2	Q572	MM# 855314	1.0 GHz
RPIXP2800AB	A2	Q573	MM# 855311	1.4 GHz
RPIXP2800BA	B0	Q668	MM# 857259	1.0 GHz
RPIXP2800BB	B0	Q669	MM# 857266	1.4 GHz
RPIXP2800BC	B1	Q853	MM# 862906	650 MHz
RPIXP2800BA	B1	Q808	MM# 861093	1.0 GHz
RPIXP2800BB	B1	Q809	MM# 861099	1.4 GHz
RPIXP2800BC	B1		MM# 862907 ¹	650 MHz
RPIXP2800BA	B1		MM# 862117 ¹	1.0 GHz
RPIXP2800BB	B1		MM# 855650 ¹	1.4 GHz

1. Production-qualified devices are not marked with a QDF number.

Figure 1. Package Marking



Errata

1. DRAM/SRAM Memory Reference with Indirect_ref Qualifier

Problem: Data is returned to the incorrect transfer in register when using the DRAM register on an SRAM read with an indirect reference, or the SRAM register on DRAM read with an indirect reference.

Implication: If a process attempts to perform a DRAM/SRAM read with an indirect reference, with the destination being an SRAM/DRAM transfer register respectively, the data will be erroneously returned to the DRAM/SRAM transfer in register.

Workaround: When issuing a read, using an indirect reference, you must specify the appropriate transfer register type (DRAM register for DRAM read and SRAM register for all others).

Status: Fixed

2. ME_TIMESTAMP_HIGH Register

Problem: Invalid data returned when reading ME_TIMESTAMP_HIGH CSR from the Intel XScale® core.

Implication: Reading TIMESTAMP_LOW returns valid data, however TIMESTAMP_HIGH does not.

Workaround: None, the Intel XScale® core should not perform reads or ignore data from reads to the TIMESTAMP_HIGH register. TIMESTAMP_HIGH is normally used internally to the Microengine.

Status: Doc

3. Thread Summary Registers

Problem: Need to move the THD_MSG_SUMMARY registers for future expansion.

Implication: The base addresses for the THD_MSG_SUMMARY, SELF_DESTRUCT, and INTERTHREAD_SIG registers will change in a future stepping.

Workaround: None.

Status: Fixed

4. Scratch Ring

Problem: When setting up a ring by writing to the SCRATCH_RING_BASE, SCRATCH_RING_HEAD, and SCRATCH_RING_TAIL registers, while another Microengine is operating on an unrelated ring, the head pointer of the ring being set up is corrupted.

Implication: Writes from different Microengines to these registers could conflict, resulting in the ring head pointer being corrupted.

Workaround: All scratchpad rings must be initialized (BASE, HEAD, and TAIL) from the same Microengine at start-up time before any Microengines can operate on the rings.

Status: Doc

5. LVDS I/O

Problem: LVDS I/Os are not fully JTAG compliant.

Implication: Some JTAG commands will require a workaround to obtain full functionality.

Workaround: **For A0, A1, and A2:** Provide a clock on the REF_CLK pins and strap the device into PLL bypass mode by driving the CLK_PLL_BYP pin to logic 1. *Note:* the frequency of CLK_REF_CLK does not need to be the nominal 100 MHz and can be as low as 1 MHz. Drive the CLK_NRESET pin to logic 0 for 256 REF_CLK cycles then de-assert the CLK_NRESET, drive to a logic 1, and keep CLK_REF_CLK running for the duration of the test.

For B0: The HIGHZ command does not three-state the outputs of the LVDS output pins; in general, this should not be required as this is a point-to-point bus; however the following sequence can be used to work around this issue:

Provide a clock on the CLK_REF_CLK pins and strap the device into PLL bypass mode by driving the CLK_PLL_BYP pin to logic 1. *Note:* the frequency of CLK_REF_CLK does not need to be the nominal 100 MHz and can be as low as 1 MHz. Drive the CLK_NRESET pin to logic 0 for 256 CLK_REF_CLK cycles then de-assert the CLK_NRESET, drive to a logic 1, and keep CLK_REF_CLK running for 7 to 10 additional cycles. From that point to the end of the test, leave the CLK_REF_CLK signal pair at a fixed, static level.

Note: After the workaround has been performed, the I/Os will not be in a true HIGHZ state; however the outputs will only drive ~500 uA of current, which should permit data to be driven onto the bus externally.

Additionally the CLK_NRESET signal must be driven to a 0 during the entire test, to perform the following JTAG commands for the B0 stepping:

- EXTEST
- CLAMP
- SAMPLE/PRELOAD

All other commands will function properly without the need to drive CLK_NRESET.

Status: No fix

6. QDR I/O

Problem: QDR I/Os are not fully JTAG compliant.

Implication: Some JTAG commands will require a workaround to obtain full functionality.

Workaround: **For A0 and A1:** None

For B0: The CLK_NRESET signal must be driven to a 0 during the entire test, to perform the following JTAG commands for the B0 stepping:

- EXTEST
- CLAMP
- HIGHZ
- SAMPLE/PRELOAD

All other commands will function properly without the need to drive CLK_NRESET.

Status: No fix

7. RDRAM* I/O

Problem: RDRAM I/Os are not fully JTAG compliant.

Implication: Some JTAG commands will require a workaround to obtain full functionality.

Workaround: For A0 and A1: None

For B0: The CLK_NRESET signal must be driven to a 0 during the entire test, to perform the following JTAG commands for the B0 stepping:

- EXTEST
- CLAMP
- HIGHZ
- SAMPLE/PRELOAD

All other commands will function properly without the need to drive CLK_NRESET.

Status: No fix

8. MSF_INTERRUPT_STATUS Register

Problem: The TBUF_Error bit [4], of the MSF_INTERRUPT_STATUS register is hard-wired to a logic 0 level.

Implication: The TBUF_Error interrupt cannot be generated.

Workaround: None

Status: Fixed

9. RDRAM Controller Configuration Restriction

Problem: When employing a three-way interleave, the Rambus* controller can only support configurations in which the total number of devices per channel is 1, 2, 4, 8, 16 or 32 devices.

Implication: The controller will not be able to access all of the populated memory in configurations with the number of devices different than those listed.

Workaround: None

Status: Fixed

10. MSF write64 Command with Indirect_ref Qualifier Greater than Eight

Problem: The MSF write64 command, when used with an indirect reference count that is greater than eight, will only transfer eight 8-byte words, not the number specified in the reference count. This restriction does not apply to the read64 command.

Implication: The MSF write64 command cannot transfer more than eight 8-byte words.

Workaround: For MSF write64 transfers with a reference count that is greater than eight, two commands are required to transfer the data.

Status: Fixed

11. RDRAM Controller Programming Restriction for Three-way Interleave

Problem: In extreme conditions in which the DRAM controller is being flooded with multiple 32-DWORD burst reads *and* the RD_BP_DIS bit [16] of the RDRAM_CHAN_CONTROL register is set, (which disables backpressure on READs), *and* all three DRAM channels are being employed – the DRAM controller may not deliver all of the requested data during a burst read transaction.

Implication: The DRAM controller will erroneously signal the Microengine that issued the request, indicating that all of the data has been transferred, which will result in data corruption.

Workaround: Implementations that utilize all three RDRAM channels should never disable backpressure on *reads*. This is done by setting the RD_BP_DIS bit [16], of the RDRAM_CHAN_CONTROL register. The default value for this register is for backpressure to be enabled.

Status: Doc

12. Simultaneous fast_wr and cap[read] to SELF_DESTRUCT_n Register

Problem: A simultaneous fast_wr and cap[read] to the SELF_DESTRUCT_n register can result in the write data being cleared before the data has actually been read. This is possible as the fast_wr has a separate path from cap[read/write] commands and hence both can happen on the same clock cycle.

Implication: The following read of the register will not get the intended data.

Workaround: Use the cap[write] command in place of the fast_wr command, as this command cannot happen simultaneously with a cap[read] command.

Status: Fixed

13. Watchdog Interrupt with Watchdog History Enabled Fails to Perform Soft Reset

Problem: If the watchdog history bit is set (WATCHDOG_HISTORY register, WD_HIS bit [0]) indicating that a watchdog timer has expired and the Watchdog timer reset enable bit is disabled, (IXP_RESETO register bit [24]) then when the watchdog timer expires for a second time, a soft reset should be initiated. This does not occur.

Implication: Cannot generate a soft reset using this mechanism.

Workaround: Always set the watchdog time reset enable bit (IXP_RESETO register bit [24]) to generate a watchdog timer initiated reset.

Status: Fixed

14. Watchdog Timer Interrupt when the IXP2800 is Configured as Central Function

Problem: If the IXP2800 is configured to be the central function, i.e., CFG_RST_DIR and CFG_PCI_BOOT_HOST straps set to 1, with the Watchdog timer reset enable bit disabled (IXP_RESETO register bit [24]), and the watchdog timer expires, the WDI bit in PCI_OUT_INT_STATUS will incorrectly be set. The correct behavior would be to set the watchdog history bit (WD_HIS bit [0]) of the WATCHDOG_HISTORY register.

Implication: The WDI bit will be set in the PCI_OUT_INT_STATUS when the WD_HIS bit should have been set.

Workaround: Always set the watchdog time reset enable bit (IXP_RESETO register bit [24]) to generate a watchdog timer initiated reset.

Status: Fixed

15. SPI-4 Packet is Autopushed with Length Error when EOP is Lost Due to Overflow

Problem: While receiving a packet, if the FIFO in the Media Switch Fabric (MSF) interface overflows and loses the EOP of the packet currently being received, this packet may be autopushed to the Microengines and marked with a length error.

Implication: When an overflow occurs, the EOP value returned in the status for the packet that preceded the overflow is not guaranteed, and the packet should be discarded.

Workaround: Software must discard the packet that precedes the overflow condition.

Status: Fixed

16. RDRAM_ERROR_STATUS2 Register

Problem: The functionality for the MULT_UNCORR_ERR REGISTER (RMW_ERROR bit [30] and RDRAM_ERROR_STATUS2 bit [29]) has not been implemented.

Implication: These registers will not be set for these events.

Workaround: None

Status: Fixed

17. Updating Microengine Indirect Local Memory Address CSR Also Overwrites Working Copy

Problem: When a local_csr_write instruction is executed to the INDIRECT_LM_ADDR_0 or INDIRECT_LM_ADDR_1 register, both the indirect local memory address register *and* the current active local memory address register will be modified.

The registers affected by this errata item are:

```
ACTIVE_LM_ADDR_0
ACTIVE_LM_ADDR_1
INDIRECT_LM_ADDR_0
INDIRECT_LM_ADDR_1
ACTIVE_LM_ADDR_0_BYTE_INDEX
ACTIVE_LM_ADDR_1_BYTE_INDEX
INDIRECT_LM_ADDR_0_BYTE_INDEX
INDIRECT_LM_ADDR_1_BYTE_INDEX
```

Implication: The active local memory register will be corrupted with the data written to the indirect local memory register.

Workaround: During initialization, the ACTIVE_LM_ADDR_* register should be written last to avoid corruption. Additionally, whenever a write to an INDIRECT_LM_ADDR_* register listed is executed, the value of the ACTIVE_LM_ADDR_* register must be stored and re-written.

Status: Fixed

18. CAP (Enumerated CSR Addressing) Commands Modify T_INDEX and NN_GET for Certain CSR Addresses

Problem: When performing a CAP-enumerated address operation, if the enumerated address is one of the following CSR register addresses, it also causes the T_INDEX and NN_GET pointers to be updated:

The registers/addresses affected are:

```
THD_MSG_CLR_0_2_1 = 0x 4144
THD_MSG_CLR_0_2_3 = 0x 414C
THD_MSG_CLR_1_0_0 = 0x 4200
THD_MSG_CLR_1_0_1 = 0x 4204
THD_MSG_CLR_1_0_2 = 0x 4208
THD_MSG_CLR_1_0_3 = 0x 420C
THD_MSG_CLR_1_0_4 = 0x 4210
THD_MSG_CLR_1_0_5 = 0x 4214
THD_MSG_CLR_1_0_6 = 0x 4218
THD_MSG_CLR_1_0_7 = 0x 421C
```

THD_MSG_CLR_1_1_0 = 0x 4220
 THD_MSG_CLR_1_1_1 = 0x 4224
 THD_MSG_CLR_1_1_2 = 0x 4228
 THD_MSG_CLR_1_1_3 = 0x 422C
 THD_MSG_CLR_1_1_4 = 0x 4230
 THD_MSG_CLR_1_1_5 = 0x 4234
 THD_MSG_CLR_1_1_6 = 0x 4238
 THD_MSG_CLR_1_1_7 = 0x 423C
 THD_MSG_CLR_1_2_0 = 0x 4240
 THD_MSG_CLR_1_2_1 = 0x 4244
 THD_MSG_CLR_1_2_2 = 0x 4248
 THD_MSG_CLR_1_2_3 = 0x 424C
 THD_MSG_CLR_1_2_4 = 0x 4250
 THD_MSG_CLR_1_2_5 = 0x 4254
 THD_MSG_CLR_1_2_6 = 0x 4258
 THD_MSG_CLR_1_2_7 = 0x 425C
 THD_MSG_CLR_1_3_0 = 0x 4260
 THD_MSG_CLR_1_3_1 = 0x 4264
 THD_MSG_CLR_1_3_2 = 0x 4268
 THD_MSG_CLR_1_3_3 = 0x 426C
 THD_MSG_CLR_1_3_4 = 0x 4270
 THD_MSG_CLR_1_3_5 = 0x 4274
 THD_MSG_CLR_1_3_6 = 0x 4278
 THD_MSG_CLR_1_3_7 = 0x 427C
 THD_MSG_CLR_1_4_0 = 0x 4280
 THD_MSG_CLR_1_4_1 = 0x 4284
 THD_MSG_CLR_1_4_2 = 0x 4288
 THD_MSG_CLR_1_4_3 = 0x 428C
 THD_MSG_CLR_1_4_4 = 0x 4290
 THD_MSG_CLR_1_4_5 = 0x 4294
 THD_MSG_CLR_1_4_6 = 0x 4298
 THD_MSG_CLR_1_4_7 = 0x 429C
 THD_MSG_CLR_1_5_0 = 0x 42A0
 THD_MSG_CLR_1_5_1 = 0x 42A4
 THD_MSG_CLR_1_5_2 = 0x 42A8
 THD_MSG_CLR_1_5_3 = 0x 42AC
 THD_MSG_CLR_1_5_4 = 0x 42B0
 THD_MSG_CLR_1_5_5 = 0x 42B4
 THD_MSG_CLR_1_5_6 = 0x 42B8
 THD_MSG_CLR_1_5_7 = 0x 42BC
 THD_MSG_CLR_1_6_0 = 0x 42C0
 THD_MSG_CLR_1_6_1 = 0x 42C4
 THD_MSG_CLR_1_6_2 = 0x 42C8
 THD_MSG_CLR_1_6_3 = 0x 42CC
 THD_MSG_CLR_1_6_4 = 0x 42D0
 THD_MSG_CLR_1_6_5 = 0x 42D4
 THD_MSG_CLR_1_6_6 = 0x 42D8
 THD_MSG_CLR_1_6_7 = 0x 42DC
 THD_MSG_CLR_1_7_0 = 0x 42E0
 THD_MSG_CLR_1_7_1 = 0x 42E4
 THD_MSG_CLR_1_7_2 = 0x 42E8
 THD_MSG_CLR_1_7_3 = 0x 42EC
 THD_MSG_CLR_1_7_4 = 0x 42F0
 THD_MSG_CLR_1_7_5 = 0x 42F4
 THD_MSG_CLR_1_7_6 = 0x 42F8

THD_MSG_CLR_1_7_7 = 0x 42FC
THD_MSG_0_2_1 = 0x 4544
THD_MSG_0_2_3 = 0x 454C
THD_MSG_1_0_0 = 0x 4600
THD_MSG_1_0_1 = 0x 4604
THD_MSG_1_0_2 = 0x 4608
THD_MSG_1_0_3 = 0x 460C
THD_MSG_1_0_4 = 0x 4610
THD_MSG_1_0_5 = 0x 4614
THD_MSG_1_0_6 = 0x 4618
THD_MSG_1_0_7 = 0x 461C
THD_MSG_1_1_0 = 0x 4620
THD_MSG_1_1_1 = 0x 4624
THD_MSG_1_1_2 = 0x 4628
THD_MSG_1_1_3 = 0x 462C
THD_MSG_1_1_4 = 0x 4630
THD_MSG_1_1_5 = 0x 4634
THD_MSG_1_1_6 = 0x 4638
THD_MSG_1_1_7 = 0x 463C
THD_MSG_1_2_0 = 0x 4640
THD_MSG_1_2_1 = 0x 4644
THD_MSG_1_2_2 = 0x 4648
THD_MSG_1_2_3 = 0x 464C
THD_MSG_1_2_4 = 0x 4650
THD_MSG_1_2_5 = 0x 4654
THD_MSG_1_2_6 = 0x 4658
THD_MSG_1_2_7 = 0x 465C
THD_MSG_1_3_0 = 0x 4660
THD_MSG_1_3_1 = 0x 4664
THD_MSG_1_3_2 = 0x 4668
THD_MSG_1_3_3 = 0x 466C
THD_MSG_1_3_4 = 0x 4670
THD_MSG_1_3_5 = 0x 4674
THD_MSG_1_3_6 = 0x 4678
THD_MSG_1_3_7 = 0x 467C
THD_MSG_1_4_0 = 0x 4680
THD_MSG_1_4_1 = 0x 4684
THD_MSG_1_4_2 = 0x 4688
THD_MSG_1_4_3 = 0x 468C
THD_MSG_1_4_4 = 0x 4690
THD_MSG_1_4_5 = 0x 4694
THD_MSG_1_4_6 = 0x 4698
THD_MSG_1_4_7 = 0x 469C
THD_MSG_1_5_0 = 0x 46A0
THD_MSG_1_5_1 = 0x 46A4
THD_MSG_1_5_2 = 0x 46A8
THD_MSG_1_5_3 = 0x 46AC
THD_MSG_1_5_4 = 0x 46B0
THD_MSG_1_5_5 = 0x 46B4
THD_MSG_1_5_6 = 0x 46B8
THD_MSG_1_5_7 = 0x 46BC
THD_MSG_1_6_0 = 0x 46C0
THD_MSG_1_6_1 = 0x 46C4
THD_MSG_1_6_2 = 0x 46C8
THD_MSG_1_6_3 = 0x 46CC

```

THD_MSG_1_6_4 = 0x 46D0
THD_MSG_1_6_5 = 0x 46D4
THD_MSG_1_6_6 = 0x 46D8
THD_MSG_1_6_7 = 0x 46DC
THD_MSG_1_7_0 = 0x 46E0
THD_MSG_1_7_1 = 0x 46E4
THD_MSG_1_7_2 = 0x 46E8
THD_MSG_1_7_3 = 0x 46EC
THD_MSG_1_7_4 = 0x 46F0
THD_MSG_1_7_5 = 0x 46F4
THD_MSG_1_7_6 = 0x 46F8
THD_MSG_1_7_7 = 0x 46FC
SCRATCH_RING_HEAD_4 = 0x 4844
PRODUCT_ID = 0x 4A00
MISC_CONTROL = 0x 4A04
IXP_RESET_0 = 0x 4A0C
IXP_RESET_1 = 0x 4A10
CLOCK_CONTROL = 0x 4A14
STRAP_OPTIONS = 0x 4A18
    
```

Implication: Whenever one of these registers is accessed using enumerated addressing, the T_INDEX and NN_GET pointers will be incorrectly updated.

Workaround: Use CAP-calculated addressing instruction in place of enumerated addressing i.e.,

```

immed [csr_reg, 0x4A18, <<0]; //Move strap_options address to csr_reg
cap [read, $readdata, csr_base, csr_reg, 1]; //read strap options register
    
```

Status: Fixed

19. Performance Monitor Unit

Problem: The Performance Monitor Unit (PMU) functionality will not be available in the A0 stepping of the IXP2800.

Implication: Performance monitor commands cannot be executed and will result in undefined behavior.

Workaround: None

Status: Fixed

20. Writes to Slowport Interface when Intel XScale® Core is Configured in Big Endian Mode

Problem: Writes to the Slowport PROM address range when the Intel XScale® core is configured in Big Endian format and the Slowport interface is in 8-bit or 16-bit mode, will result in the lower two bits of the address generated on the Slowport interface being flipped as described below:

8-Bit Address Case	
Lower address bits from Intel XScale® Core	Slowport Address
00	11

8-Bit Address Case	
Lower address bits from Intel XScale® Core	Slowport Address
01	10
10	01
11	00

16-Bit Address Case	
Lower address bits from Intel XScale® Core	Slowport Address
1X	0X
0X	1X

Implication: This issue does not affect read commands. For write commands, the incorrect address is generated on the Slowport interface.

Workaround: The lower two bits of the address need to be inverted per the tables above, to get the desired address. For example, a write to address 0x10 would be programmed as 0x13 – i.e., flip the two least significant bits. The resulting address generated on the Slowport interface would be the desired 0x10.

Status: Fixed

21. MSF Overflow Count Not Incremented Correctly

Problem: The MSF overflow count is not incremented correctly when an overflow occurs on an SPI-4 control word with both the EOP and SOP bits set. In this case, the hardware correctly discards both packets. However, the overflow counter is only incremented by 1 instead of 2.

Implication: On an overflow with both EOP and SOP set in the control word, the overflow count will not be incremented correctly.

Workaround: None

Status: Fixed

22. PCI DMA Channel Removed

Problem: The number of DMA channels for the A0 stepping has been reduced to 1.

Implication: Only one of the two DMA channels will be available in the A0 stepping.

Workaround: None

Status: Fixed

23. Enqueue and Dequeue Operations from Intel XScale® Core to SRAM Channel 3

Problem: Enqueue and Dequeue instructions executed from the Intel XScale® core to SRAM channel 3 queue array controller can corrupt other queue array entries with incorrect enqueue data.

Implication: Enqueue and Dequeue operations from the Intel XScale® core to SRAM channel 3 can corrupt other queue array entries.

Workaround: The Microengines must perform enqueue and dequeue operations to the channel 3 SRAM controller.

Status: Fixed

24. Flushing MSF TBUF Entries

Problem: Under certain conditions when using the TX_FLUSH_PAR n register bits to flush valid entries from the TBUF, the valid bits are not cleared. This happens when the flush operation occurs on the same clock cycle as an element that is finished being transmitted and the hardware is clearing the valid bit for that element.

Implication: The valid bits for the entries are not cleared, which could result in the TBUF elements being transmitted when they should have been flushed.

Workaround: The TX_FLUSH_PAR n bits should be written multiple times to ensure that the valid bits are cleared.

Status: Fixed

25. Intel XScale® Core Timers Removed

Problem: Intel XScale® core timers #2 and #3 have been removed for the A0 stepping due to area constraints within the device. Intel XScale® core timers #1 and #4 (watchdog timer) will remain in the device.

Implication: Only two of the four Intel XScale® core timers will be available. Register accesses to timers #2 and #3 should not be performed and will result in undefined behavior.

Workaround: Use timers #1 and #4.

Status: Fixed

26. Intel XScale® Core UART FIFO Depth Reduced

Problem: The Intel XScale® core UART FIFO depth for receive has been reduced from 64 to 32 entries and from 64 to 16 entries for transmit, due to area constraints within the device. Additionally, the Interrupt Trigger Level control bits in the UART FIFO CONTROL register, UART_FCR, have been modified such that a value of 0x3 or 2'b11 will result in an interrupt being generated when there are 24 bytes of data in the receive FIFO (rather than a value of 32 bytes as documented in the *IXP2400/IXP2800 Network Processor Programmers Reference Manual*).

Implication: Programmers should to be aware of the change to the Interrupt Trigger Level control register described above and change the code if required; otherwise, the change should be transparent to the user.

Workaround: None

Status: Fixed

27. XPI Timer Control Register

Problem: The XPI timer control registers can be un-intentionally overwritten during a write to any of the timer read-only registers.

Implication: The timer control register can un-intentionally be corrupted during a write to a read-only timer register.

Workaround: Do not perform writes to any of the read-only timer registers. Specifically, READ ONLY register bits within a register that contains READ ONLY, WRITE ONLY, and READ WRITE register bits,

should always be written with 0. Additionally, no write commands should be performed to the T#_CSR registers as these are all READ ONLY.

Status: Fixed

28. Microengine Initiated PCI Burst Reads

Problem: The PCI controller re-connects with the incorrect address for a Microengine-initiated burst read when the transfer crosses a 64-byte boundary *and* the requested byte count results in less than eight bytes being transferred on the final transfer of the burst.

For example, a burst of 60 bytes starting at address 0x1D0 would be broken into three transfers by the controller before being sent to the PCI FIFO such as:

1. Starting at address 0x1D0 for 48 bytes. // At this point, the PCI address crosses the 64-byte boundary and the controller will disconnect.
2. Starting at address 0x200 for 8 bytes.
3. Starting at address 0x208 for 4 bytes. // For this last transfer, the controller will reconnect with address 0x200 instead of 0x208.

Implication: The PCI read data would be corrupted as a result of re-connecting with the incorrect address.

Workaround: For Microengine-initiated burst read transfers to the PCI, the burst sizes should be limited to multiples of 64-bit (quadword) sizes, and the starting address should be limited to a 64-bit (quadword) boundary with bits[2:0] being zeros. This guarantees that the final transfer is always eight bytes.

Status: Fixed

29. Simultaneous fast_wr and cap[read] to THD_MSG Register

Problem: A simultaneous fast_wr and cap[read] to the THD_MSG (generic address) register can result in the read data that is returned being corrupted. This is possible as the fast_wr has a separate path from cap[read/write] commands and hence both can happen on the same clock cycle.

Implication: The read data will be corrupted, as it will come from the address of the register that is being written in the fast_wr command.

Workaround: Use the cap[read] command and the actual thread-specific "THD_MSG_#_\$_&" register addresses to read these registers instead of the generic address. Refer to the *IXP2400/IXP2800 Network Processor Programmer's Reference Manual* for more information about the THD_MSG registers.

Note:

= Microengine Cluster Number, 0 to 1
 \$ = Microengine Number in Cluster, 0 to 7
 & = Thread Number, 0 to 7

i.e., cap[read, \$xfer0, THD_MSG_0_1_2] //read THD_MSG register for ME1, Thread 2

Status: Fixed

30. MSF TX Flush Bits

Problem: After writing a 1 to MSF_TX_CONTROL[TX_FLUSH_PARn] bits which are write-only, the MSF_TX_CONTROL register will always read back with the corresponding bit asserted. This does not affect the operation of the flush.

Implication: After writing a 1 to the TX_FLUSH_PARn bits, the MSF_TX_CONTROL register will read back with the corresponding bit asserted.

Workaround: The user should re-write the MSF_TX_CONTROL register with TX_FLUSH_PARn = 0 after performing a flush operation.

Status: Fixed

31. MSF Training

Problem: Any write to the MSF_TX_CONTROL register following a write of 1 to MSF_TRAIN_DATA[SING_TRAIN] register results in two training sequences being sent on the TDAT bus. Additional writes to the MSF_TX_CONTROL register results in additional training sequences being transmitted on the data bus.

Implication: Unwanted training sequences will be transmitted on the data bus.

Workaround: To avoid sending unwanted training sequences, re-write TRAIN_DATA with SING_TRAIN = 0.

Status: Fixed

32. Dequeue Read Passes an Enqueue Write Command

Problem: In certain conditions, a queue array dequeue read may pass an enqueue write command, which will cause the head pointer of the queue to be corrupted.

Implication: The queue array head pointer will be corrupted.

Workaround: There are two possible workarounds for this issue:

1. Disable “enqueue_performance_mode” by setting bit 15 in the SRAM_CONTROL register. In this mode, enqueues are serialized behind dequeues to the same queue, thus there is a performance penalty (on the order of 15%) on back-to-back references to the same queue.

2. Do not perform generic SRAM writes with a burst length greater than 2, atomic, ring PUT commands with a burst length greater than 4, or JOURNAL commands with a burst length greater than 4 to channels that are doing queue array operations. This ensures that the enqueue write is able to win arbitration to the write pin interface before the dequeue read is issued.

Status: Fixed

33. PCI Hold Time

Problem: The PCI specification states that PCI signal hold time is 0 ns. The IXP2800 requires the PCI signals to have a hold time of ~1.0 ns to latch the data correctly.

Implication: PCI Specification violation. System designer must manage clock/data skew more carefully to ensure sufficient hold time.

Workaround: None

Status: Fixed

34. SRAM PUT May Pass an SRAM GET Command

Problem: If an SRAM ring is full or close to full *and* a GET command is started and subsequently stalled, then a PUT command(s) can pass the GET and overwrite the ring before the read is complete.

Implication: The ring will be overwritten and corrupt the data.

Workaround: There are two possible workarounds for this issue:

1. PUT and GET commands to a given SRAM ring must not use a burst count greater than five.

2. Software must keep track of the number of items on an SRAM ring and inhibit further PUT commands when the ring is close to full (64 entries). *Note:* the number of current 4-byte words on the ring is returned in the status word for each PUT command.

Status: Fixed

35. Atomicity of SRAM Burst Reads and Burst Writes

Problem: The A0 design does not guarantee that SRAM burst operations are atomic. For example, if Microcode issues reads and writes to a data structure with a burst length of 2, the two DWORDs returned from the read operations can be one of the following:

1. Two DWORDs of "new" (post-write) data
2. Two DWORDs of "old" (pre-write) data
3. One DWORD of "old" data and one DWORD of "new" data

Implication: Unexpected data may be returned on SRAM burst read operations.

Workaround: While it is highly unlikely that case #3 above could occur, software should employ one or more of the following strategies in cases where SRAM bursts must be atomic:

1. Ensure that the data is never read while it is being written by waiting at least ~500 cycles after the write-complete signal before issuing the read command.
2. Use a valid bit on each DWORD of the data. The reader needs to check the valid bit on the first and last DWORDs of the read and re-read the data if necessary.
3. Use a semaphore to inhibit multiple accesses in flight to the same address.

Status: Fixed

36. FCEFIFO Overflow

Problem: In Simplex mode, if the application software erroneously overflows the FCEFIFO, an incorrect packet with correct parity will be transmitted on the Flow Control Bus.

Implication: If the FCEFIFO does overflow, a bad packet with valid parity will be transmitted on the Flow Control Bus.

Workaround: Software must ensure that the FCEFIFO does not overflow by programming the High Water Mark (HWM) and testing whether the FCEFIFO is full prior to writing any data.

Status: Doc

37. MSF Dynamic Jitter Compensation

Problem: The MSF dynamic jitter compensation functionality is not available in the A0 stepping of the IXP2800.

Implication: The dynamic jitter compensation function should not be enabled and will result in undefined behavior.

Note: This errata item does not affect training sequences for data path deskew.

Workaround: None.

Status: Fixed

38. SPI-4 LVDS FIFO Status

Problem: When using the LVDS I/O for SPI-4 flow control, if the same port address is programmed in consecutive timeslots as an even/odd number in the calendar sequence and the status value changes, then the first value is written into the status RAM instead of the second value. The second value is the more recent status for the port.

Note: This item does not apply when LVTTL I/O is used for SPI-4 flow control.

Implication: The incorrect status is captured for the port address.

Workaround: Initialization software should not program the same port address in consecutive timeslots in the calendar.

Status: Fixed

39. UART Transmit Interrupt

Problem: The UART transmit interrupt (UART_IER register bit [1]), which indicates that the transmit FIFO is empty, may falsely assert when the FIFO is not empty.

Implication: An interrupt is generated to the Intel XScale® core indicating that the FIFO is empty when it is not.

Workaround: Software should poll the Transmit Empty bit (UART Line Status Register, UART_LSR bit [6]), to determine that the FIFO is empty in place of the transmit interrupt.

Status: Fixed

40. MSF Train Flow Control Register

Problem: The ALPHA value of the Train_Flow_Control register is incorrectly mapped to bits [16:11] of the register instead of the desired [15:10].

Implication: The ALPHA count will be shifted by one bit. Additionally bit 16 is also the FORCE_FCIDLE register, and will be used as the most significant bit of the ALPHA count.

Workaround: Software should use bits [15:11] of the Train_Flow_Control register for the ALPHA count. Software should only write bit [16] to enable FORCE_FCIDLE. This bit should be cleared after the force idle operation has completed.

Status: Fixed

41. MSF Flow Control RX Interface Training

Problem: When using training sequences on the flow control receive interface for data path deskew, a parity error may be generated upon receiving a valid packet if the packet immediately follows a single training sequence; i.e., there is no IDLE control word between the end of the training sequence and the start of the packet.

Implication: A parity error will be incorrectly generated, although the packet received was valid.

Workaround: For system configurations that employ a DUPLEX topology, when the flow control bus is connected between two IXP2800 devices, no workaround is required, as the hardware will always insert an IDLE control word after training. For SIMPLEX mode the ALPHA value of the transmitting device should be set to 2, which will result in two back-to-back training sequences to be initiated on the bus; this prevents the parity error from occurring. Only an alpha value of 2 is supported in the A0 stepping of the device for SIMPLEX mode.

Status: Fixed

42. MSF Train Flow Control Register

Problem: Setting an ALPHA count in the TRAIN_FLOW_CONTROL register to a count value greater than 1, results in training sequences initiated on the flow control TXCDAT, to run indefinitely.

Implication: When the ALPHA count is set to a value greater than 1, training sequences continue indefinitely. Training sequences initiated via the CONT_TRAIN, SING_TRAIN, and TD_EN_CDEAD registers are all affected by this item.

Workaround: The ALPHA value of the TRAIN_FLOW_CONTROL register must be set to 1.

Status: Fixed

43. MSF Transmit Restart

Problem: If the transmit section is restarted by disabling/enabling the CSIX Control and Data or SPI-4 via the TX_EN_CC, TX_EN_CD, or TX_EN_S register bits within the MSF_TX_CONTROL register, upon re-enabling the Control and Data or SPI-4 sections, no further packets will be transmitted by the MSF unit. Additionally, the EMPTY bit in the TX_SEQUENCE_N registers will not assert after the restart operation.

Implication: If the MSF transmit section is restarted by disabling/enabling the TX_EN_CC, TX_EN_CD, or TX_EN_S registers, no further packets are transmitted after the transmit section has been re-enabled.

Workaround: After the TX section has been enabled, software should never restart the transmit section via the TX_EN_CC, TX_EN_CD, or TX_EN_S registers.

Status: Fixed

44. MSF Checksum

Problem: The checksum that is calculated by the MSF unit includes the final 1's complement as defined in RFC793. This is different from the IXP2400, which does not include the final 1's complement.

Implication: For packets spanning multiple RBUF entries, software must take the 1's complement for each mpacket checksum that is part of the packet, sum them all together, and then take the final 1's complement of the computed checksum.

Workaround: None

Status: Doc

45. Multiple TBUF Partitions

Problem: When using multiple TBUF partitions, certain prepend and payload offset values may cause an extra DWORD of data to be written into one of the transmit FIFOs. This results in the packet or CFrame that is being transmitted, to be corrupted.

Implication: When using the multiple TBUF partitions, transmitted CSIX packets may be corrupted.

Workaround: For two-partition CSIX mode, only use the data partition when using variable prepend and payload offsets and lengths. When using multiple partitions, the prepend length, prepend offset, payload length, and payload offset must end on DWORD boundaries.

Workaround: Do not use multiple TBUF partitions.

Status: Fixed

46. PLL Lock/Ring Oscillator

Problem: The enable for the Ring Oscillator has the incorrect polarity and does not start up after reset. Without the ring oscillator running, there is no clock during reset and the control logic and clock divider is in an uninitialized state that never recognizes a PLL lock. Additionally, the enable for the

feedback clock from the ring oscillator to the PLL also has the incorrect polarity, which causes the PLL to disable its clock.

Implication: The internal PLL clock will not be functional.

Workaround: To provide a clock, the device must be operated in PLL Bypass mode in which the external clock inputs, REF_CLK pins, are used in place of the PLL clock. This can be accomplished by using either an LVDS Oscillator or a signal generator. *Note:* in this mode, the operating frequency of the device is significantly less than when using the PLL because it is a function of the input REF_CLK. For example, when using a 500 MHz REF_CLK input, this is then divided by 2 internally to derive the Microengine clock, which in this case would be 250 MHz. All of the external interface clocks would then be derived by the programmed integer divide of the Microengine clock frequency as programmed in the CLOCK_CONTROL CSR. Also note that the lowest supported divisor is 3, which sets the maximum external interface clocks at 83.33 MHz.

Status: Fixed

47. Interrupt Control Registers Cannot Set/Reset All Bits

Problem: All of the supported interrupts in the following CSRs cannot be enabled and will retain their reset states. The breakdown of affected CSRs and register bits is:

FIQ_ENABLE_SET

Passing: 26,25,24,17,16,15,14,9,8,7,6,4,0

Fail WRITE: 31,29,27,19,18,13,12,11,10,5,3

Fail READ : 30,28,23,22,21,20,2,1

IRQ_ENABLE_SET

Passing: 27,19,17,16,15,14,12,11,8,0

Fail WRITE: 30,24,23,20,18,13,10,9,7,6,4

Fail READ : 31,29,28,26,25,22,21,5,3,2,1

FIQ_THD_ENABLE_A_0 2,5,6,8,10,11,18,21,22,25

FIQ_THD_ENABLE_A_1 1,4,7,8,10,11,13,15,16,17,21,28,31

FIQ_THD_ENABLE_A_2 1,2,5,10,11,13,14,17,19,24,25,26

FIQ_THD_ENABLE_A_3 0,2,5,6,10,11,15,16,17,20,22,23,24,26,30,31

FIQ_THD_ENABLE_B_0 0,3,6,10,11,18,16,17,19,20,23,24,25,28,29,30,31

FIQ_THD_ENABLE_B_1 2,5,7,9,10,11,18,19,23,25,26,29,30,31

FIQ_THD_ENABLE_B_2 0,1,6,8,11,13,15,18,19,20,24,25,27,28

FIQ_THD_ENABLE_B_3 2,5,7,8,16,17,20,22,24,28,29,31

IRQ_THD_ENABLE_A_0 0,1,3,5,6,7,9,10,12,13,14,20,25,28,

IRQ_THD_ENABLE_A_1 5,6,13,15,22,23,27,

IRQ_THD_ENABLE_A_2 0,4,11,16,17,19,20,24,26,27,28,30

IRQ_THD_ENABLE_A_3 0,7,8,11,13,16,17,22,27,29

IRQ_THD_ENABLE_B_0 3,11,16,18,22,27,31

IRQ_THD_ENABLE_B_1 4,5,8,14,15,17,18,20,21,23,25,29,30,31

IRQ_THD_ENABLE_B_2 2,4,5,9,11,12,13,14,15,24,29,31

IRQ_THD_ENABLE_B_3 2,4,5,6,7,18,19,23,24,25,26,29

FIQ_ERR_ENABLE 1,6,9,12,15,20,31

IRQ_ERR_ENABLE 2,14,17,18,22,29,30

FIQ_ATTN_ENABLE 1,3,6,7,9,10,11,13,14

IRQ_ATTN_ENABLE 1,3,5,7,8,15

SOFT_INT 0

BRK_ENABLE_SET 1,10,11,12,13

Implication: The register bits listed above cannot be enabled to generate an interrupt or breakpoint.

Workaround: None

Status: Fixed

48. Intel XScale® Core Gasket SRAM Burst Write Followed by Local CSR Read

Problem: An Intel XScale® core burst write to SRAM followed by an Intel XScale® core local CSR read, results in incorrect data being written to SRAM. In addition, any following IO command results in undefined behavior.

Implication: After this condition occurs, any following IO command from the Intel XScale® core gasket results in undefined behavior.

Workaround: Software should configure the Intel XScale® core to use only write-through caching, which is controlled via the “C”, “B”, and “X” bits of the page table descriptors. Write Buffer Coalescing must also be disabled by setting the “Write Buffer Coalescing Disable” (bit 0) in the CP15 Auxiliary Control Register. This prevents any multi-word stores from being issued to the Intel XScale® core gasket. Refer to *Intel XScale® Technology Developers Manual* for more information about the CP15 registers. See the example code below, for disabling coalescing:

```
@#####
@#Disable coalescing

    mrc p15, 0, r0, c1, c0, 1
    orr r0, r0, #1
    mcr p15, 0, r0, c1, c0, 1
@# Synch
    mrc p15, 0, r0, c2, c0, 0
    mov r0, r0
    sub pc, pc, #4
@#####
```

Status: Fixed

49. MSF RCLK DLL Fails To Lock

Problem: The MSF RCLK DLL does not lock reliably.

Implication: To obtain the lock reliably, the receive clock must be present prior to the de-assertion of CLK_NRESET.

Workaround: None

Status: Fixed

50. LVDS Receive Buffers Invert Incoming Data

Problem: The LVDS interfaces are inverting the receive data on all LVDS input receive buffers (except the CLK_REF_CLK inputs).

Implication: The data received by the LVDS input buffers will be inverted.

Workaround: The problem can be worked around on A0 silicon by swapping the _H and _L inputs on the PCB, or by inverting the incoming data. The following signals need to be inverted:

- FC_RXCCLK
- FC_RXCCLK_L
- FC_TXCFC
- FC_TXCFC_L
- FC_RXCPAR
- FC_RXCPAR_L
- FC_RXCSOF
- FC_RXCSOF_L
- FC_RXCSRB
- FC_RXCSRB_L
- FC_RXCDAT(3:0)
- FC_RXCDAT_L(3:0)
- SPI4_TCLK_REF
- SPI4_TCLK_REF_L
- SPI4_RCTL
- SPI4_RCTL_L
- SPI4_RPAR
- SPI4_RPAR_L
- SPI4_RPROT
- SPI4_RPROT_L
- SPI4_RDAT(15:0)
- SPI4_RDAT_L(15:0)
- SPI4_RCLK
- SPI4_RCLK_L

Status: Fixed

51. PCI Signals Being Driven During Reset when Not the Central Function

Problem: PCI signals PCI AD[31:0], CBE[3:0], and PAR are being driven during reset when not configured as the Central Function.

Implication: If the PCI clock is not running during RESET, the following PCI signals will be driven (0/1) when the device is *not* configured as the Central Function (i.e., CLK_CFG_RST_DIR = 0):

- PCI_AD[31:0]
- CBE[3:0]
- PAR

Workaround: The workaround is to provide at least two PCI clock cycles for the logic to initialize properly.

After the PCI clocks have been provided, the signals will be three-stated.

Status: Fixed

52. LVDS RComp Circuits are Driven by the Receive Clock

Problem: LVDS receivers use resistive compensation circuits (RComp) to establish the termination resistor at all the receiver inputs. However, the state machine is driven by the receive clock, which also has a termination resistor controlled by the RComp circuit.

Implication: The RComp circuits will not function properly.

Workaround: Override the RComp circuit. This is done by setting the RCOMP_OVERRIDE bit and RCOMP_VALUE fields in the MSF_IO_BUF_CTL and FC_IO_BUF_CTL registers.

A value of 0x00001420 should be programmed for these registers.

Status: Fixed

53. LVDS Static Alignment Does Not Function Properly

Problem: The LVDS static alignment mode does not function properly due to a logic problem in the unit that does not allow the value programmed into the RX_DESKEW_#[DESKEW_VALUE] register to be used as the capture clock for the interface.

Implication: The LVDS static alignment mode will not function properly.

Workaround: Use training cycles to deskew the data path in place of static alignment mode.

Status: Fixed

54. Incorrect QDR and MSF Clocks

Problem: Programming an odd or even clock divisor value in the Clock Control Register (CCR) results in an incorrect clock frequency generated for the QDR_K/K# and MSF clock outputs. Programming the CCR requires the clocks to be stopped for at least 16 PLL clock cycles. The circuit implementation does not guarantee this dead period between clock changes.

Implication: When programming an odd or even divisor, the QDR and MSF clocks may not function properly and cause undefined behavior.

Workaround: First, program a clock divisor of zero for all clocks except for the APB, wait at least 500 μ s, and write the desired odd or even clock divisor value. To update a single clock divisor value then, the desired divisor must be written to zero (leaving the other divisors unmodified), and after waiting at least 500 μ s, write the desired odd/even divisor. *Note:* writing the same divisor value back to the CCR register does not initiate a change and does not cause the problem to occur. Note that the APB_CLK can never be turned off.

Status: Fixed

55. PCI Memory and I/O Byte Reads from Intel XScale[®] Core Do Not Assert Byte-enables Correctly

Problem: Intel XScale[®] core initiated PCI Memory and I/O byte read operations do not assert the correct byte-enables on the PCI bus.

On the PCI bus, all byte-enables will be asserted instead of the requested byte(s).

This only applies to read operations; write operations are not affected.

Implication: All four bytes will be read from the target during byte-read operations.

Workaround: None

Status: Fixed

56. Master Abort on PCI Controller Causes Wrong Data on Next PCI Burst

Problem: If a Master Abort is received on a write transaction, the transfer is terminated on the PCI bus. After clearing the RMA interrupt, the next PCI burst write will use the wrong data from the previous transaction.

Implication: The next PCI burst write will use the incorrect data from the previous transaction.

Workaround: Perform a “dummy write” to clear the data from the previous transaction.

Status: Fixed

57. MSF, Interleaved SPI-4/CSIX, Three-Partition Mode

Problem: SPI-4 and CSIX packets may be transmitted with extra bytes of padding when using multiple TBUF partitions in certain conditions: when sending CSIX data, control, and SPI-4 packets with different prepend offsets, prepend lengths, payload offsets, and payload lengths.

Implication: When using the multiple TBUF partitions, transmitted SPI-4 and CSIX packets may be corrupted.

Workaround: Do not use multiple TBUF partitions with prepend lengths greater than 0.

Status: Fixed

58. PCI_PAR and PCI_AD are Not Driven Correctly when Soft Reset is Asserted

Problem: When a soft reset is asserted internally and the grant has been parked on the IXP2800, the PCI_ADD lines are three-stated and the PCI_PAR line is driven.

Implication: During the soft reset condition, the PCI_ADD and PCI_PAR lines are not driven according to the PCI specification.

Workaround: None

Status: Fixed

59. PCI DMA Read FIFO Full Causes DMA to Hang

Problem: When doing long DMA bursts from PCI to DRAM, there is a corner case that when the DMA FIFO becomes full, the DMA state machine incorrectly writes to the DMA FIFO.

Workaround: This results in PCI protocol violations and data corruption and then the DMA state machine hangs.

Workaround: None

Status: Fixed

60. PCI RCOMP Control

Problem: The control bus from the RCOMP logic to the IO pad is inverted, which results in the incorrect driver strength being assigned to the I/O driver.

Implication: The incorrect driver strength will be assigned to the I/O driver.

Workaround: The workaround is to let the RCOMP logic find the correct strength, read the value it has found from the PCI_RCOMP_STATUS, and then write the inverse strength into the P_STREN and N_STREN override registers and enable the override mode.

Status: Fixed

61. FCEFIFO Full Bit Asserts Late

Problem: Regardless of the setting of the FCEFIFO_HWM field of the HWM_CONTROL register, the FCEFIFO Full bit in the FC_EGRESS_STATUS register asserts after four CWORDS more than the programmed threshold that has been placed in the FCEFIFO.

Implication: The FCEFIFO Full does not assert as expected.

Workaround: None

Status: Fixed

62. Product_ID Register Not Updated for the A1 Stepping

Problem: The PRODUCT_ID register MIN_REV and MAJ_REV revision numbers were not updated for the A1 stepping.

Implication: The PRODUCT_ID register MIN_REV and MAJ_REV revision numbers are still 0 for the A1 stepping.

Workaround: None

Status: No fix

63. PCI Impedance Compensation Pin Wiring

Problem: The PCI Impedance Compensation pins were wired incorrectly. The pci_zq1 pad is connected to the pci_zq2 package pin and the pci_zq2 pad is connected to the pci_zq1 package pin.

Implication: The PCI RComp circuits will not function properly.

Workaround: For systems using the A0, A1, and A2 steppings to function properly, pin R12 (PCI_ZQ2) must be connected to VSS through a 31.6 ohm resistor. Pin N08 (PCI_ZQ1) must be connected to VCC3.3 through a 28 ohm resistor.

Status: Fixed

64. RDRAM Data Corruption

Problem: RDRAM data corruption may occur when operating the RDRAM interface with a DRAM_CLK_RATIO divisor that is less than 8.

Implication: Random RDRAM data corruption can occur.

Workaround: To work around this issue, the DRAM_CLK_RATIO divisor value programmed in the CLOCK_CONTROL register must be 8 or greater to avoid data corruption.

Status: Fixed

65. LVDS I/O are Reset by Software Reset

Problem: The LVDS RCOMP circuit does not start searching for the correct output impedance until the RX_SECTION_EN and RSX_SECTION_EN bits within the MSF_CLOCK_CONTROL register are enabled. The RX_SECTION_EN and RSX_SECTION_EN bits control this function for the MSF and Flow Control interfaces, respectively. The RCOMP circuit should be enabled to search for the proper impedance when NRESET is de-asserted.

Implication: The RCOMP circuit will not be enabled until the RX_SECTION_EN and RSX_SECTION_EN bits within the MSF_CLOCK_CONTROL register are enabled.

Workaround: The RX_SECTION_EN and RSX_SECTION_EN bits within the MSF_CLOCK_CONTROL register must be enabled for the RCOMP circuit to be enabled.

Status: Fixed

66. ESD Failure on VREFHI_CLK and VREFLO_CLK Input Pins

Problem: The VREFHI_CLK and VREFLO_CLK pins fail at +/- 500 V CDM level test.

Implication: ESD failures may occur on the VREFHI_CLK and VREFLO_CLK pins.

Workaround: None

Status: Fixed

67. ESD Failures on Slow Port/GPIO Port/JTAG/Serial Port Pins

Problem: The Slow Port/GPIO Port/JTAG/Serial Port pins fail at +/- 500 V CDM level test.

Implication: ESD failures may occur on the Slow Port/GPIO Port/JTAG/Serial Port pins.

Workaround: None

Status: Fixed

68. QDR Read Data Slips 1/2 or Full Cycle

Problem: Intermittently, some SRAM channels may initialize incorrectly. The failure mechanism is that the read data shifts by 1/2 or full cycle i.e., for a write of 0x11223344 and 0xAABBCCDD the returned read data would be 0xXXXX1122 or 0xAABBCCDD for a 1/2 or full cycle shift, respectively. Once the channel is in this mode, it must be reset and re-initialized to make it operational.

Implication: When the SRAM channel does not initialize correctly, the read data returned to the SRAM controller is corrupted.

Workaround: There are two parts to the workaround for this issue:

- First, the QDR_RX_DESKEW register should be set to a value of 0x8.
- Second, the DLL needs to be disabled and re-enabled to ensure that it locks correctly. An application note describing the entire QDR channel initialization, including these workarounds, is available from your Intel sales representative.

Status: Fixed

69. Intel XScale® Core/Microengine Initiated and Target PCI Access Anomalies

Problem: A step devices may intermittently return incorrect data or hang when performing Intel XScale® core/Microengine initiated or target PCI accesses. Additionally, an un-commanded PCI transaction may be initiated upon enabling master operations in the command register. The hang condition occurs due to the Intel XScale® core/Microengine initiated or target PCI access not completing on the PCI bus or returning incorrect data. The issues have been root-caused to an improper synchronization of reset in which a glitch can be latched on the de-assertion of reset, resulting in misaligned/initialized FIFO pointers.

Implication: The Intel XScale® core/Microengine or target transaction may hang due to the PCI access not completing on the PCI bus, an un-commanded PCI transaction may be initiated, or undefined behavior may occur due to corrupted data being returned on reads.

Workaround: When the error condition is detected, a RESET is required to clear the failure mode. The software reset – IXP_RESET_0[16](RST_ALL) or the hardware reset – NRESET/PCI_RST, can be used to clear the error condition. When the device comes out of reset correctly, all subsequent operations are error-free. A document describing a software workaround is available from your Intel sales representative.

Status: Fixed

70. MSF TCLK Duty Cycle when Sourced from TCLK_REF Pins

Problem: When the MSF TCLK is sourced using the TCLK_REF pins, the worst-case duty cycle can be 60% - 40%. The positive phase of the clock is longer (60%) than the negative phase (40%). *Note:* if TCLK is sourced from the internal PLL, the duty cycle is good.

Implication: The MSF TCLK may violate the SPI-4 specification. The additional clock uncertainty must be accounted for in the timing analysis of the interface.

Workaround: None

Status: Fixed

71. SPI-4 DIP-4 Errors when the Clock Edge of Training Changes

Problem: When the clock edge of the first training phase changes from rising to falling or from falling to rising, a DIP-4 error may erroneously be detected. For example, if a training sequence was initiated and the first phase was launched on a rising edge of the clock and a subsequent training sequence was launched on a falling edge, a DIP-4 error is incorrectly reported. This issue does not affect training sequences initiated between IXP2800 devices because the IXP2800 always initiates training on the same clock edge.

Implication: A DIP-4 error is erroneously reported when subsequent training sequences are launched on the opposite (rising-to-falling, falling-to-rising) edge of the clock. Additionally, if a packet immediately follows the training sequence, the packet will be pushed to the Microengines with a bad status, when in fact the packet was good.

Workaround: DIP-4 errors detected coincident with training should be ignored. However, if a packet immediately follows a training sequence (no IDLE cycle in-between) and the training sequence was initiated on a different clock edge than in the preceding cycle, then this packet will be pushed to the Microengine with a bad status, as described above. To prevent this from occurring, the Alpha count value on the transmitting side should be set to a value of 2 or greater. This ensures that two or more back-to-back training cycles are performed each time training is requested, and this will make the DIP-4 error occur *between* training cycles because only the first sequence initiated from a different clock edge will trigger the error. This eliminates the need for software to discern between actual corrupted packets and packets that may have been marked corrupted due to training.

Status: Fixed

72. QDR Dequeue Performance

Problem: To meet OC-192 line rates, dequeue-to-dequeue turnaround must complete within a minimum-size POS packet time with the Intel POS Reference design. The QDR dequeue performance is insufficient to meet this requirement with an INTERNAL_PIPELINE CSR set to 0x2. The extra cycle of latency is required to capture the read data when operating the interface at 200 MHz (or greater) in the A step of the device.

Implication: The performance impact occurs only if the transmit scheduler issues back-to-back dequeues to the same queue at minimum packet rates. In that case, the hardware requires one extra cycle over budget to perform the second dequeue (at 200 MHz).

This extra cycle of latency can result in the following two scenarios:

1. The dequeue results arrive at the transmit code late; thus possibly, the transmit itself may be late. We do not believe that this will have much impact on performance as the extra delay is well within the uncertainty of the dequeue command to dequeue response.
2. If the back-to-back sequence to the same queue is issued multiple times, the input buffering in the SRAM controller can fill up, in which case back pressure will assert and stall the command bus. The chip-wide impact of this behavior can be quite severe because no further SRAM

commands will be accepted from one Microengine cluster for any of the SRAM channels until the back pressure is cleared.

It is best if the transmit scheduler avoids back-to-back dequeues to the same queue number. A small number of these will not have a large impact on performance in A step devices. If, for example, only one queue is active and all packets are of minimum size, then the impact of this issue will be seen as a drop in the supported line rate.

Workaround: This can be overcome by running the QDR interface at 233 MHz.

Status: Fixed

73. JTAG_RST Pin Input Receiver Sensitivity to 2.5 V Supply

Problem: The JTAG_RST input pin may not detect a valid receive high state when the pin is driven to the specified V_{ih} level when the 2.5 V supply is dropped below 2.5 V.

Implication: A valid receive high state may not be detected although the input pin has been driven to the specified V_{ih} level.

Workaround: To guarantee that a valid receive high state is detected, the LVDS supply, VCC25, must be greater than or equal to 2.5 V and the temperature of the system must also be greater than or equal to 25° C when using this test pin.

Status: Fixed

74. Parity Errors Detected During RMW Operations are Written with Good Parity

Problem: When a parity error is detected during the read phase of an RMW, the modified data is written back to memory with correct parity. The specification indicates that the data should be written with incorrect parity. The detection of the parity error is preserved and an interrupt is generated to the Intel XScale® core. Also, the event signal is not returned to the Microengine if the device and instruction are configured to do so. The address of the location where the parity error occurred is captured correctly in the SRAM_PARITY_STATUS_1 register.

Implication: Upon reading the data that was written back as part of the RMW instruction marked with a parity error, a subsequent parity error will not be reported although the data is corrupt.

Workaround: The address of the read parity error is captured and reported correctly; hence, the interrupt or error-handling software should clear this memory location upon the detection of the error.

Status: No Fix

75. MSF SPI-4 Parity Error Reported on Valid Packets

Problem: In certain conditions, the MSF unit can incorrectly mark a valid SPI-4 packet with a parity error.

This incorrect parity error status can result in the following two scenarios:

1. When a valid SPI-4 packet is immediately followed by a packet with a parity error, the good packet is sometimes marked with a parity error. This can occur when an EOP control word with good DIP-4 parity and an SOP control word with incorrect parity, are received in back-to-back cycles. In this case, both packets may be marked with a parity error when only the second packet had an error.
2. When an invalid SPI-4 packet precedes a valid SPI-4 packet, then both packets may be marked with a parity error when only the first packet had an error.

Implication: A valid packet may be pushed to the Microengine with the parity error indication set in the status word and hence the packet will be dropped.

Workaround: None

Status: No Fix

76. LVDS TSTAT Port Incorrectly Recognizes Data Patterns as Training

Problem: When the TSAT port is configured in LVDS mode under certain data flow conditions, the calendar data patterns may be incorrectly detected as training sequence patterns and as a result, initiate a training update.

Implication: As a result of the training update, the TSTAT interface will deskew incorrectly, resulting in incorrect data or DIP2 errors being detected on the interface.

Workaround: The problem can be avoided by setting the IGN_TRAINING bit in the TRAIN_CALENDAR register after performing training. The bit should only be cleared by software when training is required and when training patterns are present on the interface (i.e., no calendar information is being received). IGN_TRAINING is then set again, after training has been successfully completed. This prevents the deskew logic from incorrectly recognizing data patterns as training.

The problem can also be worked around by maintaining a skew relationship between FC_RXCDATA[0] and FC_RXCDAT[1] signals to less than 1 bit time in the PCB layout.

Status: Fixed

77. Slowport Mode 1, Mode 3, and Mode 4 May Read Data Incorrectly

Problem: During a read operation when the Slowport interface is configured in self-timing Mode 1, Mode 3, or Mode 4 and the interface is running at 50 MHz, then the read data may not be captured correctly. The slowport interface has a one-clock delay between the assertion of SP_OE_N and when the IXP2800 strobes the first byte of data; this may not allow enough time for the glue logic to drive the data onto the SP_AD bus and meet the setup requirements.

Implication: When reading data from the Slowport, the captured read data may be corrupted.

Workaround: Ensure that the glue logic timing characteristics are fast enough to drive the data onto the bus to meet the setup requirement in one clock period, as well as to keep the device physically close to the IXP2800 to reduce the propagation delay on the PCB. Additionally, the interface can be run at a slower speed than 50 MHz to alleviate the timing requirements on the glue logic.

Status: Fixed

78. FCIFIFO Errors Reported when Using LVDS TSTAT Mode

Problem: When the MSF is configured to use LVDS I/O for the TSTAT status interface, false FCIFIFO_ERRs are reported in the MSF_INTERRUPT_STATUS register.

Implication: An interrupt to the Intel XScale[®] core will be incorrectly generated for this condition when the corresponding bit 6 is set in the MSF_INTERRUPT_ENABLE register.

Workaround: FCIFIFO_ERR interrupts should not be enabled in the MSF_INTERRUPT_ENABLE register and the FCIFIFO_ERR indication in the MSF_INTERRUPT_STATUS register should be ignored when the status interface is configured to use LVDS I/O.

Status: Fixed

79. PCI Drives REQ64# During Soft Reset

Problem: When the PCI interface is not configured to be the central function, is operating in 64-bit mode, and the bus has been parked on the IXP2800, then the device may incorrectly drive the REQ64# signal if a soft reset is initiated via the IXP_RESET registers.

Implication: In the scenario described above, the REQ64# signal will be driven while the bus is parked on the IXP2800. After the park state (bus idle) has been removed, the REQ64# signal is no longer driven.

Workaround: None

Status: Fixed

80. TXCSR Disabled in Duplex Mode

Problem: When writing to the MSF TRAIN_FLOW_CONTROL register to enable FORCE_FCDEAD and CONTINUOUS_TRAIN bits at the same time, the FORCE_FCDEAD takes precedence over the CONTINUOUS_TRAIN; hence, continuous dead cycles will be transmitted. However, the TXCSR logic disables itself when the CONTINUOUS_TRAIN bit is set in anticipation of training.

Implication: The TXCSR is disabled (remains low) during the duration of dead cycles although it should continue normal operation. The TXCSR signal resumes normal operation when the FORCE_FCDEAD and CONTINUOUS_TRAIN bits are cleared and a single training sequence is transmitted and completed.

Workaround: Do not set the TRAIN_FLOW_CONTROL[FORCE_FCDEAD] and [CONTINUOUS_TRAIN] bits at the same time.

Status: No Fix

81. Flow Control Vertical Parity Error Not Detected

Problem: When a training sequence starts immediately after receiving a CFrame with incorrect vertical parity, and the MSF is running in CSIX full-duplex mode, the CFrame is accepted and placed into the FCIFIFO.

Implication: A CFrame with incorrect vertical parity is placed into the FCIFIFO and no error is detected or reported.

Workaround: None

Status: Fixed

82. Training Sequence Mistaken for Type 3 CFrame

Problem: When Type 3 CFrames (Multicast ID) are mapped to FCEFIFO in full-duplex mode and RPROT is statically tied to a logic 1 during training sequence, the training sequence is enqueued into FCEFIFO and eventually transmitted in cut-through mode across the flow control interface.

Implication: Training sequences will be incorrectly enqueued into the FCIFIFO as Multicast CFrame.

Workaround: Type 3 CFrames must not be mapped to the FCEFIFO when the MSF is configured for CSIX full-duplex mode.

Status: No Fix

83. MSF FCEFIFO Overflow Interrupt Does Not Increment Correctly

Problem: When the MSF flow control interface is configured in simplex mode, and the FCIFIFO overflows the FCEFIFO_OVFLW_CNT, the MSF_INTERRUPT_STATUS register does not increment correctly for each packet that causes an overflow condition.

Implication: The FCEFIFO_OVFLW_CNT will not correctly reflect the actual number of packets that have overflowed.

Workaround: None

Status: No Fix

84. RDRAM Read Failures in Three-channel Mode

Problem: In extreme conditions in which the DRAM controller is being flooded with multiple 32-DWORD burst reads *and* all three DRAM channels are being employed, the DRAM controller may deliver corrupted data during a burst read transaction.

Implication: The DRAM controller will return corrupted data to the requesting unit.

Workaround: Operate the RDRAM interface in one- or two-channel mode.

Status: Fixed

85. Reflect in Four Context Mode Sets Incorrect Event Signal

Problem: The reflect logic in the SHaC unit receives two parameters that define the source and destination transfer registers and the event signal. The reflect logic uses the upper three bits of the 7-bit transfer register address as the context number to determine which context should be signaled in the remote Microengine. The SHaC unit has no knowledge that the remote Microengine is in 4-context mode. At the remote Microengine, when configured in 4-context mode, the 7-bit transfer register address is partitioned as two bits for context and five bits for relative register number. *Note:* in 4-context mode, only contexts 0, 2, 4, and 6 are used; hence, bit 5 can be used to decode the additional 16 transfer registers available in this mode. If the transfer register number is in the low 16 (0x0 – 0x0f) then using the upper three bits as context number will still choose 0, 2, 4, or 6, as desired. However if the transfer register number is in the high 16 (0x10 – 0x1f), then using the upper three bits will choose contexts 1, 3, 5, or 7. So, in that case, instead of setting the event signal for the intended context, an event signal will be set for one of the unused contexts. *Note:* this problem occurs only when the remote Microengine is configured in 4-context mode.

Implication: The SHaC unit will signal one of the unused contexts and the code will stall, waiting for a signal that will never occur.

Workaround: When the remote Microengine is configured in 4-context mode, only the lower 16 context-relative transfer register can be used for reflect operations.

Status: No Fix

86. Slowport I/O May Be Repeated Once for Every Access or Loop Indefinitely

Problem: When APB_CLK is running slower than SP_CLK, for APB_CLK_RATIO divisors of 10, 12, or 15 any Slowport write operation will loop indefinitely. For all other divisors, every Slowport I/O may be repeated once for every access.

Implication: Slowport I/O accesses may loop indefinitely or be repeated once on the Slowport interface; this can cause a side-effect in the application if the data is changing or make the application wait indefinitely for the operation to complete.

Workaround: The SP_CCR register should not be configured such that the Slowport clock is greater than the APB clock. Additionally, no Slowport write transactions should be performed when the APB_CLK_RATIO is set to 10, 12, or 15 if the Slowport clock is running faster than the APB clock. *Note:* the APB clock is set via the APB_CLK_RATIO divisor in the CAP_CLOCK_CONTROL register. The nominal clock frequency for the APB is 50 MHz; hence, at a Microengine frequency of 1 GHz, the APB_CLK_RATIO should be set to 0x5 and at 1.4 GHz, it should be set to 0x7.

Status: No Fix

87. DIP-4 Error Reported when Receive Enable for SPI-4 is Disabled

Problem: The MSF unit incorrectly reports DIP-4 parity errors when valid SPI-4 packets are received and the receive section is disabled in the RX_CONTROL register – i.e., RX_CONTROL[RX_EN_S] is set to 0.

Implication: DIP-4 parity errors are incorrectly reported in the MSF_INTERRUPT_STATUS register.

Workaround: Application software should ignore DIP-4 errors reported when the receive section is disabled – i.e., RX_CONTROL[RX_EN_S] = 0.

Status: No Fix

88. DIP-4 Error Reported on Receive of CIDL E After Continuous CDEAD

Problem: The MSF unit incorrectly reports a single DIP-4 parity error upon receiving the first IDLE CFrame after training sequences are received, immediately followed by continuous dead cycles on the interface.

Implication: A single DIP-4 error is incorrectly reported for the first IDLE CFrame received after the above sequence.

Workaround: None

Status: No Fix

89. High Power Consumption on the 1.3 V Power Supply

Problem: Certain devices have been observed to draw very high current while the 1.3 V supply ramps up after a power-cycle. The high current draw observed at power-up has been root-caused to the CLK_NRESET input receiver state not being propagated from the input receiver pad to the core of the device. This causes the Micro-Stores in the device to initialize incorrectly. This leads to multiple wordline drivers being concurrently active and results in the device entering a high current draw state.

Implication: Systems that provide current limiting may experience the power supply fold-back due to this high current state.

Workaround: The 2.5 V supply must come up prior to the 1.3 V supply and must be at greater than 1 V before the 1.3 V supply can begin to ramp. This new power-up sequence is documented in the *Intel® IXP2800 and 2850 Network Processors Datasheet*.

Status: Doc

90. Random RDRAM Initialization Failures

Problem: Repeated system power-down followed by Rambus* initialization causes failures when the IXP2800 attempts to perform writes/reads to the RDRAM devices. The failure varies from 1 of 10 to 1 of 40 power sequences. These random RDRAM power-up initialization failures have been root-caused to an anomaly in which the RDRAM devices power up in a state in which they incorrectly drive random pins on the interface.

Implication: Systems can randomly fail the RDRAM initialization procedure after a power-sequence.

Workaround: The external RDRAM devices must reset before the RAC current calibration occurs, moving the SIO reset event before current calibration resolves the problem. This new initialization sequence is documented in the *Rambus* Initialization Application Note*, which is available from your Intel sales representative.

Status: Doc

91. PCI DMA with Microengine PCI Read Transactions Fail Intermittently

Problem: When a mixture of IXP2800-initiated PCI-to-DRAM DMA transactions of arbitrary size and address is combined with Microengine-initiated PCI Read operations with a burst count greater than 1, that are either:

- Aligned to a 64-bit address (i.e., bit 2 of the address is 0) with an odd burst count (3, 5, etc.)

- Unaligned to a 64-bit address (i.e., bit 2 of the address is 1) with an even burst count (2, 4, etc.)

then the DMA transfer can fail in one of two ways:

- The DMA engine will stall indefinitely after delivering only the first double word of the DMA transfer, and the upper word of the double word written to DRAM will be corrupted.
- The DMA engine will stall indefinitely without transferring any data.

The Microengine-initiated PCI read is not affected and it completes without error.

Note: Read transactions with a burst count of 1 and Write transactions of any burst size and address alignment are not affected. Additionally, the owner of the DMA channel is arbitrary – it can be PCI, the Intel XScale[®] core, or a Microengine.

Implication: Intermittently incorrect data may be transferred to DRAM and the DMA controller may stall indefinitely.

Workaround: Microengine-initiated PCI reads with a burst count greater than 1, when combined with DMA transactions, must be restricted to 64-bit aligned addresses (bit 2 of the starting address must be 0) and have even burst count sizes (2, 4, 6 etc.).

Status: No Fix

92. MSF Interleave TX Arbitration is Not Fair in Three-Partition Mode

Problem: When the MSF is configured in three-partition mode, 2 CSIX and 1 SPI-4, the arbitration favors the SPI-4 partition. That is, as long as the SPI-4 partition has a valid element, the SPI-4 partition will continue to be serviced.

Implication: CSIX partitions are not serviced until the SPI-4 partition reaches a TBUF element that is invalid.

Workaround: TBD

Status: Fixed

93. SRAM Controller Stall

Problem: An SRAM read command may be incorrectly ordered behind a write command that has already been retired. In certain circumstances, the falsely deferred read command can head-of-line block the Defer Command FIFO. At this point, if additional ordered read commands arrive, the Defer Command FIFO can fill and the SRAM controller will deadlock. A report with additional details is available from your Intel sales representative.

Implication: No further SRAM commands can be performed on any of the four SRAM channels after the Defer Command FIFO becomes full.

Workaround: To minimize the probability of the deadlock, the falsely deferred read commands should be moved through the Defer Command FIFO quickly. To achieve this, it must be ensured that all read commands preceding the falsely deferred read are executed quickly as well. This can be achieved by:

- Limiting the REF_CNT on SRAM READ Commands to 1

For example, if the original code performed a read burst of 2:

```
SRAM[read, $x0, address, 0, 2], ctx_swap [sig]
```

This would be changed to:

```
SRAM[read, $x0, address, 0, 1], sig_done [sig1]
```

SRAM[read, \$x1, address, 4, 1], sig_done [sig2]

Ctx_arb[sig1,sig2]

- Limiting or eliminating SRAM atomic operations, as these commands are also placed in the Defer Command FIFO.

Status: Fixed

94. Invalid Duty Cycle on Internal Clocks

Problem: Programming an odd clock divisor value in the Clock Control Register (CCR) results in an incorrect clock duty cycle for internally-generated clocks when the core clock frequency is set to 1.4 GHz.

Implication: Incorrect clocks may affect the operation of the device and result in undefined behavior.

Workaround: The device must be operated with the core voltage supply set at a minimum of 1.35V.

Status: Fixed

95. Intel XScale® Core JTAG Debug Instructions May Be Misinterpreted

Problem: When the IXP2800 is configured for Intel XScale® core JTAG debug, certain Intel XScale® core debug commands may be interpreted incorrectly. The transition from the following XScale® core JTAG commands may result in the command being misinterpreted; that is, the first command listed must be immediately followed by the second command:

- Intel XScale® core DBGRX command followed by the XScale® core DCSR command
- Intel XScale® core DBGRX followed by the XScale® core DBGTX command
- Intel XScale® core DBGRX command followed by the XScale® core SNAPDAT command
- Intel XScale® core AMPDELAY command followed by the XScale® core SNAPDAT command
- Intel XScale® core AMPDELAY command followed by the XScale® core DBGTX command
- Intel XScale® core LDIC command followed by the XScale® core DBGTX command
- Intel XScale® core BIST command followed by the XScale® core DBGTX command
- Intel XScale® core SNAPDAT command followed by the XScale® core DBGTX command

Note: this erratum is only applicable when using the Intel XScale® core debug functionality. JTAG boundary scan operations are not affected by this erratum.

Implication: The result of the command being misinterpreted can cause one of three outcomes, which can result in undefined behavior:

- The output pins may be incorrectly placed in a HIGH-Z state.
- For input pins, data from the JTAG Boundary Scan register may be passed to the core instead of the data present at the input pins.
- The output pins may incorrectly drive out the JTAG Boundary Scan Register data instead of the data from core.

Workaround: None

Status: No Fix

96. Ingress and Egress Switch Fabric Ready Bits Not Cleared on Continuous Dead

Problem: When the MSF is configured for CSIX mode, the SF_DREADY and SF_CREADY bits in the FC_EGRESS_STATUS and FC_INGRESS_STATUS registers are not cleared upon receiving continuous dead cycles. The Egress SF_DREADY and SF_CREADY bits should be cleared on the reception of continuous dead cycles on the RDAT interface, and in Simplex mode the Ingress SF_DREADY and SF_CREADY bits should be cleared upon reception of continuous dead cycles on the RXCDAT bus.

Implication: The TXDAT and TXCDAT interfaces will continue to transmit data until the continuous dead cycles are transferred across the FLOW Control bus, although the switch fabric is requesting training for its RX interface.

Workaround: None

Status: No Fix

97. Internal Divided Clocks Have Incorrect Duty Cycle After Reset

Problem: After reset is de-asserted, the internal divided clocks within the device – SRAM, RDRAM, and MSF – may have an incorrect duty cycle with the default clock divisors (15, 0xF) within the CAP_CLOCK_CONTROL register.

Implication: Incorrect clocks may affect the operation of the device and result in undefined behavior.

Workaround: Write any new clock divisor value, odd/even, to the CAP_CLOCK_CONTROL register for each of the divided clock. If the default value of 0xF is desired, then write any value other than 0xF to the desired nibble first, followed by a write of 0xF.

Note: The APB_CLOCK divisor is not affected by this erratum.

Status: Fixed

98. RDRAM I/O JTAG SAMPLE Command

Problem: The JTAG SAMPLE/PRELOAD command does not sample correctly for the RDRAM I/O pins; this affects all pins except for the RQ[7:0], SCK, and SCMD (output only) pins.

Implication: The SAMPLE/PRELOAD command cannot be used to sample data being input into the affected pins.

Workaround: Load EXTEST instruction and load zeros for all affected pins and go to the UPDATE_DR state; this effectively tri-states the bus. At this point, drive the pads with the desired pattern and go to the CAPTURE_DR state to capture the data being driven onto the pins. Shift captured data out to TDO.

Note: The EXTEST, CLAMP, and HIGHZ commands are not affected by this erratum.

Status: No Fix

99a. Write Commands with Indirect Reference in Four Context Mode

Problem: When an I/O reference with a write command is being performed in four-context mode with an indirect reference to override the transfer register and the instruction included the name of the transfer register in the XFER parameter and said name maps to a physical register between 16 and 31; then the data is pulled from the incorrect transfer register and incorrect data is written to the I/O target.

Implication: Incorrect data will be written to the specified I/O address.

Workaround: The programmer should specify "--" in the XFER parameter, i.e.:

```
MSF[write, --, addr, 0,1], indirect_ref;
```

```
SRAM[write, --, addr, 0,1], indirect_ref;
```

Status: No Fix

99b. Read Commands with Indirect Reference in Four Context Mode

Problem: When an I/O reference with a read command is being performed in four-context mode with an indirect reference to override the transfer register and the instruction included the name of the transfer register in the XFER parameter and said name maps to a physical register between 16 and 31; then the data is pushed to the incorrect transfer register.

Implication: The incorrect transfer register will be written with the I/O data.

Workaround: The programmer should specify "--" in the XFER parameter, i.e.:

```
MSF[read, --, addr, 0,1], indirect_ref;
```

```
SRAM[read, --, addr, 0,1], indirect_ref;
```

Status: No Fix

100. Small CFRAMEs Should Be Followed by Dead Cycles

Problem: When the MSF interface is configured to operate in CSIX mode with a CWORD size of 64 bits, the transmitter may not transmit a dead cycle after transmitting a flow control CFRAME that contains only a single Virtual Output Queue (VOQ) reference. Additionally, for configurations using a CWORD size of 96 or 128 bits, a dead cycle does not necessarily follow a CFRAME with a length of only a single CWORD.

Implication: On the CSIX interface bus, the SOF signal will be asserted on two successive CWORDS.

Workaround: For 64-bit CWORDS, transmit at least two VOQ references in any VOQ CFRAME. The second reference can be the same as the first (repeated) or a dummy reference. For 96- and 128-bit CWORD sizes, there is no workaround.

Status: Fixed

101. SRAM Read Latency

Problem: The SRAM read arbiter gives the highest priority to the Deferred Read Command FIFO (holds read portion of Atomic commands and read that were deferred due to conflict with pending writes) over the Read Command FIFO (holds all other reads) and Queue Controller (holds reads for queuing operations) command queues. As such, as long as the Deferred Read FIFO contains commands that no longer need to wait for pending writes, then it will consume 100% of the read bandwidth and block the "victim" read from the Read Command FIFO and/or the Queue Controller.

Implication: If the blocking continues for a long enough period of time, the additional latency incurred on the "Victim" read from a MicroEngine may impact the application performance.

If the "Victim" read is initiated from the PCI, then the data can be corrupted when the discard timer expires before the data is returned from the SRAM controller.

If the "Victim" read is initiated from the Intel XScale® core, then the watchdog timer may expire if the latency is excessive; however the data is returned correctly.

Workaround: If more than nine non-blocked reads arrive in the controller, then this stops new commands from entering into the Deferred Read FIFO and allows the Read FIFO to win arbitration, which minimizes the latency incurred on the read. A MicroEngine thread can be allocated to perform low frequency non-blocking reads to implement this workaround.

Alternatively, the Discard Timer Interrupt can be enabled to the Intel XScale® core or PCI, and appropriate action can be taken if the interrupt is detected.

Or, do not perform PCI target SRAM reads.

Status: No Fix

102. MSF Transmit Arbiter is Unfair in Multiple Partition Mode

Problem: When the MSF is configured in multiple partition mode, SPI4/CSIX, or CSIX Data and Control partitions, the arbiter may favor the CSIX partition only after a DEAD cycle is transmitted on the interface. The transmit arbiter will only transmit a DEAD cycle when there are validated TBUF elements arbitrating for the bus in the following two conditions:

- when switching from CSIX to SPI4, i.e., the previous packet is a CSIX frame and the next packet should be an SPI4 frame
- when the previous packet was a CSIX frame and the size of the packet was smaller than the configured CWORD size

Note that in CSIX two-partition mode, this is only a problem when the CWORD size is configured for 96 or 128 bits.

Implication: In both cases stated above, the previous partition granted may be granted again – thus blocking the other requesting partition.

Workaround: None

Status: Fixed

103. Incorrect Reading from Thermal Diode

Problem: Temperature monitoring devices may obtain distorted readings when sampling the onboard thermal diode and, as such, the temperature reported by the device will be incorrect.

Implication: The junction temperature reported by the thermal monitoring device will be incorrect and return a temperature that is between 5°C and 40 °C too high.

Workaround: The temperature reading accuracy can be improved to +/- 10 °C by using the following equation:

$$T_{diode} = (((T_{diodeReading} + 273.15) * 1.008) / n_{ideality}) - 273.15$$

where $n_{ideality} = 1.115178$ and $T_{diodeReading}$ is the temperature returned by the temperature monitoring device

Status: No Fix

104. Incorrect Status Data on RSTAT in RSTAT Override Mode

Problem: When the MSF interface is configured in RSTAT force override mode, $MSF_RX_CONTROL[RX_CALENDAR_MODE] = 1$, and when $TRAIN_DATA[RSTAT_EN]$ is de-asserted to request training, the last port status that was read prior to $RSTAT_EN$ being disabled is continuously transmitted on the RSTAT interface for the remaining calendar length. After the current calendar cycle is completed, the framing indicator is sent on the RSTAT interface.

Implication: After $TRAIN_DATA[RSTAT_EN]$ is de-asserted, the incorrect calendar entry may be transmitted on the RSTAT interface; as such, more data may be transmitted across the RDAT interface which could or would subsequently be dropped.

Workaround: When using $TRAIN_DATA[RSTAT_EN]$ to request training, the application software should program $MSF_RX_CONTROL[RSTAT_OV_VALUE]$ to 0x10, Satisfied, and then switch to conservative value mode, $MSF_RX_CONTROL[RX_CALENDAR_MODE] = 0$, prior to disabling $TRAIN_DATA[RSTAT_EN]$. This allows the satisfied conservative value, 0x10, to be

transmitted on the RSTAT interface until the current calendar cycle is completed – at which point the framing indicator, 0x11, will be sent.

Note: The MSF_RX_CONTROL register can be modified with a single write.

Status: No Fix

105. Extra SP_CP Pulse in Slowport Mode 1

Problem: When the Slowport interface is configured for Mode 1 operation, during read accesses an extra SP_CP pulse may occur after the SP_RD_L signal has been de-asserted.

Implication: After the SP_RD_L signal is de-asserted, the controller may tri-state its outputs — which could result in invalid data being latched in the F646 devices when the extra SP_CP pulse occurs.

Workaround: In most cases, a PLD is used to implement the glue logic and not discrete F646 devices. In this case, the rising edge of SP_RD_L should be used to latch the read data. The SP_CP signal is not used; hence, the extra pulse is irrelevant.

For unpacking the data from the glue logic back to the IXP2800, the SP_CP signal should only be used when the SP_OE_L signal is asserted. Alternately, for implementations using discrete F646 devices, the controller can register its output data, which will hold the output data valid for an additional clock cycle such that when the extra SP_CP cycle occurs, valid data will be sampled.

Status: No Fix

106. PCI Retry followed by Master Abort May Stall Outbound PCI Transactions

Problem: When an IXP2800-initiated PCI transfer is claimed and retried by a target device and is then followed by a master abort condition (no subsequent DEVSEL from the target) then the PCI unit will enter a state in which queued outbound transactions may become permanently stalled.

Note: This condition can occur only if the target device is removed, powered-down, or reset while an active PCI transaction to that device is underway — indicating that a catastrophic system error has occurred.

Implication: Once the error condition is encountered, outbound PCI operations may stall indefinitely.

Note: If the master abort condition occurs on the first access (no retry) then the error condition is not encountered and the PCI unit functions properly.

Workaround: None

Status: No Fix

107. MSF FCI_Full Signal Asserts Early

Problem: The MSF FCI_Full signal that is supplied to the Microengines asserts before the entire CFrame has been received and checked for errors.

Implication: If the application software uses this signal, FCI_Full, to detect when the FIFO is not empty, then upon reading the FCIFIFO, an IDLE CFrame may be read back, 0x0000FFFF, indicating that the FIFO is empty.

Workaround: The application software should use the FCI_Not_Empty signal to determine when to read from the FCIFIFO. Additionally, the FCI_Full signal can be logically ANDed with the FCI_Not_Empty flag to determine when the FIFO is above the programmed high water mark; with this usage model there is no issue.

Status: No Fix

108. Back-to-Back Premature SOF Error Not Reported

Problem: When a CFrame is truncated by the reception of another SOF CFrame that occurs during the extension header cycle of the first CFrame being received, then an RBUF element is allocated to both packets; however, only one status word reporting the error condition is autopushed to the Microengines.

Note: The premature start of frame condition is an illegal case and a violation of the CSIX protocol.

Implication: As an RBUF element is allocated for both CFrames, and only a single status word is autopushed to the Microengine, one of the RBUF elements is lost. If numerous premature SOF errors occur, then several (and eventually all) of the RBUF elements could be lost.

Workaround: None

Status: No Fix

109. Premature SOF on CFrames Mapped to FCEFIFO

Problem: When the MSF unit is configured in Duplex mode and a CFrame that has been mapped to the FCEFIFO is received, if the CWord size is greater than 32 bits and a premature SOF causes the CFrame being received to be truncated during the zero padding of the CFrame, then this results in the CFrame being transmitted across the flow control bus, but not marked with bad vertical parity because the error occurred in the padding of the CFrame that is not written into the FCEFIFO.

Note: The premature start of frame condition is an illegal case and a violation of the CSIX protocol.

Implication: An invalid packet is transferred across the flow control bus to the Ingress processor and no vertical parity error is injected.

Workaround: None

Status: No Fix

110. QDRI 100 MHz SRAM Unsupported

Problem: Due to lack of customer demand for the QDRI 100 MHz SRAM functionality and manufacturability issues associated with the QDRI 100 MHz SRAM lower operating frequency, this feature will no longer be supported in the IXP2800 network processor product line.

Implication: New and existing designs should only use QDRII SRAM devices.

Workaround: None

Status: Doc

111. A Stepping Command Bus Arbitration

Problem: Bandwidth sharing on internal command buses may not be fair when the command bus requests exceed either the maximum command bus bandwidth or the maximum target (for example, SRAM and DRAM) capability otherwise referred to as over-subscription of the command bus.

Note: This anomaly was found during extreme artificial stress testing of the command arbiter and has not been observed under real application use conditions.

Implication: During the timeframe of over-subscription, masters requesting access to the command bus may not all receive equal access. If the over-subscription persists, i.e., if multiple masters are continually issuing requests to the same arbiter, then certain masters may lose arbitration for an extended period of time. Short bursts of excessive requests will not result in any noticeable reduction in application performance.

Workaround: Application code should not over-subscribe any of the processor's internal or external resources.

Status: No Fix

112. B Stepping Command Bus Arbitration

Problem: Bandwidth sharing on internal command buses may not be fair when the command bus requests exceed the maximum target capability (for example SRAM, DRAM, etc.). If a master is requesting the command bus and the target of the request is full (the *Victim Master*), then the grant is issued to the next requesting master with the highest arbitration priority, *Granted Master*. When this condition occurs, the arbiter incorrectly sets the priority of the *Victim Master* to the lowest priority. This may result in the *Victim Master* not receiving grant on the first arbitration cycle after the target-full condition subsides.

Note: This anomaly was found during extreme artificial stress testing of the command arbiter and has not been observed under real application use conditions.

Implication: During the timeframe of over-subscription, masters requesting access to the command bus may not all receive equal access. If the over-subscription persists – i.e., if multiple masters are continually issuing requests to the same arbiter/target and the target is full, then certain masters may lose arbitration for an extended period of time. Short bursts of excessive requests will not result in any noticeable reduction in application performance.

Note: Over-subscription is the term used for the state where the requests exceed the maximum capability of the interface.

Workaround: Application code should not over-subscribe any of the processor's internal or external resources.

Status: No Fix

113. RDRAM Current Control Failure

Problem: The Rambus* periodic current control operations do not occur for channel designs that implement a single device.

Note: This erratum only applies to channels that implement a single RDRAM device. If a channel has two or more devices on a channel, this erratum does not apply.

Implication: The system may experience RDRAM memory errors due to the lack of the current control operations to adjust the RSL outputs to account for temperature and voltage changes.

Workaround: The problem is resolved during initialization by changing the number of devices on the channel from 1 to 2 in the CONFIG2 CSR. The initialization algorithm changes are documented in the *IXP2800 Hardware Initialization Reference Manual Revision 3*.

Status: No Fix

114. Back-to-Back CRC is not supported even if one of the CRCs is not executed due to a branching code stream

Problem: When CRC instructions are back to back, the CRC control cannot handle branch abort and as a result undefined data may be seen.

Implication: This is only affected on back-to-back CRC instructions.

Workaround: The programmer should add a non-CRC instruction between all back-to-back CRC operations.

```
br[skip_crc#], defer[1]
crc_be[crc_32, reg1, reg2]
crc_be[crc_iscsi, reg1, reg2] ;error: must add a non-CRC instruction here
skip_crc #:
continue microcode...
```

Status: No Fix

115. **When Training is received on the RXC Bus in Duplex Mode, the FC status bits are not cleared automatically.**

Problem: When training is received on the RXC Bus in Duplex mode, the FCI status bits are not cleared automatically. As FC status bits are not cleared, the Ingress IXP2800 sends incorrect ready bit status to the Switch Fabric and the Switch Fabric in turn may continue to transmit CFrames to the IXP2800 potentially causing RBUF to be filled up or overflow.

Implication: As FCI status bits are not cleared upon receiving training on RXC bus in Duplex mode, Switch Fabric may continue to send CFrames to IXP2800 and overflow the RBUF.

Workaround: When Ingress IXP2800 wants to request training for its CBUS receive inputs, it must first write 0x100 to the FC_STATUS_OVERRIDE register. This sets INGRESS_FORCE_EN bit 8 and forces the FC_INGRESS_STATUS tm_cready, tm_dready, sf_cready, sfd_ready bits to 0. After writing the FC_STATUS_OVERRIDE register, the Ingress IXP2800 can then write a 0 to the RXCFC_EN in the TRAIN_FLOW_CONTROL register to request training from the Egress IXP2800. After receiving training the Ingress IXP2800 should set RXCFC_EN in the TRAIN_FLOW_CONTROL register before clearing the INGRESS_FORCE_EN bit 8 in the FC_STATUS_OVERRIDE register. When the Egress IXP2800 initiates training of the CBUS to the Ingress IXP2800 using the CONT_TRAIN or SING_TRAIN in the TRAIN_FLOW_CONTROL register, it must first write 0x200 to the FC_STATUS_OVERRIDE register before writing to TRAIN_FLOW_CONTROL. This sets EGRESS_FORCE_EN bit 9 and forces the FC_EGRESS_STATUS tm_cready, tm_dready, sf_cready, sf_dready bits to 0. After training completes on the CBUS the Egress IXP2800 can then clear the EGRESS_FORCE_EN bit.

Status: No Fix

116. **Performance Monitor Unit CHAP counters will not handle underflow correctly**

Problem: Performance Monitor Unit CHAP counters will not be able to handle underflow properly causing counting errors.

Implication: When underflow of Performance Unit CHAP counters happen, CHAP counters may not reset properly causing counting errors.

Workaround: Do not use Performance Monitor Unit CHAP counters to count down in order to avoid underflow. In the case of where a count down is desired from an initial preset CHAP counter value, use the invert of the desired preset value and count up by clearing the CHAPENV[IOCE] in order to increment counter.

Status: No Fix

117. **When Microengine is operating at 1.4 GHz, the Performance Monitor Unit may not monitor events from the DRAM properly.**

Problem: Due to the physical location of the DRAM, the Performance Monitor Unit cannot meet the timing requirements while monitoring DRAM events when the Microengine is operating at 1.4 GHz.

- Implication:** The PMU will not be able to monitor events in the DRAM unit when the Microengine is operating at a 1.4-GHz clock.
- Workaround:** If DRAM events are to be monitored by the PMU, the Microengine clock frequency will need to be operating at 800 MHz or lower.
- Status:** No Fix

Specification Changes

None for this version of the Specification Update.



Specification Clarifications

None for this version of the Specification Update.

Documentation Changes

None for this version of the Specification Update.