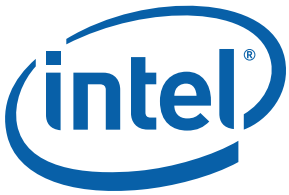


Intel® IXP4XX Product Line of Network Processors

Specification Update

July 2010

Document Number: 306428-008US

Contents

Figures

Tables

Revision History

Date	Revision	Description
July 2010	008	Modifications since the last release of this document: Incorporated the IXP42X Design Checklist into the latest IXP42X Hardware Design Guidelines (revision 009, July 2010); removed associated specification clarifications from this specification update. • Added 1 specification change (see Section 2.4 summary table). • Added 2 documentation changes (see Section 2.6 summary table). Change bars indicate areas of change.
May 2007	007	Modifications since the last release of this document: Incorporated September 2006 IXP4XX Spec Updates into the affected IXP4XX documents that were published, thereby removing them from the current (May 2007) version. • Added 1 Intel XScale® Technology errata (see Section 2.3 summary table). • Added 1 specification change (see Section 2.4 summary table). • Added 3 specification clarifications (see Section 2.5 summary table). • Added 4 documentation changes (see Section 2.6 summary table). • Updated Table 3 with part numbers for B1 stepping of the Intel® IXP42X Product Line of Network Processors.
September 2006	006	Modifications since the last release of this document: • Added item #43 “HDLC Coprocessor is Unable to Capture Alignment Error On a Specific Frame Pattern” on page 40. • Added item #4 “Update on HDLC Coprocessor” on page 53. • Edited item #39 IXP46X Datasheet and Design Guide: Replace K3 with P30 Flash Memory on Page 74 with an additional column on Table 216 and removed the two figures.
June 2006	005	Modifications since the last release of this document: • Updated Table 1 and Table 2 with part numbers for A2 stepping. • Added new SKU of IXP423BD to Table 3. • Added 2 Non-core erratas (see Section 4.0 summary table). • Added 17 specification clarifications (see Section 2.5 summary table). • Added 1 specification change (see Section 2.4 summary table). • Added 13 documentation changes (see Section 2.6 summary table). • Changed instances of *Intel XScale Core* to *Intel XScale Processor.*
December 2005	004	Modifications since the last release of this document: • Added RoHS/lead-free information footnote to Section 3.2 section (2 places). • Modified 1 specification change (see Section 2.4 summary table). • Modified 2 and added 12 specification clarifications (see Section 2.5 summary table). • Modified 1 and added 10 documentation changes (see Section 2.6 summary table).
September 2005	003	Modifications since the last release of this document: • Intel® IXP465 operating at 667 MHz requires Vcc processor Voltage increase from 1.4 V to 1.5 V. • Intel® IXP465 operating at 667 MHz is no longer available with extended temperature support. • Updated Table 1 and Table 2 with part numbers for A1 stepping. • Indicated that Errata 40 (Timer Issues) and Errata 41 (IEEE-1588) have both been fixed for A1 Stepping of IXP46X Silicon. • Added 4 specification changes (see Section 2.4 summary table). • Added 22 specification clarifications (see Section 2.5 summary table). • Added 4 documentation changes (see Section 2.6 summary table).

Date	Revision	Description
May 2005	002	Modifications since the last release of this document: • Incorporated all March 2005 IXP4XX Spec Updates into all affected IXP4XX documents that were published in May 2005, thereby removing them from this May 2005 version of the IXP4XX Spec Update. • Introduced new Intel® IXP45X Product Line of Network Processors to all Intel® IXP46X Product Line documents, including *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Datasheet*, *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual*, *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Hardware Design Guidelines*, *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors: Migrating from the Intel® IXP42X Product Line of Network Processors Application Note.* • Added Table 2, "Part Numbers for the Intel® IXP45X Product Line of Network Processors" on page 20. • Added 1 specification clarification • Added Documentation Change #1 for "Chapter Reorder for IXP45X/IXP46X Product Line Developer's Manual".
March 2005	001	Initial release of this document — combining errata for the entire Intel® IXP4XX Product Line of Network Processors. The first specification update for the Intel® IXP45X and Intel® IXP46X Product Line of Network Processors. Replacement for the separate specification update for the Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor (document number 252702). Modifications since the last release of that document: Edited Non-Core Errata 31 and added Non-Core Errata 36 through 41. Added Specification Changes 1 and 2. Added Specification Clarifications 1 through 12.

1.0 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 and specification clarifications and changes. It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools.

Note: This specification update applies to the Intel® IXP4XX Product Line of Network Processors, which includes the Intel® IXP45X and Intel® IXP46X Product Line of Network Processors and the Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor. It replaces the separate document previously published for the latter product line under the document number 252702.

Unless otherwise specified, the errata assigned to the IXP42X product line network processors applies to all the Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor.

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

This document may contain information that was not previously published.

1.1 Affected Documents/Related Documents

Title	Document Number
Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Datasheet	306261
Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual	306262
Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Hardware Design Guidelines	305261
Intel® IXP45X and Intel® IXP46X Product Line of Network Processors*: Migrating from the Intel® IXP42X Product Line of Network Processors Application Note*	306308
Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet	252479
Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual	252480
Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Hardware Design Guidelines	252817
Intel® IXDP425 / IXCDP1100 Development Platform User's Guide	273743
Intel® IXDP425 / IXCDP1100 Development Platform Quick Start Guide	253177
Intel® IXDP425 / IXCDP1100 Development Platform Documentation Kit	N/A

1.2 Nomenclature

Errata are design defects or errors. These may cause behavior of the Intel® IXP4XX Product Line of Network Processors to deviate from the published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.

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

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

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

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

2.0 Summary Table of Changes

The following tables indicate the errata, specification changes, specification clarifications, or documentation changes that apply to the Intel® IXP4XX Product Line of Network Processors. 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.

The summary tables use the following notations:

2.1 Codes Used in Summary Tables

2.1.1 Stepping

X:	This erratum 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.
N/A	This erratum or document update is not applicable to this stepping of the silicon

2.1.2 Page

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

2.1.3 Status

Doc:	Document change or update will be implemented.
Fix:	This erratum is intended to be fixed in a future step of the component.
Fixed:	This erratum has been fixed for all steppings that are still being shipped.
No Fix:	This erratum has no plans on being fixed.
Plan Fix:	This erratum may be fixed in a future stepping of the product.

Varies: This erratum applies to multiple steppings or devices and the erratum status varies between all steppings *that are still being shipped*.

2.1.4 Row

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

2.2 Non-Intel XScale® Technology Errata (Sheet 1 of 2)

Errata No.	Steppings									Page	Status	Errata
	IXP 425 A0	IXP 42X B0	IXP 42X B1	IXP 45X A0	IXP 46X A0	IXP 45X A1	IXP 46X A1	IXP 45X A2	IXP 46X A2			
1	X									21	Fixed	PCI Doorbell Register Does Not Work Properly
2	X									21	Fixed	UTOPIA Interface Status Collection Synchronization Issues
3	X									22	Fixed	No Byte Enables Asserted During a PCI I/O Read of the Intel® IXP425 Network Processor Causes Unpredictable Behavior
4	X									22	Fixed	Simultaneous AHB Access of the PCI Bus Controller
5	X									22	Fixed	66-MHz PCI Operation
6	X	X	X	X	X	X	X	X	X	22	No Fix	Ethernet Control Protocol Frames Transmit-Defer Status Bit Error
7	X									23	Fixed	Logic 0 is Driven on Both the USB D+ and D- at Reset
8	X									23	Fixed	Cannot Generate Watchdog Timer Reset
9	X									23	Fixed	PCI Non-Prefetch Reads
10	X	X	X							24	Varies	Timer Status Interrupts Get Lost During MMR Writes
11	X	X	X							25	Varies	Character Time-Out Interrupt Sticks Under Certain Software Timing Conditions
12	X									26	Fixed	Intel® IXP425 Network Processor A-0 Stepping May Have Problems Working With Some SDRAM Devices
13	X	X	X							27	Varies	PCI DMA Lock-Up Condition
14	X	X	X	X	X	X	X	X	X	28	No Fix	PCI Doorbell Register Lock-up Condition When Using Two Products Together that have Intel® IXP4XX Product Line of Network Processors
15	X	X	X							28	Varies	PCI Controller Returns Infinite Retries on PCI After AHB Pre-Fetch Error
16	X	X	X							28	Varies	UART Break Indicator
17	X	X	X	X	X	X	X	X	X	29	No Fix	IRQ 3 is Locking the System 'Disable'
18	X	X	X	X	X	X	X	X	X	29	No Fix	EX_IOWAIT_N Timing
19	X									30	Fixed	SOF During Control Read Can Corrupt USB Transfer
20	X	X	X	X	X	X	X	X	X	30	No Fix	USB - Invalid 0x81 Value in Endpoint 0 Control Register on SETUP Transfer
21	X	X	X	X	X	X	X	X	X	30	No Fix	Ethernet Coprocessors — Ethernet Pad Enable Overrides Append FCS
22	X	X	X	X	X	X	X	X	X	30	No Fix	Ethernet Coprocessors — Length Errors on Received Frames
23	X	X	X							31	Varies	PCI DC Parameter VIH Marginality Issue

Note: Unless otherwise specified, the errata assigned to IXP42X B0 processors also apply to the IXC1100 processors.

2.2 Non-Intel XScale® Technology Errata (Sheet 2 of 2)

Errata No.	Steppings									Page	Status	Errata
	IXP 425 A0	IXP 42X B0	IXP 42X B1	IXP 45X A0	IXP 46X A0	IXP 45X A1	IXP 46X A1	IXP 45X A2	IXP 46X A2			
24	X	X	X	X	X	X	X	X	X	31	No Fix	False PCI DMA Completion Notification Causing Data Corruption
25	X	X	X	X	X	X	X	X	X	32	No Fix	Expansion Bus HPI Interface Potential for Contention on Reads with T4=0
26	X	X	X	X	X	X	X	X	X	32	No Fix	PCI Hangs With a Multiple Inbound Error Condition
27	X	X	X	X	X	X	X	X	X	32	No Fix	PCI RCOMP Operation if PCI Clock Stops
28	X	X	X	X	X	X	X	X	X	33	No Fix	UART — Break Condition Asserted Too Early if Two Stop Bits are Used
29	X	X	X	X	X	X	X	X	X	33	No Fix	Ethernet Coprocessors — Address Filtering Logic Ignores the Second to Last Nibble of the Destination Address
30	X	X	X	X	X	X	X	X	X	33	No Fix	USB DC Parameter Vih Specification Change
31	X	X	X							34	Varies	Start of DMA Operation Close to Start of Non-Prefetch (NP) Operation Can Hang NP Operation on AHB
32	X	X	X	X	X	X	X	X	X	34	No Fix	PCI Accesses to the Queue Manager During Queue and SRAM Mode
33	X	X	X	X	X	X	X	X	X	34	No Fix	Ethernet MACs Detect Late Collision Earlier than Ethernet 802.3 Specifications
34	X	X	X	X	X	X	X	X	X	35	No Fix	Read of PCI Controllers BAR 32'h XXFF_FFFC Rd[N] Corrupts Subsequent Rd[N+1]
35	X	X	X	X	X	X	X	X	X	35	No Fix	UART Operating in Non-FIFO Mode Can Falsely Receive Overrun Error
36	X	X	X	X	X	X	X	X	X	35	No Fix	USB-Specification Noncompliance for Rise/Fall Transition Times
37	X	X	X	X	X	X	X	X	X	36	No Fix	Ethernet MAC Does Not Detect Transmit FIFO Underruns Reliably
38	N/A	N/A	N/A	X	X	X	X	X	X	36	No Fix	SMII late_col Occurs Earlier than Expected
39	N/A	N/A	N/A	X	X					37	Fixed	Timer Issues with Prescale Programming Sequence and Pause/ Resume Operation
40	N/A	N/A	N/A	N/A	X	N/A				39	Fixed	IEEE-1588 Time Sync Lock-up Fails to Time-Stamp a Second PTP Message
41	N/A	N/A	N/A	X	X	X	X			39	Fixed	SSP Synchronous Issue When Using External SSP Clock
42	N/A	N/A	N/A	X	X	X	X			40	Fixed	Potential Lockup Condition in The Memory Controller Unit
43	X	X	X	X	X	X	X	X	X	40	No Fix	HDLC Coprocessor is Unable to Capture Alignment Error On a Specific Frame Pattern
44	X	X		N/A	N/A	N/A	N/A	N/A	N/A	40	Fixed	B-1 Stepping Accommodates Future SDRAMs

Note: Unless otherwise specified, the errata assigned to IXP42X B0 processors also apply to the IXC1100 processors.

2.3 Intel XScale® Technology Errata

Errata No.	Steppings									Page	Status	Errata
	IXP 425 A0	IXP 42X B0	IXP 42X B1	IXP 45X A0	IXP 46X A0	IXP 45X A1	IXP 46X A1	IXP 45X A2	IXP 46X A2			
1	X	X	X	X	X	X	X	X	X	41	No Fix	Abort is Missed When Lock Command is Outstanding
2	X	X	X	X	X	X	X	X	X	41	No Fix	Aborted Store that Hits the Data Cache May Mark Write-Back Data as 'Dirty'
3	X	X	X	X	X	X	X	X	X	42	No Fix	Performance Monitor Unit Event 0x1 Can Be Incremented Erroneously by Unrelated Events
4	X	X	X	X	X	X	X	X	X	42	No Fix	In Special Debug State, Back-to-Back Memory Operations — Where the First Instruction Aborts — May Cause a Hang
5	X	X	X	X	X	X	X	X	X	43	No Fix	Accesses to the CP15 ID Register with Opcode2 > 0b001 Returns Unpredictable Values
6	X	X	X	X	X	X	X	X	X	43	No Fix	Disabling and Re-Enabling the MMU Can Hang the processor or Cause it to Execute the Wrong Code
7	X	X	X	X	X	X	X	X	X	44	No Fix	Updating the JTAG Parallel Registers Requires an Extra TCK Rising Edge
8	X	X	X	X	X	X	X	X	X	44	No Fix	Non-Branch Instruction in Vector Table May Execute Twice After a Thumb Mode Exception
9	X	X	X	X	X	X	X	X	X	45	No Fix	Intel XScale® processor Non-Branch Instruction in Vector Table

Note: Unless otherwise specified, the errata assigned to IXP42X B0 processors also apply to the IXC1100 processors.

2.4 Specification Changes

Ref. #	Steppings									Page	Status	Specification Changes
	IXP 425 A0	IXP 42X B0	IXP 42X B1	IXP 45X A0	IXP 46X A0	IXP 45X A1	IXP 46X A1	IXP 45X A2	IXP 46X A2			
1	X	X	X							45	Doc	IXC1100 Control Plane Processors Discontinued
2			X							45	Doc	Update ProdRevID in DevManual for IXP42X B-1
3	X	X	X							46	Doc	External Crystal Support is No Longer Supported with IXP42X

Note: Unless otherwise specified, the errata assigned to IXP42X B0 processors also apply to the IXC1100 processors.

2.5 Specification Clarifications

Ref. #	Steppings									Page	Status	Specification Clarifications
	IXP 425 A0	IXP 42X B0	IXP 42X B1	IXP 45X A0	IXP 46X A0	IXP 45X A1	IXP 46X A1	IXP 45X A2	IXP 46X A2			
1	X	X	X							46	Doc	Impedance Recommendation Should be Consistent at 50 W
2	X	X	X	X	X	X	X	X	X	47	Doc	RJ11 Connector Erroneously Connected to RS-232 Block
3.	X	X	X							47	Doc	IXP42X using Compact Flash Application Note: Connector Recommendations
4.	X	X	X							47	Doc	IXDP425 User's Guide: Expansion Bus Configuration Straps
5.	X	X	X	X	X	X	X	X	X	48	Doc	IXP42X / IXP46X Developer's Manual - UART Baud Settings Table
6.				X	X	X	X	X	X	51	Doc	IXP46X Developers Manual - EXP_TIMING_CS0 Reset Hex Value
7.	X	X	X	X	X	X	X	X	X	51	Doc	Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual and Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual: PCI Configuration Read/Write Discrepancy

Note: Unless otherwise specified, the errata assigned to IXP42X B0 processors also apply to the IXC1100 processors.

2.6 Documentation Changes (Sheet 1 of 2)

Ref. #	Steppings									Page	Status	Documentation Changes
	IXP 425 A0	IXP 42X B0	IXP 42X B1	IXP 45X A0	IXP 46X A0	IXP 45X A1	IXP 46X A1	IXP 45X A2	IXP 46X A2			
1	X	X	X							52	Doc	Unused Acronyms that must be Removed from the List
2	X	X	X							52	Doc	Configuration Strapping Clock Settings, and Clarification
3.	X	X	X							53	Doc	Intel® IXDP425 / IXCDP1100 Development Platform User's Guide: Update Configuration Register Table
4.	X	X	X	X	X	X	X	X	X	53	Doc	Update on HDLC Coprocessor
5.	X	X	X							55	Doc	Improvement of the SDRAM Initialization Sequence
6.	X	X	X							56	Doc	Additional Design Note for SDRAM Initialization/Reset Timing
7.	X	X	X	X	X	X	X	X	X	57	Doc	Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual and Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual: Ethernet Register Discrepancies

Note: Unless otherwise specified, the errata assigned to IXP42X B0 processors also apply to the IXC1100 processors.

2.6 Documentation Changes (Sheet 2 of 2)

Ref. #	Steppings									Page	Status	Documentation Changes
	IXP 425 A0	IXP 42X B0	IXP 42X B1	IXP 45X A0	IXP 46X A0	IXP 45X A1	IXP 46X A1	IXP 45X A2	IXP 46X A2			
8.				X	X	X	X	X	X	58	Doc	Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual: MEM_TYPE strap reference
9.				X	X	X	X	X	X	58	Doc	Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual: Update to GPCLKR Register Table
10.	X	X	X	X	X	X	X	X	X	59	Doc	Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual and Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual: Incorrect Memory Size Indicated in Memory Map

Note: Unless otherwise specified, the errata assigned to IXP42X B0 processors also apply to the IXC1100 processors.

3.0 Identification Information

3.1 Nomenclature of the Intel® IXP4XX Product Line of Network Processors

Figure 1. Nomenclature of the Intel® IXP42X Product Line of Network Processors

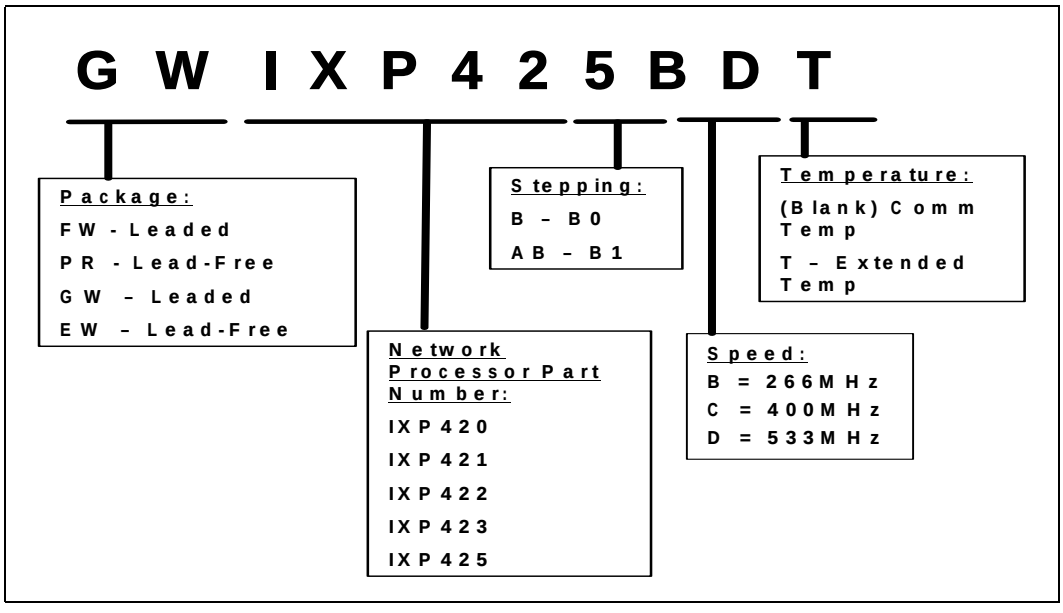
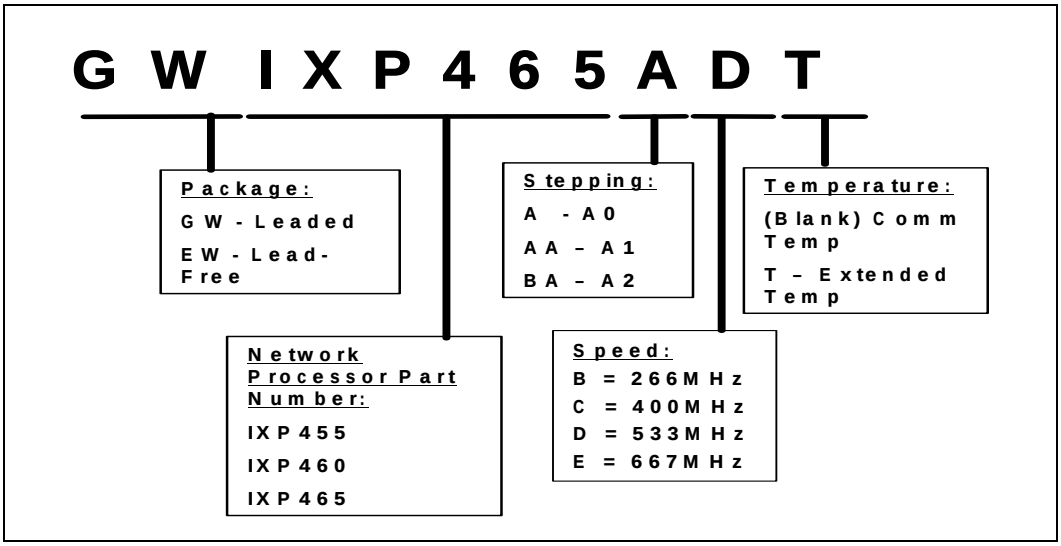
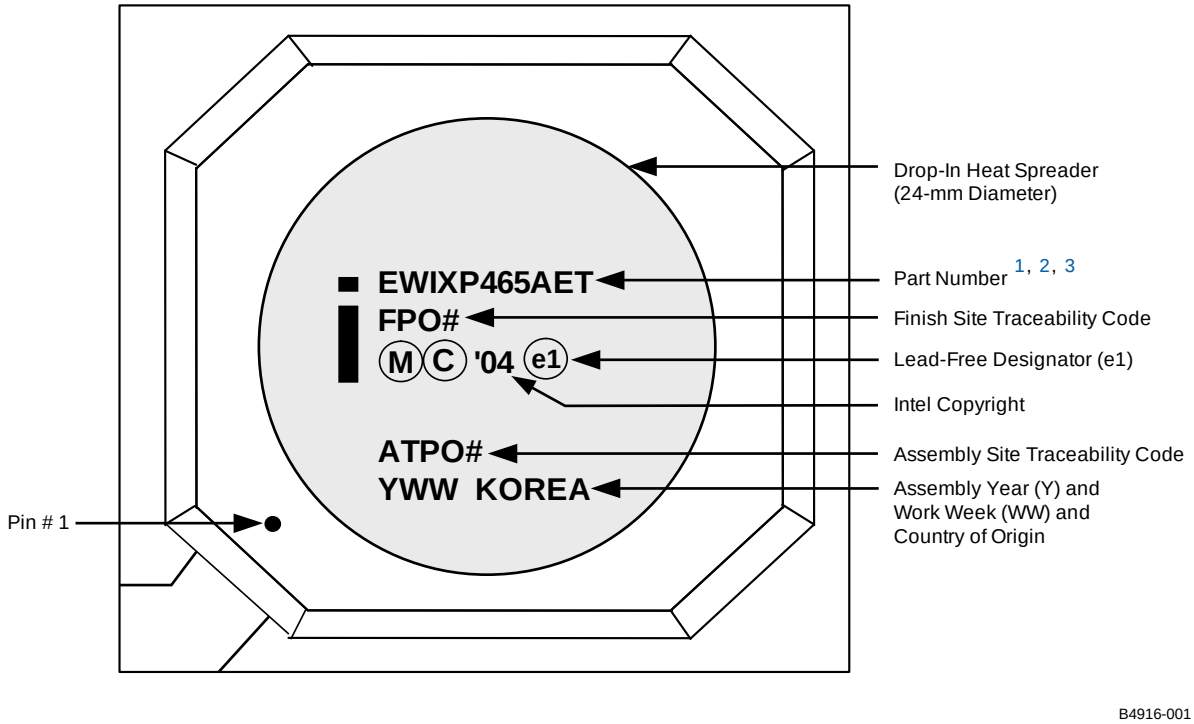


Figure 2. Nomenclature of the Intel® IXP46X Product Line of Network Processors



3.2 Markings

Figure 3. Package Markings: Intel® IXP45X and Intel® IXP46X Product Line of Network Processors — Extended and Commercial Temperature, Lead-Free / Compliant with Standard for Restriction on the Use of Hazardous Substances (RoHS)[1] Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor — Extended Temperature, Lead-Free

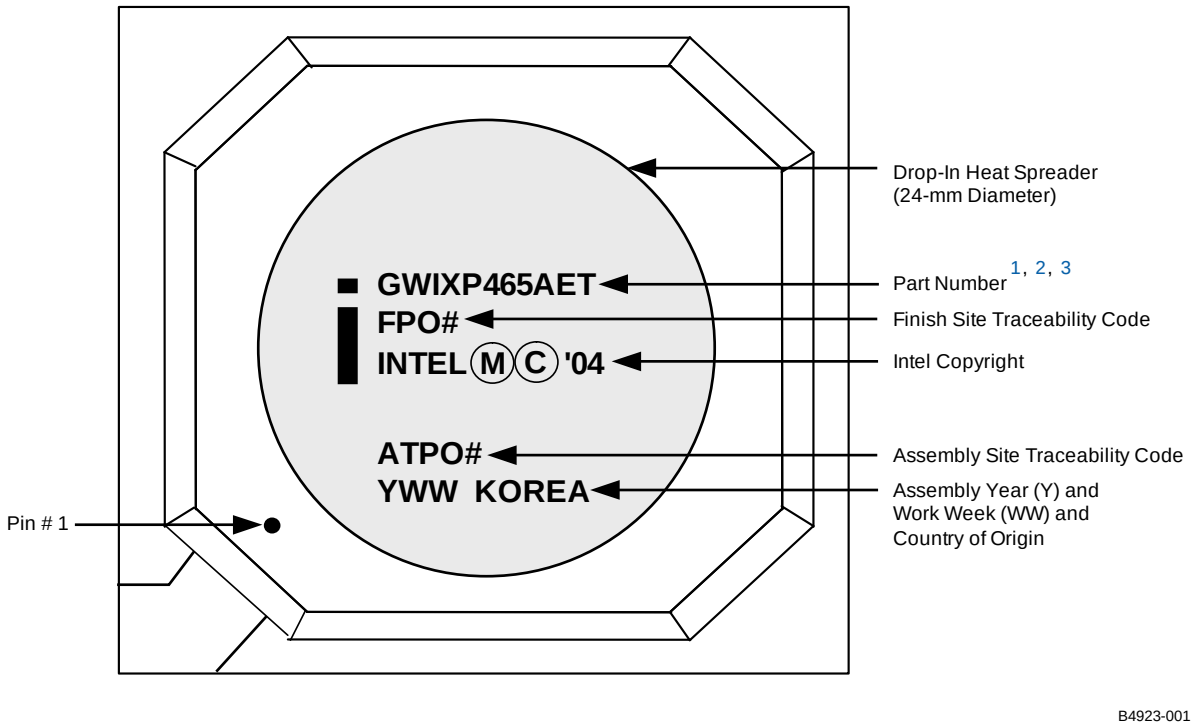


Notes:

1. **Part Number** field — For different part numbers of the Intel® IXP46X Product Line of Network Processors, see Table 1. For the Intel® IXP45X Product Line, see Table 2. For the IXP42X and IXC1100 processors, see Table 3.
2. Package ball counts — The Intel® IXP45X and Intel® IXP46X Product Line of Network Processors have a ball count of 544. The IXP42X and IXC1100 processors have a ball count of 492.
3. Drawing is not to scale. Marking content is an example.

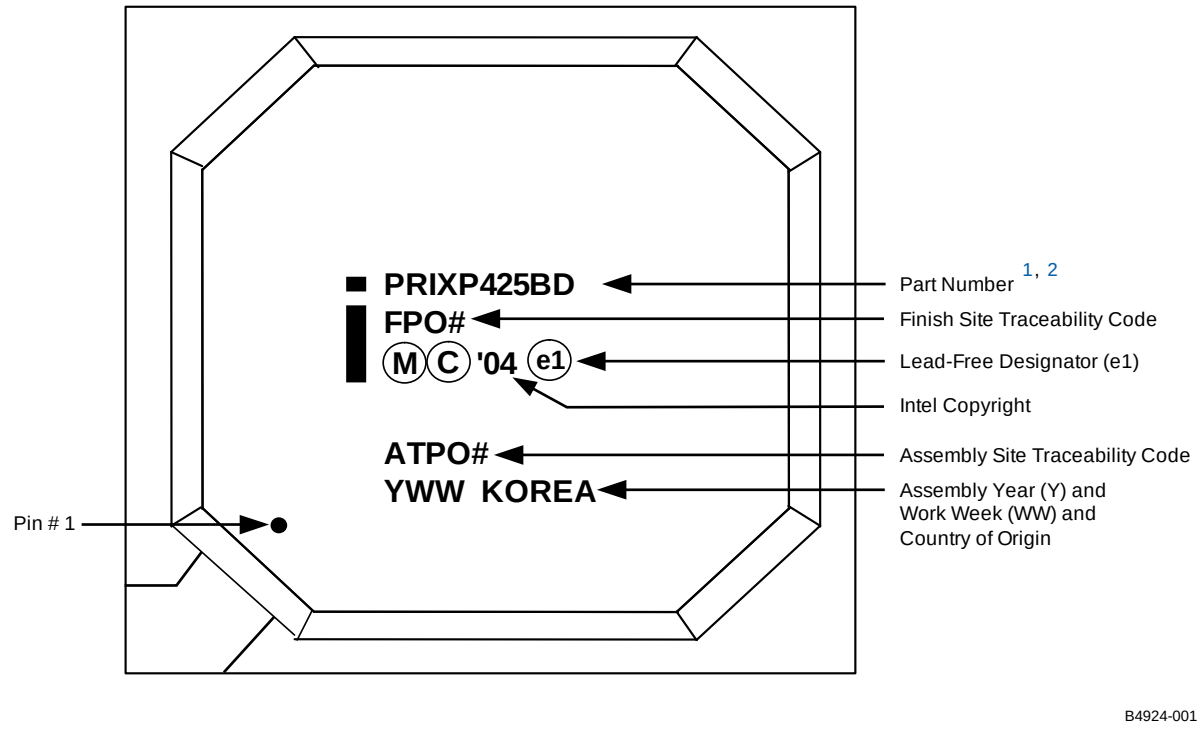
1. Further information regarding RoHS and lead-free components can be obtained from your local Intel representative; for general information, see http://www.intel.com/technology/silicon/leadfree.htm.

Figure 4. Package Markings:
Intel® IXP45X and Intel® IXP46X Product Line of Network Processors — Commercial and Extended Temperature, Lead-Based
Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor — Extended Temperature, Lead-Based



Notes:
1. **Part Number** field — For the different part numbers of Intel® IXP46X Product Line of Network Processors, see Table 1. For the Intel® IXP45X Product Line, see Table 2. For IXP42X and IXC1100 processors, see Table 3.
2. Package ball counts — Intel® IXP45X and Intel® IXP46X Product Line of Network Processors have a ball count of 544. IXP42X and IXC1100 processors have a ball count of 492.
3. Drawing is not to scale. Marking content is an example.

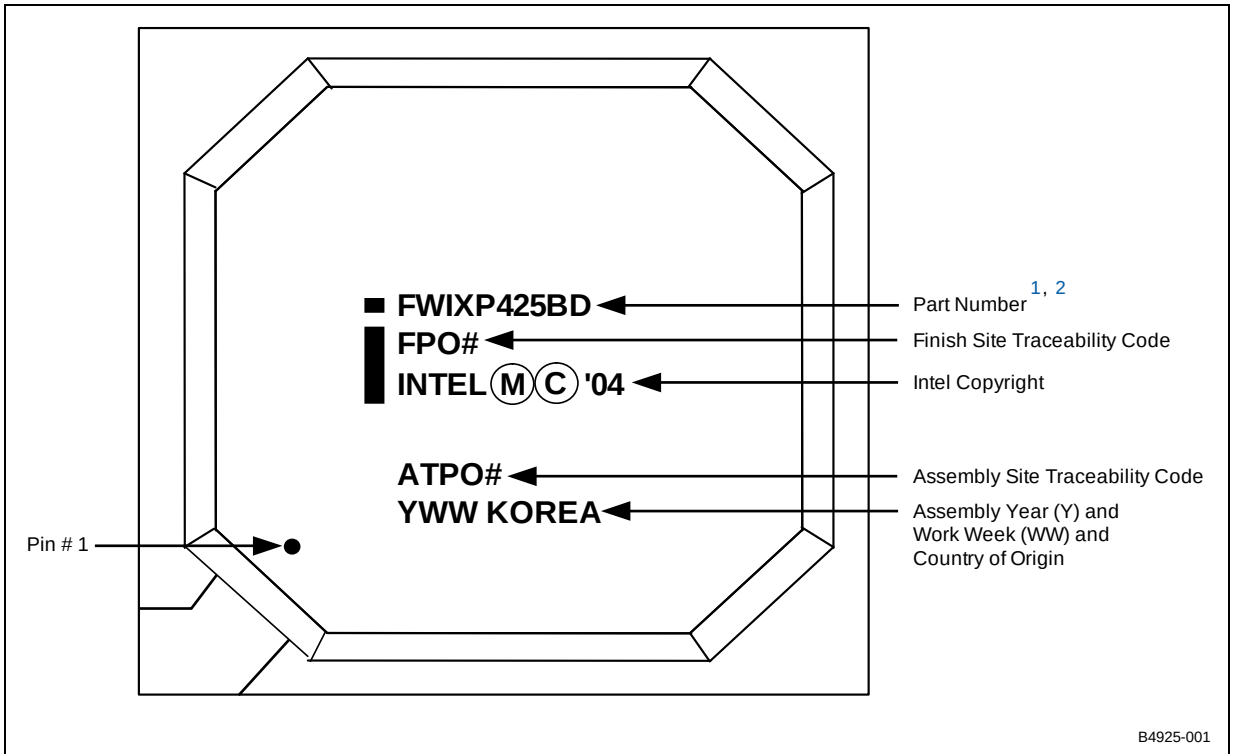
Figure 5. Package Markings: Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor — Commercial Temperature, Lead-Free / RoHS-Compliant[1]



Notes:
1. **Part Number** field — For the different part numbers of IXP42X and IXC1100 processors, see Table 3.
2. Drawing is not to scale. Marking content is an example.

1. Further information regarding RoHS and lead-free components can be obtained from your local Intel representative; for general information, see http://www.intel.com/technology/silicon/leadfree.htm.

Figure 6. Package Markings: Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor — Commercial Temperature, Lead-Based



Notes:
1. **Part Number** field — For the different part numbers of IXP42X and IXC1100 processors, see Table 3.
2. Drawing is not to scale. Marking content is an example.

3.3 Part Numbers

3.3.1 Intel® IXP46X Product Line of Network Processors

Table 1. Part Numbers for the Intel® IXP46X Product Line of Network Processors (Sheet 1 of 2)

Device	Stepping	Speed (MHz)	Temperature Offering	Lead Free	Part #
Intel® IXP465	A2	667	Commercial	Yes	EWIXP465BAE
Intel® IXP465	A2	533	Commercial	Yes	EWIXP465BAD
Intel® IXP465	A2	533	Extended	Yes	EWIXP465BADT
Intel® IXP465	A2	667	Commercial		GWIXP465BAE
Intel® IXP465	A2	533	Commercial		GWIXP465BAD
Intel® IXP465	A2	266	Commercial		GWIXP465BAB
Intel® IXP465	A2	533	Extended		GWIXP465BADT
Intel® IXP460	A2	533	Commercial	Yes	EWIXP460BAD

Table 1. Part Numbers for the Intel® IXP46X Product Line of Network Processors (Sheet 2 of 2)

Device	Stepping	Speed (MHz)	Temperature Offering	Lead Free	Part #
Intel® IXP460	A2	533	Commercial		GWIXP460BAD
Intel® IXP460	A2	266	Commercial		GWIXP460BAB
Intel® IXP460	A2	533	Extended		GWIXP460BADT

3.3.2 Intel® IXP45X Product Line of Network Processors

Table 2. Part Numbers for the Intel® IXP45X Product Line of Network Processors

Device	Stepping	Speed (MHz)	Temperature Offering	Lead Free	Part #
Intel® IXP455	A2	533	Commercial	Yes	EWIXP455BAD
Intel® IXP455	A2	400	Commercial	Yes	EWIXP455BAC
Intel® IXP455	A2	266	Commercial	Yes	EWIXP455BAB
Intel® IXP455	A2	533	Commercial		GWIXP455BAD
Intel® IXP455	A2	400	Commercial		GWIXP455BAC
Intel® IXP455	A2	266	Commercial		GWIXP455BAB

3.3.3 Intel® IXP42X Product Line of Network Processors

Table 3. Part Numbers for the Intel® IXP42X Product Line of Network Processors (Sheet 1 of 2)

Device	Stepping	Speed (MHz)	Extended Temp.	Lead Free	Part #
Intel® IXP425	B-1	533	Yes	Yes	EWIXP425ABDT
Intel® IXP425	B-1	266	Yes	Yes	EWIXP425ABBT
Intel® IXP425	B-1	533		Yes	PRIXP425ABD
Intel® IXP425	B-1	400		Yes	PRIXP425ABC
Intel® IXP425	B-1	266		Yes	PRIXP425ABB
Intel® IXP425	B-1	533	Yes		GWIXP425ABDT
Intel® IXP425	B-1	400	Yes		GWIXP425ABCT
Intel® IXP425	B-1	266	Yes		GWIXP425ABBT
Intel® IXP425	B-1	533			FWIXP425ABD
Intel® IXP425	B-1	400			FWIXP425ABC
Intel® IXP425	B-1	266			FWIXP425ABB
Intel® IXP423	B-1	533		Yes	PRIXP423ABD
Intel® IXP423	B-1	533			FWIXP423ABD
Intel® IXP423	B-1	266		Yes	PRIXP423ABB
Intel® IXP423	B-1	266			FWIXP423ABB
Intel® IXP422	B-1	266		Yes	PRIXP422ABB
Intel® IXP422	B-1	266			FWIXP422ABB
Intel® IXP421	B-1	266		Yes	PRIXP421ABB

Table 3. Part Numbers for the Intel® IXP42X Product Line of Network Processors (Sheet 2 of 2)

Device	Stepping	Speed (MHz)	Extended Temp.	Lead Free	Part #
Intel® IXP421	B-1	266			FWIXP421ABB
Intel® IXP420	B-1	266	Yes	Yes	EWIXP420ABBT
Intel® IXP420	B-1	266	Yes		GWIXP420ABBT
Intel® IXP420	B-1	533		Yes	PRIXP420ABD
Intel® IXP420	B-1	400		Yes	PRIXP420ABC
Intel® IXP420	B-1	266		Yes	PRIXP420ABB
Intel® IXP420	B-1	533			FWIXP420ABD
Intel® IXP420	B-1	400			FWIXP420ABC
Intel® IXP420	B-1	266			FWIXP420ABB

4.0 Non-Intel XScale® Technology Errata Descriptions

1. PCI Doorbell Register Does Not Work Properly

Problem: When a PCI agent external to the IXP425 network processor performs a read of the PCI Controller Control and Status Register — contained in the IXP425 network processor — by targeting the processor's PCI BAR4, the IXP425 network processor may retry the PCI-read operation for an extended period of time before returning the data to the agent. The processor may retry the reads indefinitely. Therefore, reads of PCI Controller Control and Status Registers — from the IXP425 network processor's PCI bus — are not supported. This errata does apply to the Intel® IXC1100 Control Plane Processor A0 stepping.

Implication: The PCI Doorbell Register support on the IXP425 network processor is not accessible.

Workaround: A register can be initialized in SDRAM of the IXP425 network processor and serve as the Doorbell Status Register. When an interrupt occurs to the external PCI agent, the external PCI agent does a target read of this location in the SDRAM of the IXP425 network processor.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — Fixed

Status: IXP46X (A0-Step, A1-Step and A2-Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step) — Fixed

2. UTOPIA Interface Status Collection Synchronization Issues

Problem: There is a synchronization issue in the UTOPIA receive logic that causes incorrect UTOPIA-receive statistics to be gathered. This issue affects the cell count, idle-cell count, HEC-error count, parity-error count, and cell-size error count. This errata does not apply to the Intel® IXC1100 Control Plane Processor.

Implication: This could have impact on MIB counters that may be used for SNMP functionality.

Workaround: None.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — Fixed

Status: IXP46X (A0-Step, A1-Step and A2-Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step)— Fixed

3. No Byte Enables Asserted During a PCI I/O Read of the Intel® IXP425 Network Processor Causes Unpredictable Behavior

Problem: When a PCI agent external to the IXP425 network processor performs an I/O read directed towards the IXP425 network processor by targeting the processor's PCI BAR5, with all byte enables de-asserted (PCI_CBE_N(3:0) = 0xF), the behavior of the IXP425 network processor's PCI interface may become unpredictable, including continuous retry responses on both the PCI bus and the AHB bus internal to the IXP425 network processor. This errata does apply to the Intel® IXC1100 Control Plane Processor A0 stepping.

Implication: PCI I/O reads directed towards the IXP425 network processor with all byte enables de-asserted are not supported.

Workaround: When the IXP425 network processor is placed into this state of operation, a reset to the IXP425 network processor is required.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — Fixed

Status: IXP46X (A0-Step, A1-Step and A2-Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step) — Fixed

4. Simultaneous AHB Access of the PCI Bus Controller

Problem: When two internal AHB masters attempt direct accesses of the same address in PCI memory space, the PCI Controller AHB Slave interface's behavior may become unpredictable, including continuous retry responses on the AHB bus. When a PCI-controller DMA channel is active at the same time as the AHB master operations previously described, PCI accesses directed at the IXP425 network processor may become blocked. This results in continuous retry responses on the PCI bus.

Implication: The NPEs and the Intel XScale® processor cannot access the PCI bus simultaneously using direct-memory access (address 0x48000000 to 0x4FFFFFFF) or, at a minimum, they cannot access the same address at the same time.

Workaround: AHB masters must access non-overlapping regions of the PCI memory space when performing direct memory accesses of the PCI bus.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — Fixed

Status: IXP46X (A0-Step, A1-Step and A2-Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step) — Fixed

5. 66-MHz PCI Operation

Problem: The IXP425 network processor does not meet all of the 66-MHz, AC-timing requirements of the *PCI Local Bus Specification*, Rev. 2.2. This errata does apply to the Intel® IXC1100 Control Plane Processor A0 stepping.

Implication: The PCI interface of the IXP425 network processor operates at a maximum PCI clock speed of 33 MHz.

Workaround: None.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — Fixed

Status: IXP46X (A0-Step, A1-Step and A2-Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step) — Fixed

6. Ethernet Control Protocol Frames Transmit-Defer Status Bit Error

Problem: Ethernet control protocol transmit frames, that are a size of 64 or less, result in the Transmit-Defer status bit being set, regardless of the gap between frames.

Implication: The Transmit-Defer Status bit in the Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor is unusable.

Workaround: None.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

7. Logic 0 is Driven on Both the USB D+ and D- at Reset

Problem: The IXP425 network processor drives a logic 0 onto both the USB D+ and D- lines for the duration of the assertion of the PWRON_RST_N line, after which it tri-states the lines, and lets pull-ups and pull-downs in the system take effect. This errata does not apply to the Intel® IXC1100 Control Plane Processor A0 stepping.

Implication: This appears to the USB-host controller as a “device disconnect” or USB reset. This error could cause a problem when a host controller/hub is transmitting while the PWRON_RST_N line asserts (push button, power-supply monitor reset). This produces contention on either the D+ or D- line until the host recognizes the “device disconnect” and stops driving data.

Workaround: It is recommended that PWRON_RST_N be driven only by a true power-supply event — and never a push button — to limit the occurrence of this problem.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — Fixed

Status: IXP46X (A0-Step, A1-Step and A2-Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step) — Fixed

8. Cannot Generate Watchdog Timer Reset

Problem: When using the IXP425 network processor’s watchdog timer to generate a reset, unpredictable behavior with the IXP425 network processor’s reset logic can occur. This errata does apply to the Intel® IXC1100 Control Plane Processor A0 stepping.

Implication: This problem can cause the IXP425 network processor to not boot properly.

Workaround: The reset capability of the IXP425 network processor watchdog timer should not be used.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — Fixed

Status: IXP46X (A0-Step, A1-Step and A2-Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step) — Fixed

9. PCI Non-Prefetch Reads

Problem: Using the non-prefetch registers to initiate read transactions on the PCI bus of the A0-step processor can cause corrupted data. As a result of a non-prefetch read, data is returned from the PCI bus through a FIFO and the NP_RD_DATA register located in the PCI controller. Under certain conditions the wrong data can be returned from the NP_RD_DATA register as a result of a read. Non-prefetch data is used to produce single cycle IXP425 network processor initiated configuration cycles, memory cycles, I/O cycles or any other valid PCI bus cycle types.

Non-prefetch writes are not affected by this problem. Additionally the PCI DMA channels and memory-mapped PCI windows — used for high-bandwidth, PCI-initiated IXP425 network processor transactions — are not affected by this problem. Target transactions directed to the IXP425 network processor are not affected. This errata does apply to the Intel® IXC1100 Control Plane Processor A0 stepping.

Implication: Invalid data is returned from the NP_RD_DATA register for current access. Subsequent accesses are not affected. This problem can affect the read values for configuration cycles, memory cycles, I/O cycles and any other cycle types generated using the non-prefetch registers.

Workaround: The software work around requires that the perform eight consecutive atomic non-prefetch read operations of the desired location on the PCI bus. Furthermore, the PCI_NP_RDATA register must be read twice, when retrieving the PCI read data. Data

returned from the first seven non-prefetch reads may be in error and is discarded. Data returned from the eighth read (the second read of the PCI_NP_RDATA register of the eighth non-prefetch read operation) is the correct data.

This work around works under the following conditions:

- No other intervening operations to the PCI bus can occur — during the eight non-prefetch reads — from any AHB master.
- The DMA channels in the PCI Controller must be idle.
- The location to be read, on the PCI bus, must have no side-effects on reads, for example an FIFO.
- The location to be read on the PCI bus must contain static data. Alternately — if the data is changing — application must not care which of the previous eight reads gets returned.

In-bound PCI traffic initiated from external PCI devices does not affect the work around, so these operations need not be restricted.

A possible hardware work around is to ensure that the IXP425 network processors' system clock input and PCI clock input have a fixed and known phase relationship. This would eliminate the asynchronous "jitter" between the two signals previously mentioned. Currently, analysis has shown that this known phase relationship sits inside a window that is too small to be implemented in a practical application over the full range of process variation and environmental conditions. Therefore, no hardware work around is recommended at this time.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — Fixed

Status: IXP46X (A0-Step, A1-Step and A2-Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step) — Fixed

10. Timer Status Interrupts Get Lost During MMR Writes

Problem: An interrupt becomes lost when trying to write/clear any of the timer status register bits (ost_sts) from the Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor in the same cycle that hardware is trying to update this register when a time-out occurs.

Implication: The second timer interrupt will be lost.

Workaround: Here are two possible software timer workarounds:

- Only enable one of the following timers: GP0, GP1, Timestamp, or Watchdog interrupt. If the watchdog timer is configured to do a soft reset, the GP0,GP1, or the Timestamp can be used in addition to the watchdog timer.

  Note there is a counter in the IXP425 network processor PMU that can be used.
- If the first work around is insufficient, an improved timer interrupt handler would be needed. Pseudo-code for such an improved timer handler follows:

1. Interrupt handler determines there is a timer interrupt.
2. Load R0 [timer status register].
3. Load R4 [timestamp register].
4. Load R5 [GP timer 0 register].
5. Load R6 [GP timer 1 register].
6. Software acknowledges timers that have expired.
7. Store R0 [timer status register] at software clears timer status.
8. Load R7 [timestamp register].
9. Load R8 [GP timer 0 register].
10. Load R9 [GP timer 1 register].
11. If (R7 < R4), then timestamp expired: Software must acknowledge.
12. If (R8 > R5), then GP timer 0 expired: Software must acknowledge.
13. If (R9 > R6), then GP timer 1 expired: Software must acknowledge.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step) — Fixed

11. Character Time-Out Interrupt Sticks Under Certain Software Timing Conditions

Problem: Character time-out interrupt doesn't clear and the DR bit is not set.

Implication: The processor can get into a continuous interrupt loop where the character time-out interrupt is SET although there is no data in the FIFO.

This errata results from the following implementation:

1. Read Line Status Register (LSR) and check for errors.
2. Read Data from FIFO.
3. Software Delay.
4. Read LSR, check for errors, and LOOP back to Step 2. — if DR bit in LSR is SET.
5. Done.

If the Step 3. is placed in front of 1., the issue never occurs.

Workaround: If this situation has been assessed correctly, the workarounds disabling of the interrupt — via IER[4] (Step 2.) — will prevent the RTO interrupt SM from being entered a second time. It is safe to re-enable the interrupt *after the FIFO is empty*, as the FIFO empty condition also prevents the RTO interrupt SM from being entered. To execute:

1. Read LSR and check for errors.
2. Disable Receiver Time-out Interrupt Enable (RTOIE) via Interrupt Enable Register (IER) Bit 4.
3. Read Data from FIFO.
4. Software Delay.
5. Read LSR, check for errors, and LOOP back to (3) if DR bit in LSR is SET.

6. No more data in FIFO: Re-enable RTOIE interrupt via IER bit 4.
7. DONE.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step) — Fixed

12. Intel® IXP425 Network Processor A-0 Stepping May Have Problems Working With Some SDRAM Devices

Problem: Although the Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor comply with the JEDEC SDRAM specification, some SDRAM manufacturers are shifting their devices' implementation of an optional section of the specification, to maintain consistency between SDRAM Single-Data-Rate (SDR) Memory and SDRAM Double-Data-Rate (DDR) Memory. That results in an issue involving the Mode Register Set command.

In order to support vendor-specific, extended modes, the SDRAM must receive the Mode Register Set command with the Bank Address (BA) bits set to a particular value. For normal operation, the BA bits must be set to logic 00, during the Mode Register Set command.

The Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor set these bits to logic 11, during the Mode Register Set Command. The JEDEC SDRAM specification states that the BA values must be put to a valid state during the Mode Register Set command. However, the DDR options to this specification require that the BA bits be set to support extended modes.

Memory known to work

- Micron*
  - — MT48LC8M16A2
  - — MT48LC16M16A2
  - — MT48LC32M16A2
- Winbond*
  - — W981216BH
  - — W982516BH
- Elpida*
  - — UPD45128163
  - — HM5225165B
  - — HM5259165B
  - — HM5257165B

Additional information may be added to this list as more data is collected.

Memory known to fail:

- Winbond — W987Z6CB
- Samsung* — K4S281633D-R

Additional information may be added to this list as more data is collected.

Implication: The Intel® IXP425 A-0 step processor does not work with some SDRAM memory.

Workaround: Use a memory device listed in the preceding section, "Memory known to work" on page 26.

Ask the memory vendor the following question:

If the BA bits are set to logic 11, during the Mode Register Set Command, will the memory work correctly?

— If answer is yes, the memory should work

— If answer is no, the memory will not work

Migrate to the next generation of the Intel® IXP4XX product line processors.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — Fixed

Status: IXP46X (A0-Step, A1-Step and A2-Step) — N/A

Status: IXP45X (A0-Step, A1-Step and A2-Step) — N/A

13. PCI DMA Lock-Up Condition

Problem: It is possible that the PCI bus can get in a locked condition, when multiple products are connected in a system and these systems are using the DMA controllers on the Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor. Lock-up may occur when the DMA controller of the IXP42X product line is setup to perform a DMA transfer, thus either issuing a series of DMA 8-word PCI reads or DMA 8-word PCI writes to a particular PCI device (Device A for example) when at the same time Device A issues standard PCI transfers (either PCI writes or PCI reads) to the IXP42X.

Up to three standard PCI transfers from device A are enqueued into the IXP42X processor's inbound transfer queue. However they are not de-queued from this inbound queue for execution in the IXP42X processor during the time one DMA 8-word PCI transfer (read or write) is pending completion on PCI bus. If Device A has not returned or accepted one DMA 8-word PCI transfer by the time Device A issues a fourth PCI transfer to the IXP42X processor, the processor will begin to retry this fourth inbound PCI transfer, as the IXP42X processor's inbound queue is now full.

This may cause a lock-up situation, for example, Device A's ordering rules may not permit reads to pass writes, thus Device A is waiting on completion of it's PCI writes (say) at the same time the IXP42X processor is waiting on completion of one DMA 8-word PCI read. Examples of device A are the Intel PCI-to-PCI 21154 Bridge, another IXP42X processor device. Other PCI devices may apply.

Lockup occurs due to time-out on PCI bus due to the deadlock occurring between the Intel PCI-to-PCI 21154 Bridge and the IXP42X processor. In this case, the Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor does a DMA 8-word PCI read, which gets retried by the bridge as the bridge fetches that 8 words of data. The bridge, asynchronously, issues a series of posted PCI writes in quick succession to the IXP42X processor filling it's inbound queue, three PCI writes are enqueued however the forth is retried and deadlock occurs.

When the DMA 8-word PCI transfer has completed and before the next DMA 8-work PCI transfer starts, the IXP42X processor will at least de-queue one entry in its inbound queue.

Implication: Deadlock condition on the PCI bus controller.

Workaround: Ensure inbound PCI transfer rate is slower than DMA 8-word PCI transfer rate. Assuming an empty inbound PCI queue to start with, guaranteeing that only one inbound PCI transfer will occur for any one DMA 8-word PCI transfer, will resolve this deadlock issue. Once a DMA 8-word PCI transfer completes at least one entry in the IXP42X processor's inbound queue will be de-queued.

For example, in the bridge case mentioned above, the workaround was to reduce the inbound PCI write rate to 1 in every 64 DMA 8-word PCI transfers. This prevented the deadlock and subsequent lock-up condition.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step) — Fixed

14. PCI Doorbell Register Lock-up Condition When Using Two Products Together that have Intel® IXP4XX Product Line of Network Processors

Problem: It is possible that the PCI bus can get in a locked condition when multiple products — using IXP4XX product line processors — are connected in a system and these systems are using the PCI doorbell registers of the IXP4XX product line processors. This lockup only occurs when both of the IXP4XX product line processors attempt to access each other's PCI doorbell register at a particular instant. This error occurs only on reads of the of the doorbell register.

Implication: When using two products using IXP4XX product line processors and their PCI doorbell registers, PCI doorbell register reads cannot be implemented.

Workaround: Only do doorbell register write from PCI bus to generate interrupt, and use regular memory to pass information.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

15. PCI Controller Returns Infinite Retries on PCI After AHB Pre-Fetch Error

Problem: If an external PCI master performs a memory read operation targeting the Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor, and the last word read is "close" to a hole in the AHB memory map, for example, if the read was close to the top of the 1 Gbyte of SDRAM space, the Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor may retry all subsequent inbound PCI transactions; this locks up the PCI target interface.

This problem occurs because inbound PCI reads are pre-fetch on the AHB bus. If the read is close to the top of a valid memory region that borders a reserved memory region, the pre-fetch read on AHB may cross into the reserved region and produce an error response on the AHB. Under certain conditions, this error condition is not cleared properly and results in retry responses to all following PCI transactions that target the Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor. The initial read completes normally on PCI.

Implication: The Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor do not support the alias of the above 256 Mbyte of SDRAM memory space.

Workaround: None.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step) — Fixed

16. UART Break Indicator

Problem: When the UART break indicator is de-asserted, it is possible for the UART to detect a start bit and receive an incorrect 0xFF byte. This 0xFF byte has no indicators set.

Implication: UART may receive an incorrect 0xFF byte and have no indicators set.

Workaround: None

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step) — Fixed

17. IRQ 3 is Locking the System 'Disable'

Problem: When performing bi-directional wire-speed bridging of 64-byte Ethernet packets — between both NPE Ethernet ports using MontaVista* Linux Support Package (LSP) 3.0 and System Test code to do the bridging — the IRQ for IxQMgr interrupts can be disabled by the kernel when it detects the occurrence of more than 100,000 IxQMgr interrupts. This symptom occurs when the interrupt source does not get cleared in time before the next interrupt occurs — causing the interrupt to constantly trigger and overload the CPU with fake interrupt requests and "lock the system."

Implication: The system gets locked because the IRQ is not getting cleared in time before the next interrupt occurs.

Workaround: Implement the following software routine to enforce a write completion by reading the very same memory mapped register and forcing a data-dependency stall:

```
mov r0, #regloc

str r1, [r0] @ initiate a write operation

ldr r1, [r0] @ read back: this will flush the write

mov r1,r1 @ stall: ensure the read is complete
```

Note: This workaround requires Intel XScale processor cycles; so it should be done carefully.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

18. EX_IOWAIT_N Timing

Problem: There are two problems with the functionality of the expansion bus IOWAIT protocol. If T2 and T3 are both programmed to be 0 (normal timing), the expansion bus controller will not extend the T3 data state as described in Figure 60 "Expansion Bus I/O Wait Operation" on page 303, of the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* (252480-002). This occurs because there is a synchronizer on the EX_IOWAIT_N signal that causes the expansion bus controller to transition to the T4 state before EX_IOWAIT_N is detected.

Additionally, the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* states that the expansion bus controller will transition to the T4 state upon the de-assertion of EX_IOWAIT_N. The expansion bus controller does not do this — instead waiting for the T3 count to expire before proceeding to T4. This issue also affects HRDY signal for HPI mode.

Implication: The expansion bus will not extend the T3 data state as shown in Figure 60 of the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual*(252480-002).

Workaround: To avoid unexpected timing issues, T2 or T3 must be programmed to non-zero values and assurances made that EX_IOWAIT_N is asserted at least three cycles before the deasserting edge of EX_RD_N. Additionally, the extended wait states will not be changed after the deassertion of EX_IOWAIT_N.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

19. SOF During Control Read Can Corrupt USB Transfer

Problem: An SOF packet sent in between the setup and data stage of a control read transfer is decoded as an OUT token by the UDC core. This causes the command state machine to prematurely transit to the status stage.

Implication: When a USB host issues an IN token for the data stage, the UDC responds with a null data packet instead of the requested data bytes. This is because the UDC core has transitioned to status stage and assumes that this is a status in transaction.

Workaround: None.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — Fixed

Status: IXP46X (A0-Step, A1-Step and A2-Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step) — Fixed

20. USB - Invalid 0x81 Value in Endpoint 0 Control Register on SETUP Transfer

Problem: After the status-OUT stage of an USB standard Control Read Command — such as GET_DESCRIPTOR, GET_INTERFACE, and GET_STATUS — if the UDCCS0(OPR) is not cleared by the users before the next SETUP packet is received, the UDCCS0 could contain an invalid value.

The invalid value is UDCCS0 = 0x81, which indicates that a SETUP packet was received, but the UDDR0 Data FIFO is empty, however, the SETUP packet data is actually in the UDDR0 Data FIFO.

Implication: Software can get confused if the status register indicates that a SETUP packet was received (UDCCS0[SA]=1), an OUT packet is ready (UDCCS0[OPR]=1), but the UDDR0 Data FIFO is empty (UDCCS0[RNE]=0).

Workaround: Software should treat UDCCS0 = 0x81 as a valid value and read 8 bytes from the UDDR0 Data FIFO while ignoring UDCCS0[RNE]. This 8 bytes of data will be the correct data from the SETUP Command.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

21. Ethernet Coprocessors — Ethernet Pad Enable Overrides Append FCS

Problem: The IXP4XX product line processors have an Ethernet coprocessor that is configured by the Intel® IXP400 Software. (Some IXP4XX product line processors have two Ethernet coprocessors.) The coprocessor can be programmed via the Ethernet Transmit Control Registers to either append or not append the FCS on the transmitted Ethernet frames. When the frame payload size is less than 60 bytes, the Pad Enable control bit has priority over the Append FCS control bit on whether or not the FCS is appended on a frame.

Implication: When the frame payload size is less than 60 bytes, the FCS will be appended to the Transmit frames even though the Append FCS control bit is *not* set because the Pad Enable control bit overrides the Append FCS control bit.

Workaround: None.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

22. Ethernet Coprocessors — Length Errors on Received Frames

Workaround: The IXP4XX product line processors have an Ethernet coprocessor that is configured by the Intel® IXP400 Software. (Some IXP4XX product line processors have two Ethernet

coprocessors.) The Ethernet coprocessor can indicate length error on received frames only when stripping of pad bytes from the received frame is enabled.

Implication: Length errors on received frames when pad stripping is disabled will not be indicated to the NPE software when it reads the Receive status. When pad stripping is enabled, length error indicates that the packet length is not equal to 64 bytes, and the entry in the length field is less than 46, but not zero.

Workaround: None.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

23. PCI DC Parameter VIH Marginality Issue

Problem: The input-high voltage (VIH) for the PCI bus signals does not meet the documented specification. In the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (252479), Table 25 (“PCI DC Parameters”) specifies the VIH minimum value as 0.5 V_{CCP}. This specification is changed to 0.6 V_{CCP}.

Implication: At 66-MHz PCI bus operation, the Tprop timing would be a slightly longer. Refer to the PCI Local Bus Specification, Revision 2.2, and see the section “System Timing Budget.”

Workaround: To ensure proper PCI bus operation at 66 MHz, designers must pay careful attention to the maximum trace length and loading. Board simulation should be done prior to finalizing layout. For PCI topologies and routing recommendations, see the *Intel® IXP4XX Product Line and IXC1100 Control Plane Network Processors Hardware Design Guidelines* (252817).

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step) — Fixed

24. False PCI DMA Completion Notification Causing Data Corruption

Problem: The PADC1, PADC0, APDC1, and APDC0 complete bits in the PCI_DMACTRL register will not be cleared under certain conditions when the Intel XScale processor performs a write 1 to clear to the appropriate bit. If another PCI DMA transfer is initiated after the clear to the PCI_DMACTRL register, an indication of complete will occur before the DMA transfer has been finished (because the complete bit may have not been cleared).

Implication: DMA data will not be transferred as programmed in the PCI DMA registers.

Workaround: There are two workarounds available:

- Mask the PADC/APDC enables in the PCI_INTEN register and use software to poll the EN (bit 31) of the appropriate PCI_ATPDMA0_LENGTH, PCI_PTADMA0_LENGTH and PCI_ATPDMA1_LENGTH, PCI_PTADMA1_LENGTH register to indicate whether the DMA transfer was completed.
- If interrupts are preferred, after writing a 1 to clear the appropriate complete bit in the PCI_DMACTRL register, read the PCI_DMACTRL register back and ensure the appropriate complete bit was cleared. If not cleared, repeat this step until the appropriate complete bit is cleared.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

25. Expansion Bus HPI Interface Potential for Contention on Reads with T4=0

Problem: If any HPI slave on the expansion bus has T4 configured to 0, there is potential for contention on the data during a read. The HPI specification states that it stops driving data a maximum of 10 ns after the deassertion of DS (which is ex_wr_n). The IXP4XX product line processors turns on the output enable in the T4 state, which is the same cycle where ex_wr_n gets deasserted, so there is no turnaround cycle.

Implication: With T4 configured to 0 on any HPI slave in the EXP_TIMING_CS register there could be contention on EX_DATA for up to 10 ns during a read.

Workaround: The appropriate EXP_TIMING_CS register (each CS with HPI) must have T4 configured to a non-zero value, which will extend the T4 state for at least one cycle and eliminate the possibility of contention on EX_DATA.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

26. PCI Hangs With a Multiple Inbound Error Condition

Problem: The PCI controller may lock up if there are multiple errors occurring around two different inbound PCI transactions. When an inbound PCI read that targets an internal slave such as the expansion bus, or Queue Manager, results in an AHB error which occurs due to the PCI controller generating an illegal AHB transfer type on the target, and a second inbound PCI transfer is started while the first PCI read is still pending and the second PCI transfer detects a PCI address or data parity error a lock-up will occur.

Implication: The PCI controller will continue to retry all inbound transactions, and the PCI bus will lock up.

Workaround: When the PCI controller has an AHB error logged (PCI_ISR.AHBE = 1), a PCI parity error logged (PCI_SRCR.DPE = 1), and the PCI controller retries every inbound transaction, the system board must reset the IXP4XX product line processors.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

27. PCI RCOMP Operation if PCI Clock Stops

Problem: The PCI specification states that PCI_CLKIN can be any frequency from 0 to 66 MHz and can change in frequency at any time. The PCI_CLKIN frequency *cannot* be changed on the fly on the IXP42X product line as the AC timing specifications can not be guaranteed.

Implication: The IXP42X processors do not support switching between 33 and 66 MHz on the fly because the AC timing specifications cannot be guaranteed.

- Never drive PCI_CLKIN < 1 MHz or the PCI AC timings/slew rates will exceed the specification.
- If performing a PCI software reset, wait at least 2 ms after the deassertion of software reset before using the PCI interface.
- To switch between 33 MHz and 66 MHz PCI operation, and to guarantee specified AC timings, the IXP42X processor must go into reset first, and then change the PCI CLKIN by pulling up or pulling down the EX_ADDR[4] pin.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

28. UART — Break Condition Asserted Too Early if Two Stop Bits are Used

Problem: The break condition is asserted after the time of the first stop bit, even if two stop bits are used.

Implication: In the following scenario, a break condition will be raised on valid data:

1. A byte consisting only of zeros is received.
2. The first stop bit sampling is missed, and only the second one is sampled.

Workaround: Don't use two stop bits.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step)) — No Fix

29. Ethernet Coprocessors — Address Filtering Logic Ignores the Second to Last Nibble of the Destination Address

Problem: The Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor has an Ethernet coprocessor configured by the Intel® IXP400 Software. Some IXP42X processors have two Ethernet coprocessors. The Ethernet coprocessor logic ignores the second last nibble of the destination address regardless of the packet type (unicast, multicast, broadcast), that is, Destination Address: 11 22 33 44 55 x6. The reason it is the second last is that the address is transmitted on the line with the high nibble first and then the low nibble.

Implication: Some Ethernet frames with the wrong destination address can get through the Address Filter.

Workaround: A software workaround is possible using the ixEthDb filtering capabilities.

Status: No Fix. See the "Summary Table of Changes" on page 8.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

30. USB DC Parameter Vih Specification Change

Problem: The Vih levels for the IXP4XX product line processors' USB device interface do not meet the USB-1.1 industry specification of 2.0 V minimum. Under certain board-level and worst-case conditions, Vih can be a minimum of 2.15 V. The IXP4XX product line processors are therefore not fully compliant to the USB-1.1 Specification.

Implication: Issues related to this should not be noticeable. However, signal integrity issues could occur and board-level designs should be simulated.

Workaround: The revised USB Vih specification is 2.15 V minimum. Board-level simulations are recommended and based upon those simulations, correct external circuitry should be put in place if full USB compliance is required.

Affected Docs: IXP42X — *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (252479-004);
IXP45X/IXP46X — *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Datasheet* (306261-002)

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

31. Start of DMA Operation Close to Start of Non-Prefetch (NP) Operation Can Hang NP Operation on AHB

Problem: If a DMA operation is initiated in close proximity to an outbound NP operation being initiated, the NP operation may not terminate properly within the PCI Controller, resulting in no further AHB operations being accepted by the controller (retries issued). This errata only occurs if two different masters start an NP and DMA operation.

Implication: If the NP operation is sent to the PCI Core for execution on the PCI bus, but the DMA engine starts as well, the DMA should wait until the NP operation is done before proceeding. The active DMA engine blocks the “cycle complete” indication from the PCI for the NP operation, which leaves the AHB target interface hung since no other AHB operations can be accepted until the NP operation completes (retries issued).

Workaround: Only use PCI DMA/Non-prefetch PCI operations. Additionally, if the processor is starting a PCI DMA transfer, it must always write to the PCI_NP_AD register before accessing the PCI_NP_CBE or PCI_NP_WDATA. A dummy CSR read could be performed before the NP operation is initiated. This would separate the NP op initiation from a previously started DMA

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step) — Fixed

32. PCI Accesses to the Queue Manager During Queue and SRAM Mode

Problem: Under certain data traffic, the PCI controller may generate spurious write transfers and may return incorrect data on reads when accessing the Queue Manager in SRAM mode. Additionally, if the Queue Manager is being used in the Queue mode, PCI accesses must not use memory-mapped registers BAR0-3 since these accesses cause pre-fetches during reads.

Implication: Pre-fetches will cause queue data to be lost.

Workaround: Do not use the Queue Manager's SRAM mode during PCI accesses. Instead use the SDRAM memory space when generating PCI accesses to the IXP42X processor memory space. An external PCI master must use PCI BAR5 when accessing the Queue Manager when in Queue mode.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

33. Ethernet MACs Detect Late Collision Earlier than Ethernet 802.3 Specifications

Problem: On an improperly designed network, when a collision occurs on the threshold of the smallest valid Ethernet frame, it is detected as a late collision rather than an early collision.

Implication: The collided frame will not be retried up to the programmed retry count and will be dropped.

Workaround: Cable lengths, number of repeaters, and other parameters that affect the network design must be planned to not operate on the boundary of the Ethernet specifications.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

34. Read of PCI Controllers BAR 32'h XXFF_FFFC Rd[N] Corrupts Subsequent Rd[N+1]

Problem: If specifically reading the 'last word' address of a BAR register, read(n), and if that BAR register is set up adjacent to undefined memory space (that is, not adjacent to another BAR register), this read(n) will complete correctly, but will cause data corruption in the subsequent read(n+1).

Implication: Upon the next subsequent external PCI master read Rd[N+1], the PCI controller returns incorrect read data of Rd[N].

Workaround: Avoid reading the last word of the BAR or avoid reading this one BAR entirely. The setup could be changed such that the BAR registers are adjacent to each other in memory space and place the config BAR 4 on top of the final BAR so that no "last word" address in each memory address BAR0-3 is adjacent to undefined memory space. For example:

BAR4 - config BAR: highest address
BAR3
BAR2
BAR1
BAR0 - lowest address

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

35. UART Operating in Non-FIFO Mode Can Falsely Receive Overrun Error

Problem: If the UART is operating in non-FIFO mode, there is a small possibility under certain conditions of falsely receiving an Overrun Error even though an over run did not occur.

Implication: An erroneous interrupt to the Intel XScale processor may occur indicating that data was lost in the UART.

Workaround: Use the UART in FIFO mode or in non-FIFO mode, the data should be rejected and resent

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

36. USB-Specification Noncompliance for Rise/Fall Transition Times

Problem: The USB host and USB device interfaces do not meet the USB-1.1 specifications for the rise and fall transition times. For full-speed operation, the values from the USB specification state that the rise/fall time values should be in a range of 4.0 ns to 20 ns. The IXP4XX product line processors will actually produce the rise/fall time for full-speed operation in the range from 3.3 ns to 5.4 ns. For low-speed operation, the values from the USB specification state that the rise/fall time values should be in a range of 75 ns to 300 ns. The IXP4XX product line processors will actually produce the rise/fall time for low-speed operation in the range from 42 ns to 289 ns.

Implication: Issues related to this shouldn't be noticeable. However, signal integrity issues could occur and should be simulated.

Workaround: The revised rise/fall transition times range from 3.0 ns to 20 ns for full-speed operation and from 40 ns to 300 ns for low-speed operation. Board-level simulations are recommended and based upon those simulations, correct external circuitry should be

put in place to slow the transition times into the appropriate window if full USB compliance is required.

Affected Docs: IXP42X — *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (252479-004); IXP46X — *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Datasheet* (306261-002)

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

37. Ethernet MAC Does Not Detect Transmit FIFO Underruns Reliably

Problem: There is a transmit race condition in the 10/100 Ethernet MAC. The race condition happens when the 10/100 Ethernet MACs transmit FIFO is filled on the same transmit clock as the transmit FIFO under-run occurs on the line side. If the race condition were to happen, the 10/100 MAC may end up appending four more bytes of indeterminate data to the Ethernet frame which is transmitted on the line. The FCS will be generated on the frame that includes the four extra bytes so the receiver will not detect any problem with the Frame.

There are currently no customers who have experienced this problem or a problem with similar symptoms with years of testing and millions of units productized and deployed.

Implication: This problem can potentially occur in situations of intense NPE load (VLAN/QoS, fire wall, header conversion) resulting in corrupt frames to be transmitted. This problem is detectable in applications running any upper-layer protocol. For example, Layer 3/4 in the network application stack will detect this problem (because of IP checksum error) and drop the packet before it reaches the application. This error is undetectable by the receiver.

Workaround: None

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

38. SMII late_col Occurs Earlier than Expected

Problem: When configured for SMII mode, a collision during Byte 59 of the transmitted packet will be detected as a “late collision” by the ECP coprocessor. This should not be declared until after Byte 64, according to the MII IEEE 802.3 specification.

Implication: The result of a smaller detection window is that collisions occurring on small, but valid 64-byte packets on a long half-duplex line will be undetected by an endpoint on the same ethernet segment, as this MAC device will not see the collision, and it will not know that it must retry the packet. At the pins of the IXP46X network processors, the collision may have happened in time, but the collision signal will be ignored by the time it passes through the SerDes of the PHY and into the IXP46X network processor MAC.

When utilizing the SMII interface in half-duplex segments directly connected to the SMII PHY, the cable length will be limited to less than the IEEE 802.3-specified maximum length. This length is will be marginally smaller as determined by the round time propagation delay based upon five bit times less than the specified 64 bits. This can be determined by looking at the IEEE 802.3 specification. This length will not be much smaller.

Workaround: When utilizing the SMII interface in half-duplex segments directly connected to the SMII PHY, the cable length will have to be limited to less than the IEEE 802.3-specified maximum length. This length is will be marginally smaller as determined by the round time propagation delay based upon five bit times less than the specified 64 bits. This can be determined by looking at the IEEE 802.3 specification. This length will have to only be marginally smaller.

Status: IXP425 (A0-Step) — N/A

Status: IXP42X (B0-Step and B1-Step) — N/A

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

39. Timer Issues with Prescale Programming Sequence and Pause/Resume Operation

Problem: There are two timer problems, each with unique implications and workarounds:

Problem #1 Description:

When the 3/4 clock (20-ns clock from a 66.667-MHz clock) is enabled and a prescale value (P) and a reload value (R) are used as a combination to get a desirable interrupt sequence, the timer might be too fast.

Problem #2 Description:

The pausing of a timer works properly — for example, pausing Timer 0 by setting the tim0_cnt_en bit in the ost_tim0_cfg register. However, when clearing that bit, the timer does not resume counting.

Implication: **Problem #1 Implication:**

If the P or R values or both values are divisible by 3, the timer works as specified. If the values are *not* divisible by 3, the timer will roll over erroneously and produce a faster-than-expected count. When the timer expires and rolls over, it fails to check if the 20-ns clock is enabled which can make the counter roll over when it should not have.

Problem #2 Implication:

The prescale counter must be refreshed by writing the prescale value.

Workaround: **Problem #1 Workaround:**

If an even-numbered, on-going, accurately timed interrupt is needed (for example, an interrupt every microsecond), it can be obtained in at least the following two ways:

- Use the 15-ns mode and a P or R value that is divisible by three.
  For example, the timer can accurately create an interrupt every 3 µs or 9 µs.
- If the first approach won't work — If an interrupt is needed every 1 µs, for example, and the P and R values cannot be divisible by 3:
  Simulate the desired behavior by having the timer use the 15-ns clock and divide by 67 for two out of every three interrupts and divide by 66 for the other, one out of three interrupts. This creates interrupts after 1.005 µs, 1.005 µs, and 0.990 µs. This adds up to exactly 3.0 µs, so time is neither gained nor lost.
  This approach works because only the least-significant bits (LSBs) in the configuration register are written. (The LSBs are loaded into the timer only when the timer rolls over.) This would *not* work if the Reload Register was written because the reload value is immediately loaded into the timer. (The Reload Register does not wait for the timer to rollover.)
  This approach change be used to obtain 10-µs or 1-ms interrupts, for example, by writing the appropriate value to the prescale register.

Example workaround routines — to simulate 1-µs, 10-µs, 1-ms ticks — follow:

Routine for 1-µs tick

1. Timer CFG <- 0x02 — Set up timer.
2. Timer PRE <- 0x00 — No prescale clock is 15 ns.
   Timer RL <- 0x41 — Starts the timer and the reload value is 0x41(R=67)
   loop_1us
3. Wait for interrupt.

4. Timer STS <- 0x01 — Clear the interrupt.
   Timer CFG <- 0x01 — Changes the reload R to be 66.
5. Wait for interrupt.
6. Timer STS <- 0x01 — Clear the interrupt.
   Timer CFG <- 0x02 — Changes the reload R to be 67.
7. Wait for Interrupt.
   Timer STS <- 0x01 — Clear the interrupt.
8. end_loop_1us

Routine for 10-µs tick

1. Timer CFG <- 0x02 — Set up timer.
   Timer PRE <- 0x10 — Prescale P=10.
   Timer RL <- 0x41 — Starts the timer and the reload value is 0x41=67).
   loop_10us
2. Wait for interrupt.
3. Timer STS <- 0x01 — Clear the interrupt.
   Timer CFG <- 0x01 — Changes the reload to be 66.
4. Wait for interrupt.
5. Timer STS <- 0x01 — Clear the interrupt.
   Timer CFG <- 0x02 — Changes the reload to be 67.
6. Wait for interrupt.
7. Timer STS <- 0x01 — Clear the interrupt.
   end_loop_10us

Routine for 1-ms tick

1. Timer CFG <- 0x02 — Set up timer.
   Timer PRE <- 0x3e7 — Prescale P = 1000.
   Timer RL <- 0x41 — Starts the timer and the reload value is 0x17(R=67).
   loop_1ms
2. Wait for interrupt.
3. Timer STS <- 0x01 — Clear the interrupt.
   Timer CFG <- 0x01 changes the reload to be 66.
4. Wait for interrupt.
5. Timer STS <- 0x01 — Clear the interrupt.
   Timer CFG <- 0x02 — Changes the reload to be 67.
6. Wait for interrupt.
7. Timer STS <- 0x01 — Clear the interrupt.
   end_loop_1ms

Problem #2 Workaround:

Example workaround routines for enabling a timer, pausing it, and enabling the counting again follow:

- To enable timer0 and then pause it:

1. Timer CFG <- 0x07 — Set ups up timer.
   Timer PRE <- 0x19 — Loads the prescaler (P=26).
   Timer RL <- 0x15 — Starts the timer, and the reload value is 0x17(R=24).
2. If it is desirable to pause the timer, then use the tim0_cnt_en to halt counting.

Timer CFG <- 0x0F — Pauses timer.

- To enable the counting again:

1. Timer CFG <- 0x07 — Enables the timer again.
2. Timer PRE <- 0x19 — Refresh the pre scaler.
   The counter will continue from where it was paused.

Status: IXP425 (A0-Step) — N/A

Status: IXP42X (B0-Step and B1-Step) — N/A

Status: IXP46X (A0-Step) — No Fix

Status: IXP46X (A1-Step and A-2 Step) — Fixed

Status: IXP45X (A0-Step) — No Fix

Status: IXP45X (A1-Step and A-2 Step) — Fixed

40. IEEE-1588 Time Sync Lock-up Fails to Time-Stamp a Second PTP Message

Problem: A lock-up condition has been identified in the IEEE-1588 Time Sync block that prevents all subsequent messages form being time-stamped. It occurs when a message has been time-stamped and the registers are locked, after which — if a new message comes in before the previous time stamp is read and the lock bit is cleared — the Time Sync block enters a lock-up condition and prevents all furthers messages from being time-stamped. This occurs for both received and transmitted messages.

Implication: Under very typical usage scenarios, the IEEE-1588 unit can lock up and will not capture time stamps after the initial one is locked.

Workaround: If IEEE-1588 functionality is required, it must be implemented with external hardware.

Status: IXP425 (A0-Step) — N/A

Status: IXP42X (B0-Step and B1-Step) — N/A

Status: IXP46X (A0-Step) — No Fix

Status: IXP46X (A1-Step and A-2 Step) — Fixed

Status: IXP45X (A0-Step, A1-Step and A2-Step) — N/A

41. SSP Synchronous Issue When Using External SSP Clock

Problem: When the external SSP clock is used, the A-0 and A-1 stepping of the Intel® IXP45X/ IXP46X Product Line of Network Processors can enter a condition whereby the received data on SSP cannot be registered correctly due to possible synchronous issue on data and control signals at SSP clock synchronizer output. The data must be valid before assertion of the write enable signal to ensure that the correct data is being captured.

Implication: When an external SSP clock is used, there is a possibility that the received data is incorrectly captured into the RXFIFO, which will cause an incorrect data read. A data error will also occur when data is sent out from the TXFIFO to an external device.

Workaround: Customer may use internal SSP clock, SSP_CLK to prevent the occurrence of synchronous issue.

Status: IXP425 (A0-Step) — N/A

Status: IXP42X (B0-Step and B1-Step) — N/A

Status: IXP46X (A0-Step and A1-Step) — No Fix

Status: IXP46X (A2-Step) — Fixed

Status: IXP45X (A0-Step and A1-Step) — No Fix

Status: IXP45X (A2-Step) — Fixed

42. Potential Lockup Condition in The Memory Controller Unit

Problem: The A-0 and A-1 stepping of the Intel® IXP45X/IXP46X Product Line of Network Processors can enter a condition whereby the system locks up when certain combinations of memory accesses occur *nearly simultaneously*.

After more extensive investigation, it has been identified that the lockup condition can be caused by near-simultaneous accesses to 1 or 2 cache lines. The accesses to the cache lines must be performed by the Intel XScale processor, South AHB, and/or North AHB for the lockup to happen.

Workaround: The A-2 stepping of the IXP45X/46X Product Line of Network Processors removes the potential for a lockup condition.

Status: IXP425 (A0-Step) — N/A

Status: IXP42X (B0-Step and B1-Step) — N/A

Status: IXP46X (A0-Step and A1-Step) — No Fix

Status: IXP46X (A2-Step) — Fixed

Status: IXP45X (A0-Step and A1-Step) — No Fix

Status: IXP45X (A2-Step) — Fix

43. HDLC Coprocessor is Unable to Capture Alignment Error On a Specific Frame Pattern

Problem: Based on ITU-T Q.921, during transmission, a bit stuffing ‘0’ is inserted after five consecutive ‘1’ bits (including the FCS field) and then removed at the receiving end. The bit stuffing is used to ensure that data does not appear as the ‘end of frame’ flag (01111110). If five ‘1’s are received without a bit stuffing ‘0’, a ‘byte alignment error’ will be issued. The IXP4XX HDLC controller will not issue a ‘byte alignment error’ when the following boundary conditions occur:

1. The receiving frame ends with five ‘1’s (FCS) followed by an ‘end of frame’ flag: “xx011111 01111110”

   AND

2. The ‘end of frame’ flag received is byte aligned.

Implication: An HDLC frame received by IXP4XX HDLC controller from a Q.921 or ISO-3309 compliant HDLC controller with the same pattern as described above will not generate a ‘byte alignment error’; however, as this condition can only occur if a bit toggles on the line due to noise and so on, an FCS error will be issued by the HDLC controller.

Note: Refer to Section 4.1 of the *IXP4XX Developer’s Manual* or “Update on HDLC Coprocessor” on page 53 for more details on the HDLC coprocessor supported features list.

Workaround: None

Status: IXP42X (A0-Step, B0-Step and B1-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step, A2-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step, A2-Step) — No Fix

44. B-1 Stepping Accommodates Future SDRAMs

Problem: Intel is anticipating a change in technology and manufacturing processes by SDRAM vendors in the near future. These changes may place stricter tolerances on the operation and timing of the individual devices. As a result, the current IXP42X product line of network processors may not interoperate well with the new SDRAMs coming out in the market.

Workaround: Anticipating change in the SDRAM technology, Intel has dash stepped the IXP42X product line of network processors for interoperability with 133 MHz speed grade SDRAMs that have a minimum refresh time of 67.5 ns or less.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step) — No Fix

Status: IXP42X (B1-Step) - Fixed

Status: IXP46X (A0-Step, A1-Step and A2-Step) — N/A

Status: IXP45X (A0-Step, A1-Step and A2-Step) — N/A

5.0 Intel XScale® Technology Errata Descriptions

1. Abort is Missed When Lock Command is Outstanding

Problem: A bus abort occurs on a code fetch while an instruction TLB or I-Cache lock *Move to Coprocessor from* Intel XScale processor *processor Register* (MCR) command is outstanding. The processor fails to abort and instead executes the instruction returned on the aborting transaction. Parity errors are not affected. The bus abort may be due to an abort pin assertion.

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

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

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

Problem: When there is an aborted store that hits clean data in the data cache (data in an aligned 4-word range that has not been modified from the processor since it was last loaded from memory or cleaned), the data in the array is not modified (the store is blocked), but the "dirty" bit is set. When the line is then aged out of the data cache or explicitly cleaned, the data in that four-word range is evicted to external memory, even though it has never been changed. In normal operation this is nothing more than an extra store on the bus that writes the same data to memory that is already there.

The boundary condition where this might occur:

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

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

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

3. Performance Monitor Unit Event 0x1 Can Be Incremented Erroneously by Unrelated Events

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

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

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

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

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

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

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

4. In Special Debug State, Back-to-Back Memory Operations — Where the First Instruction Aborts — May Cause a Hang

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

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

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

5. Accesses to the CP15 ID Register with Opcode2 > 0b001 Returns Unpredictable Values

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

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

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

Workaround: No workaround.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

6. Disabling and Re-Enabling the MMU Can Hang the processor or Cause it to Execute the Wrong Code

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

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

```
@ The following code sequence takes r0 as a parameter. The value of r0 will be

@written to the CP15 control register to either enable or disable the MMU.

mcr p15, 0, r0, c10, c4, 1 @ unlock I-TLB

mcr p15, 0, r0, c8, c5, 0 @ invalidate I-TLB

mrc p15, 0, r0, c2, c0, 0 @ CPWAIT

mov r0, r0

sub pc, pc, #4

b 1f @ branch to aligned code

.align 5

1:

mcr p15, 0, r0, c1, c0, 0 @ enable/disable MMU, caches

mrc p15, 0, r0, c2, c0, 0 @ CPWAIT

mov r0, r0

sub pc, pc, #4
```

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

7. Updating the JTAG Parallel Registers Requires an Extra TCK Rising Edge

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

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

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

8. Non-Branch Instruction in Vector Table May Execute Twice After a Thumb Mode Exception

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

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

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

9. Intel XScale® processor Non-Branch Instruction in Vector Table

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

Implication: Instruction may be executed twice if an exception occurs in thumb mode and if it is a non-branch instruction.

Workaround: If a no-op is placed at the beginning of the FIQ handler, the no-op will execute twice and no incorrect behavior will result. If a branch instruction is placed at the beginning of the handler, it will not be executed twice.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step and B1-Step) — No Fix

Status: IXP46X (A0-Step, A1-Step and A2-Step) — No Fix

Status: IXP45X (A0-Step, A1-Step and A2-Step) — No Fix

6.0 Specification Changes

1. IXC1100 Control Plane Processors Discontinued

Issue: Shipments of IXC1100 Control Plane Processor are discontinued as of October 15, 2005. Therefore, all references to the IXC1100 Product part number and ordering information contained in all IXP42X-related collateral must be removed. Part numbers and ordering information associated with the IXC1100 have also been removed from this Specification Update.

2. Update ProdRevID in DevManual for IXP42X B-1

Issue: The IXP42X B-1 stepping comes with a revised Product Revision ID. This Product Revision ID is located in bits 3:0 of Register 0 of CP15. Table 10 is changed to reflect the new B-1 stepping Product revision ID.

FROM

Table 10

12:10	Read / Write Ignored	Intel XScale processor Revision: This field reflects revisions of core generations. Differences may include errata that dictate different operating conditions, software work-around, etc. Value returned will be 000b
9:4	Read / Write Ignored	Product Number for: IXP42X 533-MHz processor - 011100b IXP42X 400-MHz processor - 011101b IXP42X 266-MHz processor - 011111b
3:0	Read / Write Ignored	Product Revision for:IXP42X product line IXP42X 533-MHz processor - 0001b IXP42X 400-MHz processor - 0001b IXP42X 266-MHz processor - 0001b

TO

Table 10

12:10	Read / Write Ignored	Intel XScale processor Revision: This field reflects revisions of core generations. Differences may include errata that dictate different operating conditions, software work-around, etc. Value returned will be 000b
9:4	Read / Write Ignored	Product Number for: IXP42X 533-MHz processor - 011100b IXP42X 400-MHz processor - 011101b IXP42X 266-MHz processor - 011111b
3:0	Read / Write Ignored	Product Revision for:IXP42X product line IXP42X 533-MHz processor - 0001b for B-0, 0010b for B-1 IXP42X 400-MHz processor - 0001b for B-0, 0010b for B-1 IXP42X 266-MHz processor - 0001b for B-0, 0010b for B-1

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual*

3. External Crystal Support is No Longer Supported with IXP42X

Issue: External crystal support for the IXP42X processor oscillator clock input is no longer supported as of April 2005, identical to the non-support of crystals for the Intel® IXP46X Product Line of Network Processors. An oscillator must be used for the system clock.

7.0 Specification Clarifications

1. Impedance Recommendation Should be Consistent at 50 Ω

Issue: The *Intel® IXDP425 / IXCDP1100 Development Platform User's Guide* specifies a 65 Ω impedance requirement in two places, both of which should be changed to 50 Ω as follows:

In Appendix A.1 (General Guidelines), change impedance:

— **FROM:** Target impedance 65 Ω on all signal layers

— **TO:** Target impedance 50 Ω on all signal layers

In Appendix A.5 (PCI), change impedance:

— **FROM:** Characteristic impedance on compactPCI signal traces must be 65 Ω

— **TO:** Characteristic impedance on compactPCI signal traces must be 50 Ω

Affected Docs: *Intel® IXDP425 / IXCDP1100 Development Platform User's Guide* (Sept. 2004)

2. RJ11 Connector Erroneously Connected to RS-232 Block

Issue: On three block diagrams, RJ11 is erroneously connected to RS-232 Transceiver Block. The following diagrams must be modified as described below:

- **Figure 1** of Intel® IXP45X and Intel® IXP46X Product Line of Network Processors*: Migrating from the Intel® IXP42X Product Line of Network Processors Application Note* (May 2005). Replace the RJ11 block connection to the RS-232B XCVR block with a DB9 block.

Affected Docs: Intel® IXP45X and Intel® IXP46X Product Line of Network Processors*: Migrating from the Intel® IXP42X Product Line of Network Processors Application Note* (May 2005)

3. IXP42X using Compact Flash Application Note: Connector Recommendations

Issue: In the December 2004 release of the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor: Using CompactFlash Application Note,* bullets for REG# and WE# incorrectly state that the pins should be connected to the ground. The schematics in the Figure 3 are correct. As per the Compact Flash specification in page 28 (REG) and 29 (WE), REG# and WE# should be driven high and connected to VCC when they are unused. Update the bullets for REG# and WE# on page 12 of the application note as mentioned below:

FROM:

- REG# – The Register Select line from the CF card is not used in True IDE mode and connected to Ground.
- WE# – The Write Enable line from the CF card is not used in True IDE mode and connected to Ground.

TO:

- REG# – The Register Select line from the CF card is not used in True IDE mode and connected to 3.3 V.
- WE# – The Write Enable line from the CF card is not used in True IDE mode and connected to 3.3 V.

Affected Docs: Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor*: Using CompactFlash Application Note (December 2004).*

4. IXDP425 User's Guide: Expansion Bus Configuration Straps

Issue: In the September 2004 release of the *Intel® IXDP425 / IXCDP1100 Development Platform User's Guide*, the expansion bus configuration straps [16:5] are shown as used. According to the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual*, these are Reserved. The User's Guide should be updated to match the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual*, which will make some of these signals as reserved. On page 29, update the Table 9 as follows:

FROM:

Bit	Name	Description
[20:18]	AVAILABLE	
[17:16]	HSS0_MODE[1:0]	Identifies the HSS-0 card type '00' – No card or unidentified card These bits can be used per the users desire when building own cards
[15:13]	HSS1_MODE[2:0]	Identifies the HSS-1 card type and number of cards '000' – No card or unidentified card These bits can be used per the users desire when building own cards.
[12:11]	MII1_MODE[1:0]	Identifies the MII-1 card type '00' – No card or unidentified card '01' – LXT972A These bits can be used per the users desire when building own cards.
[10:9]	MII0_MODE[1:0]	Identifies the MII-0 card type '00' – No card or unidentified card '01' – LXT972A These bits can be used per the users desire when building own cards
[8:6]	DSL_MODE[2:0]	Identifies the DSL card type '000' – No card or unidentified card '001' – Alcatel* 20150 ADSL Annex A These bits can be used per the users desire when building own cards.
[5]	RESERVED	No connection may be made – Internal only strap.

TO:

Bit	Name	Description
[20:17]	AVAILABLE	
[16:5]	RESERVED	No connection may be made – Internal only strap.

Affected Docs: *Intel® IXDP425 / IXCDP1100 Development Platform User's Guide (September 2004)*.

5. IXP42X / IXP46X Developer's Manual - UART Baud Settings Table

Issue: In Table 129 of the September 2006 release of the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual*, where Baud Rate = 9600, the Baud Rate Generator Clock states 115.2 KHz. This should be changed to 153.6 KHz.

In Table 244 of the August 2006 IXP46X Developer's Manual, where Baud Rate = 9600, the Divisor Latch Low Register states 0x80 and this should be changed to a value of 0x60. The Divisor Hexadecimal states 0x0080 and this should be changed to a value of 0x0060. The Baud Rate Generator Clock states 115.2 KHz and this should be changed to a value of 153.6 KHz.

FROM

Table 129

Divisor Latch High Register	Divisor Latch Low Register	Divisor		Baud Rate Generator Clock Output	Baud Rate
		Hexadecimal	Decimal		
0x00	0x01	0x0001	1	14.7456 MHz	921,600
0x00	0x02	0x0002	2	7.3728 MHz	460,800
0x00	0x04	0x0004	4	3.6864 MHz	230,400
0x00	0x08	0x0008	8	1.8432 MHz	115,200
0x00	0x10	0x0010	16	921.6 KHz	57,600
0x00	0x20	0x0020	32	460.8 KHz	28,800
0x00	0x30	0x0030	48	307.2 KHz	19,200
0x00	0x40	0x0040	64	230.4 KHz	14,400
0x00	0x60	0x0060	96	115.2 KHz	9,600
0x00	0xC0	0x00C0	192	76.8 KHz	4,800
0x01	0x80	0x0180	384	38.4 KHz	2,400
0x03	0x00	0x0300	768	19.2 KHz	1,200

TO

Table 129

Divisor Latch High Register	Divisor Latch Low Register	Divisor		Baud Rate Generator Clock Output	Baud Rate
		Hexadecimal	Decimal		
0x00	0x01	0x0001	1	14.7456 MHz	921,600
0x00	0x02	0x0002	2	7.3728 MHz	460,800
0x00	0x04	0x0004	4	3.6864 MHz	230,400
0x00	0x08	0x0008	8	1.8432 MHz	115,200
0x00	0x10	0x0010	16	921.6 KHz	57,600
0x00	0x20	0x0020	32	460.8 KHz	28,800
0x00	0x30	0x0030	48	307.2 KHz	19,200
0x00	0x40	0x0040	64	230.4 KHz	14,400
0x00	0x60	0x0060	96	153.6 KHz	9,600
0x00	0xC0	0x00C0	192	76.8 KHz	4,800
0x01	0x80	0x0180	384	38.4 KHz	2,400
0x03	0x00	0x0300	768	19.2 KHz	1,200

FROM

Table 244

Divisor Latch High Register	Divisor Latch Low Register	Divisor		Baud Rate Generator Clock Output	Baud Rate
		Hexadecimal	Decimal		
0x00	0x01	0x0001	1	14.7456 MHz	921,600
0x00	0x02	0x0002	2	7.3728 MHz	460,800
0x00	0x04	0x0004	4	3.6864 MHz	230,400
0x00	0x08	0x0008	8	1.8432 MHz	115,200
0x00	0x10	0x0010	16	921.6 KHz	57,600
0x00	0x20	0x0020	32	460.8 KHz	28,800
0x00	0x30	0x0030	48	307.2 KHz	19,200
0x00	0x40	0x0040	64	230.4 KHz	14,400
0x00	0x80	0x0080	96	115.2 KHz	9,600
0x00	0xC0	0x00C0	192	76.8 KHz	4,800
0x01	0x80	0x0180	384	38.4 KHz	2,400
0x03	0x00	0x0300	768	19.2 KHz	1,200

TO

Table 244

Divisor Latch High Register	Divisor Latch Low Register	Divisor		Baud Rate Generator Clock Output	Baud Rate
		Hexadecimal	Decimal		
0x00	0x01	0x0001	1	14.7456 MHz	921,600
0x00	0x02	0x0002	2	7.3728 MHz	460,800
0x00	0x04	0x0004	4	3.6864 MHz	230,400
0x00	0x08	0x0008	8	1.8432 MHz	115,200
0x00	0x10	0x0010	16	921.6 KHz	57,600
0x00	0x20	0x0020	32	460.8 KHz	28,800
0x00	0x30	0x0030	48	307.2 KHz	19,200
0x00	0x40	0x0040	64	230.4 KHz	14,400
0x00	0x60	0x0060	96	153.6 KHz	9,600
0x00	0xC0	0x00C0	192	76.8 KHz	4,800
0x01	0x80	0x0180	384	38.4 KHz	2,400
0x03	0x00	0x0300	768	19.2 KHz	1,200

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual*

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual*

6. IXP46X Developers Manual - EXP_TIMING_CS0 Reset Hex Value

Issue: In section 12.5.1 of the August 2006 release of the *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual*, the Reset Hex Value of EXP_TIMING_CS0 is defined as 0xBFFF3C4x. This value is incorrect and should be updated to BFFF024X.

FROM

Register Name:	**EXP_TIMING_CS0**			
Hex Offset Address:	0XC4000000		**Reset Hex Value:**	**CS0: 0xBFFF3C4x**
Register Description:	Timing and Control Registers			
Access: Read/Write				

31	30	29	28	27	26	25			22	21	20	19			16	15	14	13				9	8	7	6	5	4	3	2	1	0
CSx_EN	PAR_EN	T1		T2		T3				T4		T5				CYCLE_ TYPE		CNFG[4:0]					Sync_Intel	EXP_CHIP	BYTE_RD16	HRDY_POL	MUX_EN	SPLT_EN	WORD_EN	WR_EN	BYTE_EN

TO

Register Name:	**EXP_TIMING_CS0**			
Hex Offset Address:	0XC4000000		**Reset Hex Value:**	**CS0: 0xBFFF024x**
Register Description:	Timing and Control Registers			
Access: Read/Write				

31	30	29	28	27	26	25			22	21	20	19			16	15	14	13				9	8	7	6	5	4	3	2	1	0
CSx_EN	PAR_EN	T1		T2		T3				T4		T5				CYCLE_ TYPE		CNFG[4:0]					Sync_Intel	EXP_CHIP	BYTE_RD16	HRDY_POL	MUX_EN	SPLT_EN	WORD_EN	WR_EN	BYTE_EN

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual*

7. *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* and *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual*: PCI Configuration Read / Write Discrepancy

Issue: In Section 6.4 of the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* and in Section 10.2.5 of the *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual*, there are some discrepancies regarding how to write to the PCI Configuration Register. According to the section's paragraph, PCI_CRP_WDATA must be updated first and then followed by PCI_CRP_AD_CBE. However, in the example given, PCI_CRP_AD_CBE is written first followed by PCI_CRP_WDATA. The correct sequence is write to PCI_CRP_AD_CBE and then follow with PCI_CRP_WDATA.

- In Section 6.4 of the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* and in Section 10.2.5 of the *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual*, update the fifth paragraph as below:

FROM:

When a write to the PCI Configuration Space is desired, the AHB master requesting the write must update the PCI Configuration Port Write Data (PCI_CRP_WDATA) Register with the data that is to be written to the PCI Configuration Register. Once the PCI Configuration Port Write Data (PCI_CRP_WDATA) Register has been updated, a write command is written into the command field of the PCI Configuration Port Address/ Command/Byte Enables (PCI_CRP_AD_CBE) Register along with the appropriate byte enables and address of the PCI Configuration register to be accesses.

TO:

When a write to the PCI Configuration Space is desired, the AHB master requesting the write must send a write command to the command field of the PCI Configuration Port Address/Command/Byte Enables (PCI_CRP_AD_CBE) Register along with the appropriate byte enables and address of the PCI Configuration register to be accessed. Once the PCI_CRP_AD_CBE Register has been updated, the data that is to be written to the PCI Configuration Register is written into the PCI Configuration Port Write Data (PCI_CRP_WDATA) Register.

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual (August 2006).*

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual (September 2006)*

8.0 Documentation Changes

1. Unused Acronyms that must be Removed from the List

Issue: The Acronym List found in Section 1.5 of the *Intel® IXDP425 / IXCDP1100 Development Platform User's Guide* must be updated to remove all unused acronyms.

Affected Docs: *Intel® IXDP425 / IXCDP1100 Development Platform User's Guide* (Sept. 2004)

2. Configuration Strapping Clock Settings, and Clarification

Issue: It is not very clear how to set the configuration strapping switch ON or OFF to set bits to 0 or 1.

- Replace Table 16 Configuration Strapping Clock Settings with the following table. The below table clearly shows the position of the switch to configure each supported clock speed. ON will set the bit to 0, while OFF will set the bit to 1.

Configuration Strapping Clock Settings (Sheet 1 of 2)

Speed (Factory Part Speed)	EX_ADDR(23)	EX_ADDR(22)	EX_ADDR(21)	Actual processor Speed
667 MHz	OFF	X	X	667 MHz
667 MHz	ON	ON	ON	667 MHz
667 MHz	ON	ON	OFF	533 MHz
667 MHz	ON	OFF	ON	266 MHz
667 MHz	ON	OFF	OFF	400 MHz
533 MHz	OFF	X	X	533 MHz
533 MHz	ON	ON	ON	533 MHz
533 MHz	ON	ON	OFF	533 MHz
533 MHz	ON	OFF	ON	266 MHz
533 MHz	ON	OFF	OFF	400 MHz

Configuration Strapping Clock Settings (Sheet 2 of 2)

Speed (Factory Part Speed)	EX_ADDR(23)	EX_ADDR(22)	EX_ADDR(21)	Actual processor Speed
400 MHz	OFF	X	X	400 MHz
400 MHz	ON	ON	ON	400 MHz
400 MHz	ON	ON	OFF	400 MHz
400 MHz	ON	OFF	ON	266 MHz
400 MHz	ON	OFF	OFF	400 MHz
266 MHz	X	X	X	266 MHz

Affected Docs: *Intel® IXDP425 / IXCDP1100 Development Platform User's Guide* (Sept. 2004)

3. *Intel® IXDP425 / IXCDP1100 Development Platform User's Guide*: Update Configuration Register Table

Issue: In Table 9 of the September release of the *Intel® IXDP425 / IXCDP1100 Development Platform User's Guide*, the definition of bits [23:21] must be updated and the new SKU IXP423BD must be added. On page 29, replace Table 9 with the following table:

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>EX_ADDR[23:21]</td><td>Intel XScale processor Clock Set[2:0]</td><td>Allow a slower Intel XScale processor clock speed to override device factory settings of the on-board, 533-MHz IXP425 network processor.
<table>
<tr><th>Part in Processor socket</th><th>Desired Frequency</th><th>SW2[8: 7: 6] Setting</th></tr>
<tr><td rowspan="3">IXP420BD, IXP423BD, IXP425BD or IXC1100BD</td><td>533 MHz</td><td>[High: High: High] (default)</td></tr>
<tr><td>400 MHz</td><td>[Low: Low: High]</td></tr>
<tr><td>266 MHz</td><td>[Low: High: High]</td></tr>
<tr><td rowspan="2">IXP420BC, IXP425BC, or IXC1100BC</td><td>400 MHz</td><td>[High: High: High]</td></tr>
<tr><td>266 MHz</td><td>[Low: High: High]</td></tr>
<tr><td>IXP420BB, IXP421BB, IXP422BB, IXP423BB, IXP425BB, or IXC1100BB</td><td>266 MHz</td><td>[High: High: High]</td></tr>
</table>
Clock speed may not be set to a higher value than the value set by the device's factory setting.</td></tr>
</table>

Affected Docs: *Intel® IXDP425 / IXCDP1100 Development Platform User's Guide (September 2004)*.

4. Update on HDLC Coprocessor

Issue: Add the following paragraphs as Section 4.1 in both *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual* and *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual*:

4.1 HDLC Coprocessor

The primary function of the High-level Data Link Control (HDLC) Coprocessor is to encapsulate data into HDLC frames. Extracting data from an HDLC frame is also supported. HDLC processing is bit-oriented in nature. It is inefficient to do this in software and therefore, hardware support is required. HDLC processing is performed on data in the NPE cores data memory. The HDLC coprocessor is instantiated once and runs off the 133 MHz clock.

4.1.1 HDLC Coprocessor Features List

The hardware support for HDLC processing will only be for the bit-level processing. Specifically, only the following features are supported and performed in the order listed:

On transmit (encapsulating data in HDLC frames):

- FCS (CRC) generation – both 16- and 32-bit CRC polynomials are supported.
- Bit stuffing – any sequence of five consecutive ones is followed by a 0. ITU-T Q.921 refers to this as “transparency”.
- Flag generation – the flag sequence is defined as 0x7E (01111110). Two flags encapsulate an HDLC frame. The start/end flags of successive frames may be shared. Transmit can be configured to start a frame with 0-2 consecutive flags (0 implying start/end flag sharing). However, in FCS 32-bit mode, the number of start-of-frame flags is limited to 0 or 1 flags for frame lengths less than or equal to 4 bytes. Bit sharing between consecutive flags is not possible.
- Idle generation – two idle modes are supported - continuous flags and continuous ones.
- Abort sequence generation – the abort sequence is defined as 0x7F (1111111). This must be generated by the NPE core; the coprocessor does not assist this.
- Pre- and post-processing bit flipping on byte boundary capability.

On receive (removing framing to recover data):

- Error sequence recognition – The coprocessor will recognize an abort sequence and set the RxStat.ABT status bit. This will also set the RxStat.EOF status bit.
- Discard of inter-frame data – flags, idle indicators, and any data not within a frame, is discarded without alerting the NPE core. This includes any data at startup before the first flag is received. Two idle modes are supported: continuous flags and continuous 1s.
- Flag recognition – the flag sequence is used to indicate that a valid frame is in progress. Flag recognition is performed in conformance with the transmit rules above. Bit sharing between successive flags is supported on receive; two consecutive flags may share their start/end bits.
- Bit destuffing – any 0 that follows five consecutive ones is removed.
- FCS checking – the FCS value is checked using the same polynomial as for transmit and the RxStat.FCS status bit is set if the check fails. This bit is only valid at end-of-frame if no other error condition (abort or residual character error) is present.
- Pre and post-processing bit flipping on byte boundary capability.

No provision has been made for a 56K mode for HDLC data.

A frame contains data in an integer number of bytes (before bit stuffing). Depending on their size, frames are handled as described in Table 96.

Table 96 Received Frame Length Handling

FCS Configuration	Frame Length	Hardware Action	Firmware Notification
CRC-16	1-3 bytes	Invalid frame. Discarded.	No
	>= 4 bytes	Valid frame. Passed to NPE core.	Yes
CRC-32	1-3 bytes	Invalid frame. Discarded.	No
	4-5 bytes	Invalid frame. Passed to NPE core. Alignment error asserted.	Yes
	>= 6 bytes	Valid frame. Passed to NPE core.	Yes

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* (March 2005)

5. Improvement of the SDRAM Initialization Sequence

Issue: In *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual*/*Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Hardware Design Guidelines*, under Initializing the SDRAM section, there is a reference routine for initializing the SDRAM. This routine can be improved by re-enabling the refresh counter at the end of the routine. By doing so, this ensures that the auto-refresh commands initiated by the counter do not potentially interfere with the initialization sequence. Additionally, some clarifications are made in the initialization sequence.

- In the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual*(page 283) and *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Hardware Design Guidelines* (pages 24-25), update the initialization sequence as shown below:

FROM:

- The memory controller applies the clock pin (SDM_CKE) during power up and must stabilize the clock signal within 100 µs after power stabilizes.
- The memory controller holds all the control pins to the memory inactive (SDM_RAS_N, SDM_CAS_N, SDM_WE_N, SDM_CS_N[1:0]=1) for a minimum of 1 millisecond after supply voltage reaches the desired level.
- SDM_CKE is driven to VCC all the time. The IXP42X product line and IXC1100 control plane processors never de-assert SDM_CKE.
- Software disables the refresh counter by setting SDR_REFRESH to zero.
- Software issues one NOP cycle after the 1milliseconds SDRAM device deselect. A NOP is accomplished by setting SDR_IR to 011. The memory controller asserts SDM_CKE with the NOP.
- Software pauses 200 µs after the NOP.
- Software re-enables the refresh counter by setting the SDR_REFRESH to the required value.
- Software issues a precharge-all command to the SDRAM interface by setting SDR_IR to 010.
- Software provides eight auto-refresh cycles. An auto-refresh cycle is accomplished by setting SDR_IR to 100. Software must ensure at least Trc cycles between each auto-refresh command. Trc (active-to-active command period) is determined by the SDRAM being used.
- Software issues a mode-register-select command by writing to SDR_IR to program the SDRAM parameters. Setting SDR_IR to 000 programs the SDRAM Controller for CAS Latency of two while setting the SDR_IR to 001 programs the memory controller and SDRAM for CAS Latency of three.

- The SDRAM Controller may issue a row activate command three clocks after the mode register set command.

TO:

- The memory controller applies the clock pin (SDM_CKE) during power up and must stabilize the clock signal within 100 µs after power stabilizes.
- The memory controller holds all the control pins to the memory inactive (SDM_RAS_N, SDM_CAS_N, SDM_WE_N, SDM_CS_N[1:0]=1) for a minimum of 1 millisecond after supply voltage reaches the desired level.
- SDM_CKE is driven to VCC all the time. The IXP42X product line and IXC1100 control plane processors never de-assert SDM_CKE.
- Software disables the refresh counter by setting SDR_REFRESH to zero.
- Software issues one NOP cycle after the 1milliseconds SDRAM device deselect. A NOP is accomplished by setting SDR_IR to 011.
- Software pauses 200ms after the NOP.
- Software issues a precharge-all command to the SDRAM interface by setting SDR_IR to 010.
- Software provides eight auto-refresh cycles. An auto-refresh cycle is accomplished by setting SDR_IR to 100. Software must ensure at least Trc cycles between each auto-refresh command. Trc (active-to-active command period) is determined by the SDRAM being used.
- Software issues a mode-register-select command by writing to SDR_IR to program the SDRAM parameters. Setting SDR_IR to 000 programs the SDRAM Controller for CAS Latency of two while setting the SDR_IR to 001 programs the memory controller and SDRAM for CAS Latency of three.
- The SDRAM Controller may issue a row activate command three clocks after the mode register set command.
- Software re-enables the refresh counter by setting the SDR_REFRESH to the required value.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual (Sept. 2006)*

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Hardware Design Guidelines*

6. Additional Design Note for SDRAM Initialization/Reset Timing

Issue: The SDRAM controller of the IXP42X product line has an internal auto-refresh counter that is set to the default value of 0x384 (900 cycles). The first SDRAM access initiated by the auto-refresh counter happens ~7µs after RESET_IN_N is released. Meanwhile, the IXP42X product line requires that RESET_IN_N is released at least 10ns after PWRON_RST_N. Additionally, the SDRAM clock starts when PWRON_RST_N is released. Therefore, if a system is designed according to the minimum requirement, the first access to the SDRAM will happen 7.01µs after PWRON_RST_N is released.

Most SDRAMs require 100-200 ms after clock has stabilized before the first access is made. Therefore, system designers should take this into consideration while deciding on the time interval between the release of PWRON_RST_N and RESET_IN_N.

The following design note should supplement the SDRAM Initialization section and the Power Up section in the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Hardware Design Guidelines*:

Note: The SDRAM clock starts with the release of PWRON_RESET_N. The first access made by the internal auto-refresh counter (set to default value of 0x384) happens ~7µs after RESET_IN_N is released. Software will not be able to disable this counter before the first access is made. To meet a

specific SDRAMs 100-200 ms requirement before the first access, the designer may have to add additional delay between PWRON_RESET_N and RESET_IN_N beyond the required minimum of 10 ns.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Hardware Design Guidelines*

7. *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* and *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual*: Ethernet Register Discrepancies

Issue: There are discrepancies in Sections 15.2.24, 15.2.31, and 15.2.39 of the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* and in Sections of 6.2.30, 6.2.37, and 6.2.45 of the *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual* between the introduction paragraph and the register tables for the Ethernet Address Mask, Address, and Unicast Address.

The register tables in all of the sections mentioned above and their respective subsections should be updated to match the information in the opening paragraph.

For example, in Section 6.2.30, the opening paragraph states:

- Address Mask[47:40] — Address Mask 1
- Address Mask[39:32] — Address Mask 2
- Address Mask[31:24] — Address Mask 3
- Address Mask[23:16] — Address Mask 4
- Address Mask[15:8] — Address Mask 5
- Address Mask[7:0] — Address Mask 6

Therefore, in the following subsections, Section 6.2.31 to 6.2.36, the register tables should be updated as follows:

FROM:

31 … 8	7 … 0
(Reserved)	ADDRESS MASK[7:0]
(Reserved)	ADDRESS MASK[15:8]
(Reserved)	ADDRESS MASK[23:16]
(Reserved)	ADDRESS MASK[24:31]
(Reserved)	ADDRESS MASK[39:32]
(Reserved)	ADDRESS MASK[47:40]

TO:

31 … 8	7 … 0
(Reserved)	ADDRESS MASK[47:40]
(Reserved)	ADDRESS MASK[39:32]
(Reserved)	ADDRESS MASK[31:24]
(Reserved)	ADDRESS MASK[23:16]
(Reserved)	ADDRESS MASK[15:8]
(Reserved)	ADDRESS MASK[7:0]

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual (August 2006)*.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual (September 2006)*

8. *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual*: MEM_TYPE strap reference

Issue: In section 11.6.2 of the *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual*, bit 2 of the DDR SDRAM Control Register 0 (SDCR0) refers a MEM_TYPE reset strap. The MEM_TYPE reset strap is a reserved strapping option and will default to 1 for the IXP45X/IXP46X network processors. Update Table 229 on page 708 as below:

FROM:

Bit	Name	Description	Default	Access
02	DDR	**DDR:** Identifies the generation of DDR SDRAM selected by the **MEM_TYPE** reset strap. A **MEM_TYPE** value of 1_2 **must** be used for the IXP45X/IXP46X network processors. 0 = RESERVED for DDRII/400 MHz 1 = DDRI/266 MHz	Varies with external state of **MEM_TYPE** at reset Should default to 1	RO

TO:

Bit	Name	Description	Default	Access
02	DDR	**DDR:** Identifies the generation of DDR SDRAM selected by the **MEM_TYPE** reset strap. A **MEM_TYPE** reset strap value of 1_2 **must** be used and is set by default for the IXP45X/IXP46X network processors. 0 = RESERVED for DDRII/400 MHz 1 = DDRI/266 MHz	Should default to 1	RO

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual (August 2006)*.

9. *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual*: Update to GPCLKR Register Table

Issue: In Section 15.5.7 of the *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual*, the GPCLK register table shows that bits [19:9] are assigned as CLK1DC. The correct bit assignment for CLK1DC should be bit [19:16], where bits [15:9] are reserved. Update the table as below:

FROM:

Register Name:	GPCLKR		
Physical Address:	0xC8004018	Reset Hex Value:	0x01100000
Register Description:	This register controls the use of GPIO 15 and GPIO14 as clock sources		
Access: Read/Write			

31															16	15							8	7							0
(Reserved)								Mx15	CLK1TC						CLK1DC								Mx14	CLK0TC				CLK0DC			

TO:

Register Name:	GPCLKR		
Physical Address:	0xC8004018	Reset Hex Value:	0x01100000
Register Description:	This register controls the use of GPIO 15 and GPIO14 as clock sources		

Access: Read/Write

31												19			16	15						9	8	7							0
(Reserved)							Mx15	CLK1TC				CLK1DC				(Reserved)							Mx14	CLK0TC				CLK0DC			

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual (August 2006)*.

10. *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* and *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual*: Incorrect Memory Size Indicated in Memory Map

Issue: In Table 96 of the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* and in Table 99 of the *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual*, the Memory Map states an incorrect memory size of 1 KB for addresses X000 - XFFF. The tables should be updated to the correct memory size, 4 KB.

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual (August 2006).*

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual (September 2006)*

§ §