# Intel® IXP4XX Product Line of Network Processors

## Specification Update

***September 2005***

Document Number: 306428-003
September 2005

# Contents

# Revision History

| Date | Version | Description |
|---|---|---|
| September 2005 | 003 | Modifications since the last release of this document:<br>• Intel® IXP465 operating at 667 MHz requires Vcc Core Voltage increase from 1.4 V to 1.5 V.<br>• Intel® IXP465 operating at 667 MHz is no longer available with extended temperature support.<br>• Updated Table 1 and Table 2 with part numbers for A1 stepping.<br>• Indicated that Errata 40 (Timer Issues) and Errata 41 (IEEE-1588) have both been fixed for A1 Stepping of IXP46X Silicon.<br>• Added 4 specification changes (see Specification Changes summary table).<br>• Added 22 specification clarifications (see Specification Clarifications summary table).<br>• Added 4 documentation changes (see Documentation Changes summary table). |
| May 2005 | 002 | Modifications since the last release of this document:<br>• 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.<br>• 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.*<br>• Added Table 2, "Part Numbers for the Intel® IXP45X Product Line of Network Processors" on page 17.<br>• Added Spec Clarification #1 for "PCI Clock Slew Rate (SCR 4285)" on page 48.<br>• 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.<br>The first specification update for the Intel® IXP46X Product Line of Network Processors.<br>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:<br>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. |

# 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.

***Note:*** Unless otherwise specified, the errata assigned to Intel® IXP42X network processors apply to all of 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.

## 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® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Design Checklist* | 254238 |
| *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 |

# Nomenclature

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

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

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

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

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

# Summary Table of Changes

The following tables indicate the errata, specification changes, specification clarifications, or documentation changes which 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:

## Codes Used in Summary Tables

### 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 |

### Page

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

### 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*. |

### Row

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

# Non-Core Errata (Sheet 1 of 2)

| Errata No. | Steppings | | | | | | Page | Status | Errata |
|---|---|---|---|---|---|---|---|---|---|
| | IXP 425 A0 | IXP 42X B0 | IXP 45X A0 | IXP 46X A0 | IXP 45X A1 | IXP 46X A1 | | | |
| 1 | X | | | | | | 19 | Fixed | PCI Doorbell Register Does Not Work Properly (SCR 460) |
| 2 | X | | | | | | 19 | Fixed | UTOPIA Interface Status Collection Synchronization Issues (SCR 485) |
| 3 | X | | | | | | 19 | Fixed | No Byte Enables Asserted During a PCI I/O Read of the Intel® IXP425 Network Processor Causes Unpredictable Behavior (SCR 469) |
| 4 | X | | | | | | 20 | Fixed | Simultaneous AHB Access of the PCI Bus Controller (SCR 499) |
| 5 | X | | | | | | 20 | Fixed | 66-MHz PCI Operation (SCR 543) |
| 6 | X | X | X | X | X | X | 20 | No Fix | Ethernet Control Protocol Frames Transmit-Defer Status Bit Error (SCR 472) |
| 7 | X | | | | | | 21 | Fixed | Logic 0 is Driven on Both the USB D+ and D- at Reset (SCR 545) |
| 8 | X | | | | | | 21 | Fixed | Cannot Generate Watchdog Timer Reset (SCR 641) |
| 9 | X | | | | | | 21 | Fixed | PCI Non-Prefetch Reads (SCR 1254) |
| 10 | X | X | | | | | 22 | Varies | Timer Status Interrupts Get Lost During MMR Writes (SCR 1653) |
| 11 | X | X | | | | | 23 | Varies | Character Time-Out Interrupt Sticks Under Certain Software Timing Conditions (SCR 2235) |
| 12 | X | | | | | | 24 | Fixed | Intel® IXP425 A-0 Step Processor May Have Problems Working With Some SDRAM Devices (SCR 2411) |
| 13 | X | X | | | | | 25 | Varies | PCI DMA Lock-Up Condition (SCR 2372) |
| 14 | X | X | X | X | X | X | 26 | No Fix | PCI Doorbell Register Lock-up Condition When Using Two Products Together that have Intel® IXP4XX Product Line of Network Processors (SCR 2379) |
| 15 | X | X | | | | | 26 | Varies | PCI Controller Returns Infinite Retries on PCI After AHB Pre-Fetch Error (SCR 1289) |
| 16 | X | X | | | | | 27 | Varies | UART Break Indicator (SCR 2929) |
| 17 | X | X | X | X | X | X | 27 | No Fix | Intel XScale® Core Non-Branch Instruction in Vector Table (SCR 2871) |
| 18 | X | X | X | X | X | X | 27 | No Fix | IRQ 3 Is Locking the System ‘Disable’ (SCR 2143) |
| 19 | X | X | X | X | X | X | 28 | No Fix | EX_IOWAIT_N Timing (SCR 3051) |
| 20 | X | | | | | | 28 | Fixed | SOF During Control Read Can Corrupt USB Transfer (SCR 1553) |
| 21 | X | X | X | X | X | X | 29 | No Fix | USB - Invalid 0x81 Value in Endpoint 0 Control Register on SETUP Transfer (SCR 3077) |
| 22 | X | X | X | X | X | X | 29 | No Fix | Ethernet Coprocessors — Ethernet Pad Enable Overrides Append FCS (SCR 299) |
| ***Note:*** Unless otherwise specified, the errata assigned to IXP42X B0 processors also apply to IXC1100 processors. | | | | | | | | | |

# Non-Core Errata (Sheet 2 of 2)

| Errata No. | Steppings | | | | | | Page | Status | Errata |
|---|---|---|---|---|---|---|---|---|---|
| | IXP 425 A0 | IXP 42X B0 | IXP 45X A0 | IXP 46X A0 | IXP 45X A1 | IXP 46X A1 | | | |
| 23 | X | X | X | X | X | X | 30 | No Fix | Ethernet Coprocessors — Length Errors on Received Frames (SCR 711) |
| 24 | X | X | | | | | 30 | Varies | PCI DC Parameter VIH Marginality Issue (SCR 3121) |
| 25 | X | X | X | X | X | X | 30 | No Fix | False PCI DMA Completion Notification Causing Data Corruption (SCR 3910) |
| 26 | X | X | X | X | X | X | 31 | No Fix | Expansion Bus HPI Interface Potential for Contention on Reads with T4=0 (SCR 4117) |
| 27 | X | X | X | X | X | X | 31 | No Fix | PCI Hangs With a Multiple Inbound Error Condition (SCR 4160) |
| 28 | X | X | X | X | X | X | 32 | No Fix | PCI RCOMP Operation if PCI Clock Stops (SCR 4022) |
| 29 | X | X | X | X | X | X | 32 | No Fix | UART — Break Condition Asserted Too Early if Two Stop bits are Used (SCR 4092) |
| 30 | X | X | X | X | X | X | 32 | No Fix | Ethernet Coprocessors — Address Filtering Logic Ignores the Second to Last Nibble of the Destination Address (SCR 4185) |
| 31 | X | X | X | X | X | X | 33 | No Fix | USB DC Parameter Vih Specification Change (SCR 3111) |
| 32 | X | X | | | | | 33 | Varies | Start of DMA Operation Close to Start of Non-Prefetch (NP) Operation Can Hang NP Operation on AHB (SCR 3939) |
| 33 | X | X | X | X | X | X | 34 | No Fix | PCI Accesses to the Queue Manager During Queue and SRAM Mode (SCR 4076) |
| 34 | X | X | X | X | X | X | 34 | No Fix | Ethernet MACs Detect Late Collision Earlier Than Ethernet 802.3 Specifications (SCR 4062) |
| 35 | X | X | X | X | X | X | 34 | No Fix | Read of PCI Controllers BAR 32'h XXFF_FFFC Rd[N] Corrupts Subsequent Rd[N+1] (SCR 3850) |
| 36 | X | X | X | X | X | X | 35 | No Fix | UART Operating in Non-FIFO Mode Can Falsely Receive Overrun Error (SCR 1554) |
| 37 | X | X | X | X | X | X | 35 | No Fix | USB-Specification Noncompliance for Rise/Fall Transition Times (SCR 3633) |
| 38 | X | X | X | X | X | X | 36 | No Fix | Ethernet MAC Does Not Detect Transmit FIFO Underruns Reliably (SCR 4230) |
| 39 | N/A | N/A | X | X | X | X | 36 | No Fix | SMII late_col Occurs Earlier Than Expected (SCR 4045) |
| 40 | N/A | N/A | X | X | | | 37 | Fixed | Timer Issues with Prescale Programming Sequence and Pause/Resume Operation (SCR 4216) |
| 41 | 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 (SCR 4239) |

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

# Core Errata

| Errata No. | Steppings | | | | | | Page | Status | Errata |
|---|---|---|---|---|---|---|---|---|---|
| | IXP 425 A0 | IXP 42X B0 | IXP 45X A0 | IXP 46X A0 | IXP 45X A1 | IXP 46X A1 | | | |
| 1 | X | X | X | X | X | X | 40 | No Fix | Abort is Missed When Lock Command is Outstanding |
| 2 | X | X | X | X | X | X | 40 | No Fix | Aborted Store that Hits the Data Cache May Mark Write-Back Data as 'Dirty' |
| 3 | X | X | X | X | X | X | 41 | No Fix | Performance Monitor Unit Event 0x1 Can Be Incremented Erroneously by Unrelated Events |
| 4 | X | X | X | X | X | X | 41 | 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 | 42 | No Fix | Accesses to the CP15 ID Register with Opcode2 > 0b001 Returns Unpredictable Values |
| 6 | X | X | X | X | X | X | 42 | No Fix | Disabling and Re-Enabling the MMU Can Hang the Core or Cause it to Execute the Wrong Code |
| 7 | X | X | X | X | X | X | 43 | No Fix | Updating the JTAG Parallel Registers Requires an Extra TCK Rising Edge |
| 8 | X | X | X | X | X | X | 44 | No Fix | Non-Branch Instruction in Vector Table May Execute Twice After a Thumb Mode Exception |
| ***Note:*** Unless otherwise specified, the errata assigned to IXP42X B0 processors also apply to IXC1100 processors. | | | | | | | | | |

# Specification Changes

| Ref. # | Steppings | | | | | | Page | Status | Specification Changes |
|---|---|---|---|---|---|---|---|---|---|
| | IXP 425 A0 | IXP 42X B0 | IXP 45X A0 | IXP 46X A0 | IXP 45X A1 | IXP 46X A1 | | | |
| 1 | | | | | | X | 45 | Doc | Increase Core Voltage from 1.4 V to 1.5 V for 667 MHz Operation only (SCR 4291) |
| 2 | | | X | X | X | X | 46 | Doc | PCI ID Changes to Support A1 Stepping of IXP46X (No SCR) |
| 3 | X | X | | | | | 46 | Doc | External Crystal Support is No Longer Supported with IXP42X (SCR 4271) |
| 4 | X | X | | | | | 46 | Doc | IXC1100 Control Plane Processors Will Be Discontinued (SCR 4317) |
| ***Note:*** Unless otherwise specified, the errata assigned to IXP42X B0 processors also apply to IXC1100 processors. | | | | | | | | | |

# Specification Clarifications

| Ref. # | Steppings | | | | | | Page | Status | Specification Clarifications |
|---|---|---|---|---|---|---|---|---|---|
| | IXP 425 A0 | IXP 42X B0 | IXP 45X A0 | IXP 46X A0 | IXP 45X A1 | IXP 46X A1 | | | |
| 1 | X | X | | | | | 48 | Doc | PCI Clock Slew Rate (SCR 4285) |
| 2 | X | X | | | | | 48 | Doc | Ethernet Tx and Rx Clocks Must Be Specified as +/- ppm Frequency Tolerance (SCR 4222) |
| 3 | X | X | | | | | 49 | Doc | PCI Theoretical Peak Data Rate Should be Removed (SCR 4237) |
| 4 | X | X | | | | | 49 | Doc | PCI_CLKIN Pull-Down Recommendation (SCR 4238) |
| 5 | X | X | | | | | 49 | Doc | PCI_IDSEL Pull-up Recommendation (SCR 4244 and SCR 4260) |
| 6 | X | X | | | | | 49 | Doc | Ethernet Clock to Output Delay of 0 ns to be Removed (SCR 4257) |
| 7 | X | X | | | | | 49 | Doc | XScale Core Speed Strapping Table to be Updated (SCR 4261) |
| 8 | X | X | | | | | 50 | Doc | RCOMP Pin Type Should be an Output, Not Input (SCR 4298) |
| 9 | X | X | | | | | 50 | Doc | Impedance Recommendation Should be Consistent at 50 Ω (SCR 4307) |
| 10 | X | X | | | | | 50 | Doc | Expansion Bus Configurations Straps that are Reserved (SCR 4308) |
| 11 | X | X | | | | | 50 | Doc | PCI Initiation Clock Cycle Range is Incorrect (SCR 4309) |
| 12 | X | X | | | | | 51 | Doc | IOWAIT Functional Description Update for IXP42X (SCR 4314) |
| 13 | | | X | X | X | X | 51 | Doc | OSC_VSSP Pin Assignment Contradiction (SCR 4326) |
| 14 | | | X | X | X | X | 51 | Doc | Memory Controller Maximum Throughput Clarification (SCR 4327) |
| 15 | | | X | X | X | X | 51 | Doc | Memory Controller Refresh Counter Bit Width Clarification (SCR 4328) |
| 16 | | | X | X | X | X | 51 | Doc | Clarification on Number of MII (Ethernet) Ports for IXP460 (SCR 4331) |
| 17 | | | X | X | X | X | 52 | Doc | PCI to South AHB Address Error (SCR 4322) |
| 18 | | | X | X | X | X | 52 | Doc | APB Clocks Incorrectly Defined (No SCR) |
| 19 | X | X | | | | | 52 | Doc | IXP42X Power Clarification for Total $I_{CC}$ in Power Tables (No SCR) |
| 20 | | | X | X | X | X | 52 | Doc | IXP46X Power Clarification for Total $I_{CC}$ in Power Table (No SCR) |
| 21 | | | X | X | X | X | 53 | Doc | Memory Controller Clarification on why 8 ECC bits are supported (SCR 4336) |
| 22 | | | X | X | X | X | 53 | Doc | $T_{POWER\_UP}$ Sequence Needs to be Fixed (SCR 4337) |

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

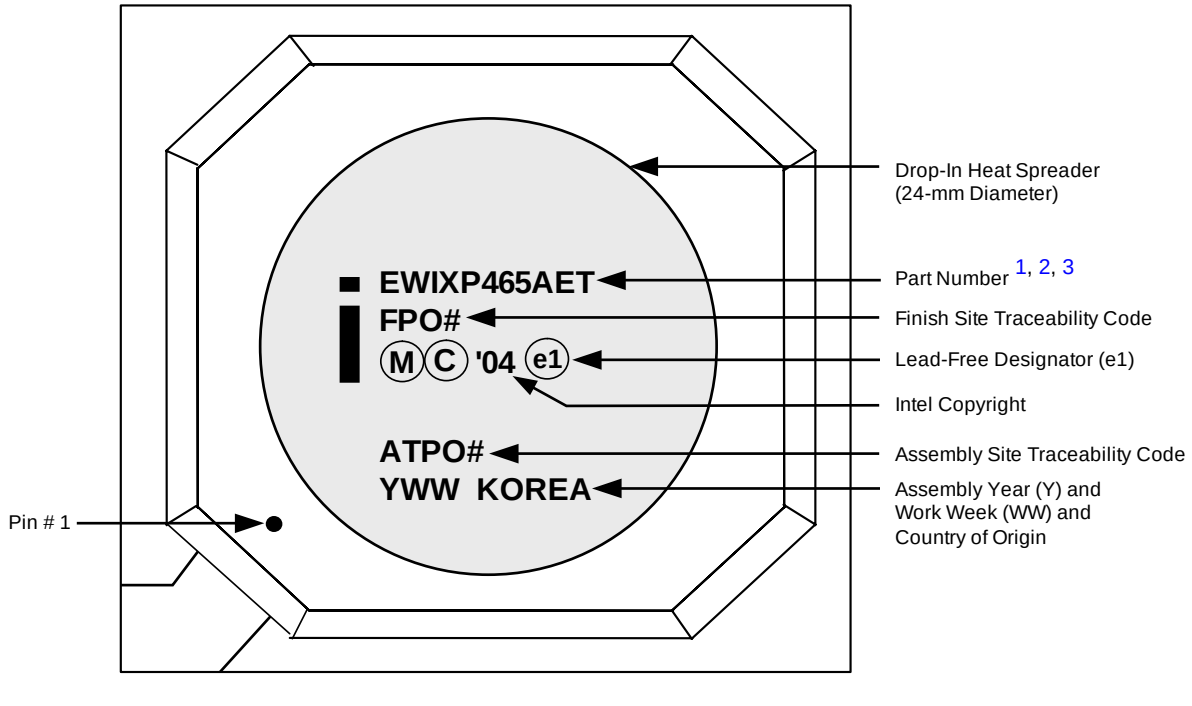
# Documentation Changes

| Ref. # | Steppings | | | | | | Page | Status | Documentation Changes |
|---|---|---|---|---|---|---|---|---|---|
| | IXP 425 A0 | IXP 42X B0 | IXP 45X A0 | IXP 46X A0 | IXP 45X A1 | IXP 46X A1 | | | |
| | X | X | | | | | 54 | Doc | Unused Acronyms Need to be Removed from List (SCR 4274) |
| | | | X | X | X | X | 54 | Doc | Memory Controller Cross Reference Error (SCR 4329) |
| | | | X | X | X | X | 54 | Doc | Memory Controller Text Clarifications in 6 Places (SCR 4330) |
| | X | X | X | X | X | X | 54 | Doc | HSS Figure and Table Text Clarifications (SCR 4333) |
| ***Note:*** Unless otherwise specified, the errata assigned to IXP42X B0 processors also apply to IXC1100 processors. | | | | | | | | | |

# Identification Information

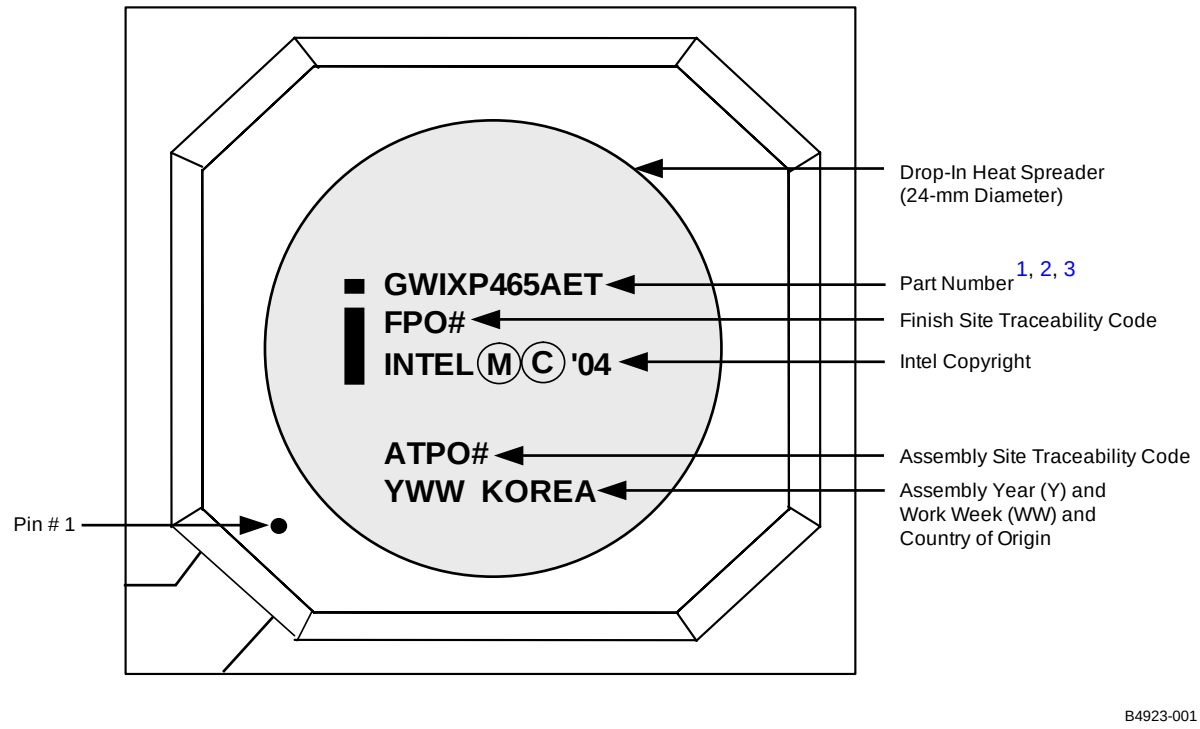
## Markings

**Figure 1. 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)**
**Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor — Extended Temperature, Lead-Free**



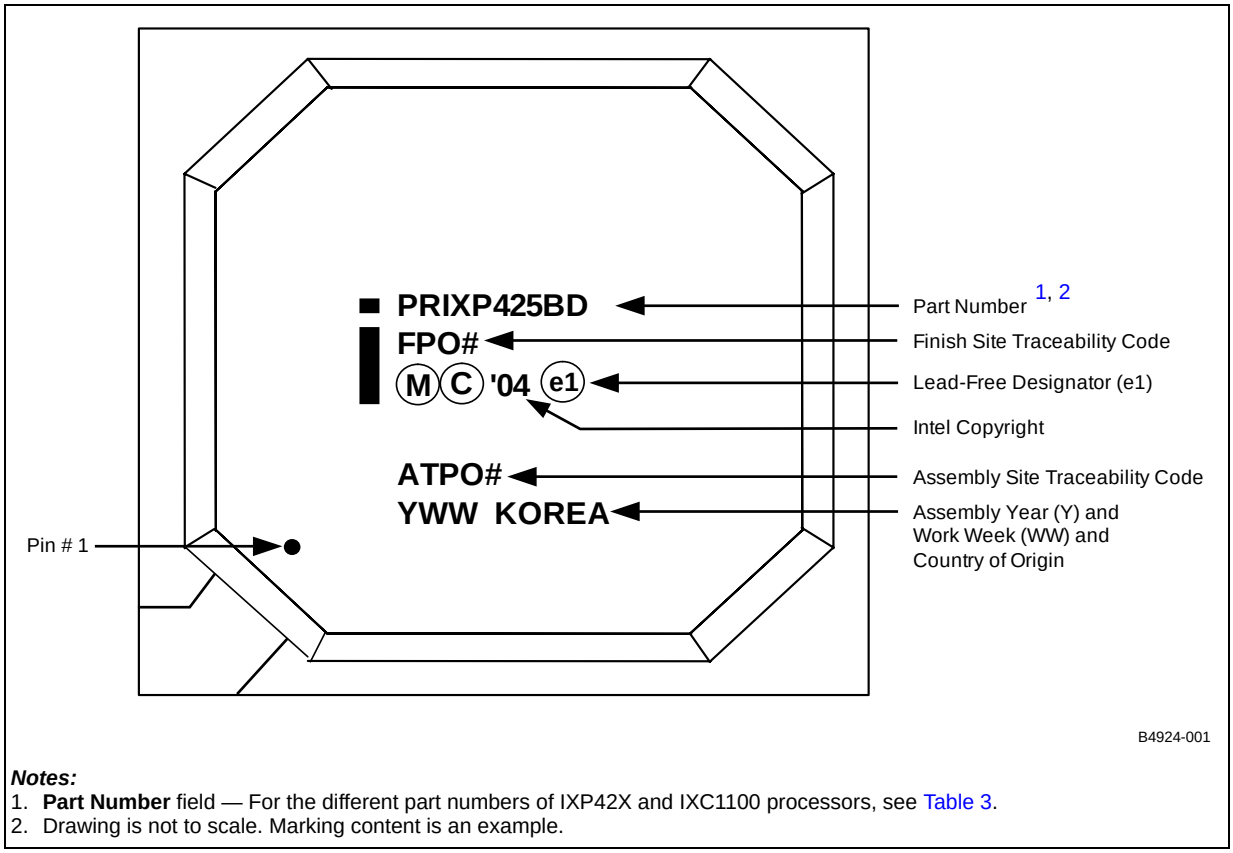

***Notes:***
1. **Part Number** field — For the different part numbers of Intel® IXP46X Product Line of Network Processors, see Table 1. For the Intel® IXP45X Product Line, see Table 2. For IXP42X and IXC1100 processors, see Table 3.
2. Package ball counts — Intel® IXP45X and Intel® IXP46X Product Line 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 2. 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**



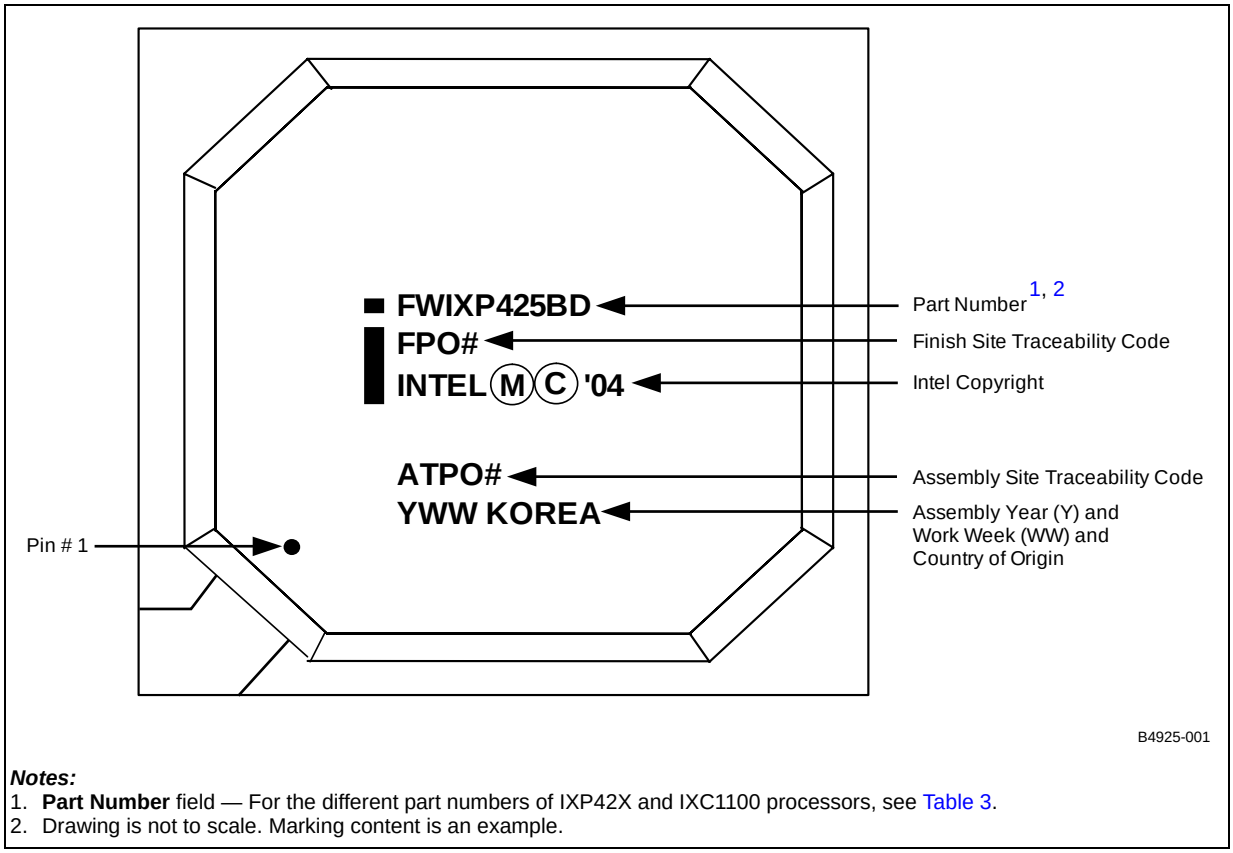

***Notes:***
1. **Part Number** field — For the different part numbers of Intel® IXP46X Product Line of Network Processors, see Table 1. For the Intel® IXP45X Product Line, see Table 2. For IXP42X and IXC1100 processors, see Table 3.
2. Package ball counts — Intel® IXP45X and Intel® IXP46X Product Line 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 3. Package Markings: Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor — Commercial Temperature, Lead-Free / RoHS-Compliant**




***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.

**Figure 4. 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.

# Part Numbers

## 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 | A1 | 667 | Commercial | Yes | EWIXP465AAE |
| Intel® IXP465 | A1 | 533 | Commercial | Yes | EWIXP465AAD |
| Intel® IXP465 | A1 | 400 | Commercial | Yes | EWIXP465AAC |
| Intel® IXP465 | A1 | 266 | Commercial | Yes | EWIXP465AAB |
| Intel® IXP465 | A1 | 533 | Extended | Yes | EWIXP465AADT |
| Intel® IXP465 | A1 | 400 | Extended | Yes | EWIXP465AACT |
| Intel® IXP465 | A1 | 266 | Extended | Yes | EWIXP465AABT |

**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® IXP465 | A1 | 667 | Commercial | | GWIXP465AAE |
| Intel® IXP465 | A1 | 533 | Commercial | | GWIXP465AAD |
| Intel® IXP465 | A1 | 400 | Commercial | | GWIXP465AAC |
| Intel® IXP465 | A1 | 266 | Commercial | | GWIXP465AAB |
| Intel® IXP465 | A1 | 533 | Extended | | GWIXP465AADT |
| Intel® IXP465 | A1 | 400 | Extended | | GWIXP465AACT |
| Intel® IXP465 | A1 | 266 | Extended | | GWIXP465AABT |
| Intel® IXP460 | A1 | 667 | Commercial | Yes | EWIXP460AAE |
| Intel® IXP460 | A1 | 533 | Commercial | Yes | EWIXP460AAD |
| Intel® IXP460 | A1 | 400 | Commercial | Yes | EWIXP460AAC |
| Intel® IXP460 | A1 | 266 | Commercial | Yes | EWIXP460AAB |
| Intel® IXP460 | A1 | 533 | Extended | Yes | EWIXP460AADT |
| Intel® IXP460 | A1 | 400 | Extended | Yes | EWIXP460AACT |
| Intel® IXP460 | A1 | 266 | Extended | Yes | EWIXP460AABT |
| Intel® IXP460 | A1 | 667 | Commercial | | GWIXP460AAE |
| Intel® IXP460 | A1 | 533 | Commercial | | GWIXP460AAD |
| Intel® IXP460 | A1 | 400 | Commercial | | GWIXP460AAC |
| Intel® IXP460 | A1 | 266 | Commercial | | GWIXP460AAB |
| Intel® IXP460 | A1 | 533 | Extended | | GWIXP460AADT |
| Intel® IXP460 | A1 | 400 | Extended | | GWIXP460AACT |
| Intel® IXP460 | A1 | 266 | Extended | | GWIXP460AABT |

## Intel® IXP45X Product Line of Network Processors

**Table 2. Part Numbers for the Intel® IXP45X Product Line of Network Processors (Sheet 1 of 2)**

| Device | Stepping | Speed (MHz) | Lead Free | Temperature Offering | Part # |
|---|---|---|---|---|---|
| Intel® IXP455 | A1 | 533 | Yes | Commercial | EWIXP455AAD |
| Intel® IXP455 | A1 | 400 | Yes | Commercial | EWIXP455AAC |
| Intel® IXP455 | A1 | 266 | Yes | Commercial | EWIXP455AAB |
| Intel® IXP455 | A1 | 533 | Yes | Extended | EWIXP455AADT |
| Intel® IXP455 | A1 | 400 | Yes | Extended | EWIXP455AACT |
| Intel® IXP455 | A1 | 266 | Yes | Extended | EWIXP455AABT |
| Intel® IXP455 | A1 | 533 | | Commercial | GWIXP455AAD |
| Intel® IXP455 | A1 | 400 | | Commercial | GWIXP455AAC |

**Table 2. Part Numbers for the Intel® IXP45X Product Line of Network Processors (Sheet 2 of 2)**

| Device | Stepping | Speed (MHz) | Lead Free | Temperature Offering | Part # |
|---|---|---|---|---|---|
| Intel® IXP455 | A1 | 266 | | Commercial | GWIXP455AAB |
| Intel® IXP455 | A1 | 533 | | Extended | GWIXP455AADT |
| Intel® IXP455 | A1 | 400 | | Extended | GWIXP455AACT |
| Intel® IXP455 | A1 | 266 | | Extended | GWIXP455AABT |

# Intel® IXP42X Product Line of Network Processors

**Table 3. Part Numbers for the Intel® IXP42X Product Line of Network Processors**

| Device | Stepping | Speed (MHz) | Extended Temp. | Lead Free | Part # |
|---|---|---|---|---|---|
| Intel® IXP425 | B-0 | 533 | Yes | Yes | EWIXP425BDT |
| Intel® IXP425 | B-0 | 266 | Yes | Yes | EWIXP425BBT |
| Intel® IXP425 | B-0 | 533 | | Yes | PRIXP425BD |
| Intel® IXP425 | B-0 | 400 | | Yes | PRIXP425BC |
| Intel® IXP425 | B-0 | 266 | | Yes | PRIXP425BB |
| Intel® IXP425 | B-0 | 533 | Yes | | GWIXP425BDT |
| Intel® IXP425 | B-0 | 400 | Yes | | GWIXP425BCT |
| Intel® IXP425 | B-0 | 266 | Yes | | GWIXP425BBT |
| Intel® IXP425 | B-0 | 533 | | | FWIXP425BD |
| Intel® IXP425 | B-0 | 400 | | | FWIXP425BC |
| Intel® IXP425 | B-0 | 266 | | | FWIXP425BB |
| Intel® IXP423 | B-0 | 266 | | Yes | PRIXP423BB |
| Intel® IXP423 | B-0 | 266 | | | FWIXP423BB |
| Intel® IXP422 | B-0 | 266 | | Yes | PRIXP422BB |
| Intel® IXP422 | B-0 | 266 | | | FWIXP422BB |
| Intel® IXP421 | B-0 | 266 | | Yes | PRIXP421BB |
| Intel® IXP421 | B-0 | 266 | | | FWIXP421BB |
| Intel® IXP420 | B-0 | 266 | Yes | Yes | EWIXP420BBT |
| Intel® IXP420 | B-0 | 266 | Yes | | GWIXP420BBT |
| Intel® IXP420 | B-0 | 533 | | Yes | PRIXP420BD |
| Intel® IXP420 | B-0 | 400 | | Yes | PRIXP420BC |
| Intel® IXP420 | B-0 | 266 | | Yes | PRIXP420BB |
| Intel® IXP420 | B-0 | 533 | | | FWIXP420BD |
| Intel® IXP420 | B-0 | 400 | | | FWIXP420BC |
| Intel® IXP420 | B-0 | 266 | | | FWIXP420BB |

# Non-Core Errata Descriptions

## 1. PCI Doorbell Register Does Not Work Properly (SCR 460)

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) — Fixed

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

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

## 2. UTOPIA Interface Status Collection Synchronization Issues (SCR 485)

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) — Fixed

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

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

## 3. No Byte Enables Asserted During a PCI I/O Read of the Intel® IXP425 Network Processor Causes Unpredictable Behavior (SCR 469)

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) — Fixed

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

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

### 4. Simultaneous AHB Access of the PCI Bus Controller (SCR 499)

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® Core cannot access the PCI bus by 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) — Fixed

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

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

### 5. 66-MHz PCI Operation (SCR 543)

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) — Fixed

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

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

### 6. Ethernet Control Protocol Frames Transmit-Defer Status Bit Error (SCR 472)

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) — No Fix

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

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

### 7. Logic 0 is Driven on Both the USB D+ and D- at Reset (SCR 545)

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) — Fixed

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

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

### 8. Cannot Generate Watchdog Timer Reset (SCR 641)

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) — Fixed

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

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

### 9. PCI Non-Prefetch Reads (SCR 1254)

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 processor's 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) — Fixed

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

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

## 10. Timer Status Interrupts Get Lost During MMR Writes (SCR 1653)

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 needs to acknowledge.
12. If (R8 > R5), then GP timer 0 expired: Software needs to acknowledge.
13. If (R9 > R6), then GP timer 1 expired: Software needs to acknowledge.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step) — No Fix

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

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

### 11. Character Time-Out Interrupt Sticks Under Certain Software Timing Conditions (SCR 2235)

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) — No Fix

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

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

## 12. Intel® IXP425 A-0 Step Processor May Have Problems Working With Some SDRAM Devices (SCR 2411)

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 24.

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) — Fixed

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

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

## 13. PCI DMA Lock-Up Condition (SCR 2372)

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 do 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) — No Fix

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

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

### 14. PCI Doorbell Register Lock-up Condition When Using Two Products Together that have Intel® IXP4XX Product Line of Network Processors (SCR 2379)

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) — No Fix

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

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

### 15. PCI Controller Returns Infinite Retries on PCI After AHB Pre-Fetch Error (SCR 1289)

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 above 256 Mbyte of SDRAM memory space.

Workaround: None.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step) — No Fix

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

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

### 16. UART Break Indicator (SCR 2929)

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) — No Fix

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

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

### 17. Intel XScale® Core Non-Branch Instruction in Vector Table (SCR 2871)

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 may 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) — No Fix

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

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

### 18. IRQ 3 Is Locking the System ‘Disable’ (SCR 2143)

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
```

Be advised that this workaround requires Intel XScale core cycles, so should be done carefully.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step) — No Fix

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

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

## 19. EX_IOWAIT_N Timing (SCR 3051)

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*. This occurs because there is a synchronizer on the EX_IOWAIT_N signal which 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 need to 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) — No Fix

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

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

## 20. SOF During Control Read Can Corrupt USB Transfer (SCR 1553)

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) — Fixed

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

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

## 21. USB - Invalid 0x81 Value in Endpoint 0 Control Register on SETUP Transfer (SCR 3077)

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) — No Fix

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

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

## 22. Ethernet Coprocessors — Ethernet Pad Enable Overrides Append FCS (SCR 299)

Problem: The IXP4XX product line processors have an Ethernet coprocessor that is configured by Intel XScale core 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) — No Fix

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

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

## 23. Ethernet Coprocessors — Length Errors on Received Frames (SCR 711)

Workaround: The IXP4XX product line processors have an Ethernet coprocessor that is configured by Intel XScale core 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) — No Fix

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

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

## 24. PCI DC Parameter VIH Marginality Issue (SCR 3121)

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) — No Fix

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

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

## 25. False PCI DMA Completion Notification Causing Data Corruption (SCR 3910)

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 core 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) — No Fix

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

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

## 26. Expansion Bus HPI Interface Potential for Contention on Reads with T4=0 (SCR 4117)

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) — No Fix

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

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

## 27. PCI Hangs With a Multiple Inbound Error Condition (SCR 4160)

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) — No Fix

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

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

## 28. PCI RCOMP Operation if PCI Clock Stops (SCR 4022)

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: 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) — No Fix

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

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

## 29. UART — Break Condition Asserted Too Early if Two Stop bits are Used (SCR 4092)

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) — No Fix

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

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

## 30. Ethernet Coprocessors — Address Filtering Logic Ignores the Second to Last Nibble of the Destination Address (SCR 4185)

Problem: Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor have an Ethernet coprocessor configured by Intel XScale core 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 7.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step) — No Fix

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

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

### 31. USB DC Parameter Vih Specification Change (SCR 3111)

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) — No Fix

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

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

### 32. Start of DMA Operation Close to Start of Non-Prefetch (NP) Operation Can Hang NP Operation on AHB (SCR 3939)

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) — No Fix

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

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

## 33. PCI Accesses to the Queue Manager During Queue and SRAM Mode (SCR 4076)

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) — No Fix

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

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

## 34. Ethernet MACs Detect Late Collision Earlier Than Ethernet 802.3 Specifications (SCR 4062)

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 need to be planned to not operate on the boundary of the Ethernet specifications.

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step) — No Fix

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

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

## 35. Read of PCI Controllers BAR 32'h XXFF_FFFC Rd[N] Corrupts Subsequent Rd[N+1] (SCR 3850)

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) — No Fix

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

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

### 36. UART Operating in Non-FIFO Mode Can Falsely Receive Overrun Error (SCR 1554)

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 XSCALE 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) — No Fix

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

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

### 37. USB-Specification Noncompliance for Rise/Fall Transition Times (SCR 3633)

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) — No Fix

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

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

## 38. Ethernet MAC Does Not Detect Transmit FIFO Underruns Reliably (SCR 4230)

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) — No Fix

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

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

## 39. SMII late_col Occurs Earlier Than Expected (SCR 4045)

Problem: When configured for SMII/SS-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 won't see the collision, and it won't know that it needs to 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/SS-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/SS-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) — N/A

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

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

## 40. Timer Issues with Prescale Programming Sequence and Pause/Resume Operation (SCR 4216)

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 needs to 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) — N/A

Status: IXP46X (A0-Step) — No Fix

Status: IXP46X (A1-Step) — Fixed

Status: IXP45X (A0-Step) — No Fix

Status: IXP45X (A1-Step) — Fixed

## 41. IEEE-1588 Time Sync Lock-up Fails to Time-Stamp a Second PTP Message (SCR 4239)

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) — N/A

Status: IXP46X (A0-Step) — No Fix

Status: IXP46X (A1-Step) — Fixed

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

# Core 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 XScale core Register* (MCR) command is outstanding. The core 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) — No Fix

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

Status: IXP45X (A0-Step and A1-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 core 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 core store instruction attempts to modify A, hits the cache, aborts because of MMU permissions, and is backed out of the cache. That line normally is not marked dirty, but because of this errata, is marked as dirty.
4. The cache line at A then ages out or is explicitly cleaned. The original data from location A is evicted to external memory, overwriting the data written by the external master. This only happens when software is allowing an external master to modify memory, that is, write-back or write-allocate in the core page tables, and, depending on the fact that the data is not dirty in the cache, to preclude the cached version from overwriting the external memory version. **When there are any semaphores or any other handshaking to prevent collisions on shared memory, this is not a problem.**

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

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step) — No Fix

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

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

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

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

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

Status: IXP45X (A0-Step and A1-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 core does not implement any CP15 ID code registers other than the Main ID register (opcode2 = 0b000) and the Cache Type register (opcode2 = 0b001). When any of the unimplemented registers are accessed by software (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) — No Fix

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

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

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

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

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

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

Status: IXP425 (A0-Step) — No Fix

Status: IXP42X (B0-Step) — No Fix

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

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

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

Problem: The IEEE 1149.1 specification states that the effects of updating all parallel JTAG registers should be seen on the falling edge of TCK in the Update-DR state. The Intel XScale core parallel JTAG registers 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) — No Fix

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

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

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

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

# Specification Changes

## 1. Increase Core Voltage from 1.4 V to 1.5 V for 667 MHz Operation only (SCR 4291)

Issue: Customers who are planning to go into production using Intel® IXP465 Network Processor operating at 667 MHz, should already be advised that the requirement for $V_{CC}$ core voltage has been increased from 1.4 V nominal to 1.5 V nominal, as published in the July 2005 Technical Advisory. All published IXP46X-related collateral (as of August 2005) have already been updated to reflect this increased $V_{CC}$ core voltage change, therefore, a summary of this Technical Advisory is being repeated in this Specification Update for historical purposes. For IXP465 and IXP460 designs operating at either 533 MHz, 400 MHz, or 266 MHz, the $V_{CC}$ core voltage remains unchanged at 1.3 V, as specified in the datasheet.

The IXP465 and IXP460 network processor at 667 MHz are expected to operate using a $V_{CC}$ core voltage of 1.5 V. The exact operating conditions for this 1.5 V core voltage has a specified range of 1.425 V minimum to 1.575 V maximum, as already documented in the *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Datasheet* (August 2005).

***Note:*** The voltages for $V_{CCOSC}$, $V_{CCPLL1}$, $V_{CCPLL2}$, and $V_{CCPLL3}$ must also operate at the same $V_{CC}$ core voltage of 1.5 V nominal, with a specified range of 1.425 V minimum to 1.575 V maximum.

The increase in $V_{CC}$ core voltage at 667 MHz at 1.5 V also increased the estimated maximum power dissipation of the IXP46X network processor to 4.0 Watts, from a previous level of 3.6 Watts at 1.4 V. This additional 400 mW power increase is created only by increased current consumption from the 1.5 V core voltage, with no increase in current from either the 2.5 V and 3.3 V supplies. Therefore, the power dissipation table in the *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Datasheet* (August 2005 version) have also been updated to reflect this increase in $V_{CC}$ core current consumption from $I_{CC}$ max of 1785 mA to $I_{CC}$ max of 1920 mA.

The updated table for power dissipation of the IXP465 network processor at 667 MHz is provided in the following table. Both power supply and thermal management designs should account for this increased voltage and current.

### Table 4. Power Dissipation Values

| Part Type | Power Rail | Icc (mA) | Power Per Rail (mW)† | Maximum Power Dissipation (Watts) |
|---|---|---|---|---|
| Intel® IXP46X Product Line — 667 MHz | 3.3 V | 88 | 305 | 4.0 |
| | 2.5 V | 255 | 669 | |
| | 1.5 V | 1920 | 3024 | |
| † Power in mW is calculated using Maximum Vcc specification for each power rail. | | | | |

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Datasheet* (Aug. 2005)

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual* (Aug. 2005)

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Hardware Design Guidelines* (Aug. 2005)

Affected Docs: *Intel® IXDP465 Development Platform Quick Start Guide* (Aug. 2005)

Affected Docs: *Intel® IXDP465 Development Platform User's Guide* (Aug. 2005)

## 2. PCI ID Changes to Support A1 Stepping of IXP46X (No SCR)

Issue: The pci_revision_id(7:0) register defined in the *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual* Section 10.5.2.3 (Class Code / Revision ID Register) has changed to support the latest IXP46X Silicon A1 Revision (stepping), as follows:

From:

— Bits 7:0 = RevisionID = Silicon Revision for this device = Reset Value = 0x00

To:

— Bits 7:0 = RevisionID = Silicon A0 Revision for this device = Reset Value = 0x00

— Bits 7:0 = RevisionID = Silicon A1 Revision for this device = Reset Value = 0x01

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual* (Aug. 2005)

## 3. External Crystal Support is No Longer Supported with IXP42X (SCR 4271)

Issue: External crystal support for the IXP42X processor oscillator clock input is no longer supported, identical to the non-support of crystals for the Intel® IXP46X Product Line of Network Processors. Therefore, all references to crystal-related specifications and usages contained in IXP42X- related collateral must be removed.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (Mar. 2005)

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

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Hardware Design Guidelines* (Jun. 2004)

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Design Checklist* (Jul. 2004)

Affected Docs: *Intel® IXP42X Product Line of Network Processors: Crystal Design Considerations Application Note* (Jan. 2005) - Discontinued

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

Affected Docs: *Intel® IXDP425 / IXCDP1100 Development Platform Quick Start Guide* (Jul. 2004)

Affected Docs: *Intel® IXDP425 / IXCDP1100 Development Platform Documentation Kit*

## 4. IXC1100 Control Plane Processors Will Be Discontinued (SCR 4317)

Issue: Shipments of IXC1100 Control Plane Processor are being discontinued as of October 15, 2005. Therefore, all references to the IXC1100 Product specifications and usages 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.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (Mar. 2005)

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

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Hardware Design Guidelines* (Jun. 2004)

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Design Checklist* (Jul. 2004)

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

Affected Docs: *Intel® IXDP425 / IXCDP1100 Development Platform Quick Start Guide* (Jul. 2004)

Affected Docs: Intel® IXDP425 / IXCDP1100 Development Platform *Documentation Kit*

# Specification Clarifications

## 1. PCI Clock Slew Rate (SCR 4285)

Issue: The PCI standard does not define clock rise/fall time, however it does have a requirement for the Slew Rate. Therefore, a change must be made to Table 43 (PCI Clock Timings) in the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet*.

**FROM:**

### PCI Clock Timings

| Symbol | Parameter | 33 MHZ | | 66 MHZ | | Units | Notes |
|---|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | | |
| $T_{PERIODPCICLK}$ | Clock period for PCI Clock | 30 | | 15 | | ns | |
| $T_{CLKHIGH}$ | PCI Clock high time | 11 | | 6 | | ns | |
| $T_{CLKLOW}$ | PCI Clock low time | 11 | | 6 | | ns | |
| $T_{RISE/FALL}$ | Rise and fall time requirements for PCI Clock | | 2 | | 2 | ns | |

**TO:**

### PCI Clock Timings

| Symbol | Parameter | 33 MHZ | | 66 MHZ | | Units | Notes |
|---|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | | |
| $T_{PERIODPCICLK}$ | Clock period for PCI Clock | 30 | | 15 | | ns | |
| $T_{CLKHIGH}$ | PCI Clock high time | 11 | | 6 | | ns | |
| $T_{CLKLOW}$ | PCI Clock low time | 11 | | 6 | | ns | |
| $T_{SLEW\ RATE}$ | Slew Rate requirements for PCI Clock | 1 | 4 | 1.5 | 4 | V/ns | |

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (Mar. 2005)

## 2. Ethernet Tx and Rx Clocks Must Be Specified as +/- ppm Frequency Tolerance (SCR 4222)

Issue: The MII standard specifies the Tx and Rx Ethernet clocks in terms +/-100 ppm frequency tolerance, and not as a $T_{RISE/FALL}$ parameter. The 2 ns $T_{RISE/FALL}$ time in the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* has been a misleading design parameter and is therefore being replaced with a specified +/- ppm frequency tolerance as follows:

In Section 5.5.1.3, Table 42 (MII Clock timings), the 2 ns $T_{RISE/FALL}$ time parameter is removed, and replaced with:

— Symbol = Frequency Tolerance

— Parameter = Frequency tolerance requirement for Tx and Rx Ethernet clocks

— Nominal = +/- 50

— Maximum = +/- 100

— Units = ppm

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (Mar. 2005)

### 3. PCI Theoretical Peak Data Rate Should be Removed (SCR 4237)

Issue: In Section 1.5 (Key Features), under PCI Interface, the last sub-bullet "High-performance support for 264-Mbps peak data transfers" should be removed because it is a theoretical limit of the PCI bus.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Hardware Design Guidelines* (Jun. 2004)

### 4. PCI_CLKIN Pull-Down Recommendation (SCR 4238)

Issue: The *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (Mar. 2005) recommends use of a 10 KΩ pull-down resistor when the PCI_CLKIN signal is not being used. The Design Checklist also has this recommendation, but the Design Guide does not, and should therefore be updated to add the following for PCI_CLKIN:

— PCI_CLKIN should be pulled low with a 10 KΩ resistor when not being used in the system.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Hardware Design Guidelines* (Jun. 2004)

### 5. PCI_IDSEL Pull-up Recommendation (SCR 4244 and SCR 4260)

Issue: The *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (Mar. 2005) was updated to recommend use of a 10 KΩ pull-up resistor when the PCI_IDSEL signal is not being used, instead of a pull-down resistor. Both Design Guide and Design Checklist should be updated to recommend this same 10 KΩ pull-up resistor, when the PCI_IDSEL is not being used.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Hardware Design Guidelines* (Jun. 2004)

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Design Checklist* (Jul. 2004)

### 6. Ethernet Clock to Output Delay of 0 ns to be Removed (SCR 4257)

Issue: In Section 5.5.2.4 (MII), Table 48 (MII Output Timing Values), the specified minimum of 0 nsec for T1 (Clock to output delay for ETH_TXDATA and ETH_TXEN) needs to be removed.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (Mar. 2005)

### 7. XScale Core Speed Strapping Table to be Updated (SCR 4261)

Issue: Table 10 (Setting the Intel XScale® Core Operation Speed) in the Design Checklist should be identical to Table 125 (Intel XScale® Core Speed Expansion Bus Configuration Strappings) in the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* (Mar. 2005).

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Design Checklist* (Jul. 2004)

## 8. RCOMP Pin Type Should be an Output, Not Input (SCR 4298)

Issue: In Table 18 (System Interface) of the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet*, the RCOMP Pin Type is specified as an Input, but it should be changed to an Output.

In Figure 20 (Devices' Signals by Function) of the Design Guide, the RCOMP signal is illustrated as an Input, but it should be changed to an Output.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (Mar. 2005)

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Hardware Design Guidelines* (Jun. 2004)

## 9. Impedance Recommendation Should be Consistent at 50 Ω (SCR 4307)

Issue: The Platform Users Guide specifies a 65 Ω impedance requirement in 2 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)

## 10. Expansion Bus Configurations Straps that are Reserved (SCR 4308)

In the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual*, Section 8.9.9 (Configuration Register 0), Table 124, the Expansion Bus Strapping bits 20:17 are defined as User-configurable and bits Bits 16:5 are defined as (Reserved), although many of these (Reserved) bits are pre-assigned as configuration bits in the Sept. 2004 IXDP425 User's Guide, Table 9. A note needs to be added to the bottom of Table 124 in the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* to clarify the use of the word (Reserved), as follows:

***Note:*** Bits that are (Reserved) should remain reserved because they are assigned to unique devices used on Intel's Development Platform Designs, resulting in unique software Board Support Packages and Device Drivers that interface with these unique devices. These (Reserved) bits should also remain reserved because they will be assigned to future steppings of the IXP4xx family of Network Processors.

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

## 11. PCI Initiation Clock Cycle Range is Incorrect (SCR 4309)

Issue: In Section 6.2 (PCI Controller Configured as Option) in the fifth paragraph, the following sentence needs to be changed:

From: If initialization is not completed in the first 2 to 25 PCI clocks

To: If initialization is not completed in the first $2^{25}$ PCI clocks after

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

## 12. IOWAIT Functional Description Update for IXP42X (SCR 4314)

Issue: The functional description in Section 8.6 (Using I/O Wait) requires additional text and diagrams, similar to the updates made to the corresponding sections in the *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Datasheet* and *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual*.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (Mar. 2005)

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

## 13. OSC_VSSP Pin Assignment Contradiction (SCR 4326)

Issue: Section 5.2.4 (OSC_VCCP Requirement) states that OSC_VSSP is made up with two pins, AD10 and AF10. This is incorrect and should be changed to read AE11 and AE12.

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Datasheet* (Aug. 2005)

## 14. Memory Controller Maximum Throughput Clarification (SCR 4327)

Issue: In the second paragraph of Section 11.1 (Memory Controller Overview), the DDRI maximum throughput should be fixed to agree with the format used in other sections of the manual, as follows:

From:

The DDRI SDRAM interface consists of a 32-bit wide data path to support up to 1.066 Gbps.

To:

The DDRI SDRAM interface consists of a 32-bit wide data path to support up to 1066 Mbyte/sec.

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual* (Aug. 2005)

## 15. Memory Controller Refresh Counter Bit Width Clarification (SCR 4328)

Issue: In Section 11.6.14 (Refresh Frequency Register RFR), Bits 12:00 need to be changed to Bits 11:00 in the register table, because the Refresh Counter is 12 bits wide, per Section 11.2.1.5.

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual* (Aug. 2005)

## 16. Clarification on Number of MII (Ethernet) Ports for IXP460 (SCR 4331)

Issue: In Section 1.2 (Model-Specific Features), in Table 1, an "X††" should be inserted into the IXP460 column within the row titled "MII/SMII/4-Port SMII (NPE B)", because the MII interfaces for both NPE B and NPE C are enabled for the IXP460 product.

The †† footnote in Table 1 should also be updated as follows:

†† 4-Port SMII is not supported on the IXP455 or IXP460 network processor.

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Datasheet* (Aug. 2005)

## 17. PCI to South AHB Address Error (SCR 4322)

Issue: There is a South AHB address error that is printed in the PCI Controller Section 10.2.4.1, in the fifth numbered step as follows:

5. PCI I/O space example is an external PCI device initiates a PCI bus transfer to BAR5 of the IXP45X/IXP46X network processors. The PCI address looks like the following PCI Address = 0xA5123418. The address placed on the South AHB is 0xA1200018.

The last sentence in this paragraph (above) should be changed to:

— The address placed on the South AHB is 0x0A120018.

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual* (Aug. 2005)

## 18. APB Clocks Incorrectly Defined (No SCR)

Issue: There are a few occurrences across both the *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Datasheet* and the *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual* where the Advanced Peripheral Bus (APB) is specified as a 66 MHz 32-bit bus. It is actually a 66.66 MHz 32-bit bus (2 * OSC_IN input pin). All occurrences of 66 MHz should therefore be changed to 66.66 MHz, as follows:

*Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Datasheet*

- Section 3.0 (Functional Overview), Figure 2, change APB to 66.66 MHz

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

- Section 2.0 (Functional Overview), Figure 2, change APB to 66.66 MHz
- Section 5.0 (Internal Buses), eighth paragraph, change APB to 66.66 MHz

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Datasheet* (Aug. 2005)

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual* (Aug. 2005)

## 19. IXP42X Power Clarification for Total $I_{CC}$ in Power Tables (No SCR)

Issue: In Section 5.7 ($I_{CC}$ and Total Average Power), the $I_{CC}$ and $I_{CCP}$ symbols used in Tables 67 and 68 are misleading, because these symbols also include the total currents from other IXP42X pins.

Therefore, the following clarifications need to be made to both Tables 67 and 68:

— All occurrences of $I_{CC}$ should be changed to $I_{CC\_TOTAL}$

— All occurrences of $I_{CCP}$ should be changed to $I_{CCP\_TOTAL}$

— Add the following two Notes:

$I_{CC\_TOTAL}$ includes total current from $V_{CC}$, $V_{CCOSC}$, $V_{CCPLL1}$, and $V_{CCPLL2}$

$I_{CCP\_TOTAL}$ includes total current from $V_{CCP}$ and $V_{CCOSCP}$

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (Mar. 2005)

## 20. IXP46X Power Clarification for Total $I_{CC}$ in Power Table (No SCR)

Issue: In Section 5.8 (Power Dissipation), the $I_{CC}$ (mA) symbol used in Power Table 83 is misleading, because this symbol also includes the currents from other IXP46x pins.

Therefore, the following clarifications need to be made to Table 83:

— All occurrences of $I_{CC}$(mA) should be changed to $I_{CC\_TOTAL}$ (mA)

— Add the following three Notes:

$I_{CC\_TOTAL}$ for 3.3 V includes total current from $V_{CCP}$ and $V_{OSC\_VCCP}$

$I_{CC\_TOTAL}$ for 2.5 V includes total current from $V_{CCM}$

$I_{CC\_TOTAL}$ for 1.3 V includes total current from $V_{CC}$, $V_{OSC\_VCCP}$, $V_{CCPLL1}$, $V_{CCPLL2}$, and $V_{CCPLL3}$

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Datasheet* (Aug. 2005)

## 21. Memory Controller Clarification on why 8 ECC bits are supported (SCR 4336)

Issue: In Section 3.1.8 (DDRI SDRAM Controller), the last two sentences in the third paragraph should be changed as follows:

From:

The controller supports the 8 bits due to the fact that internally it is a 32-bit or 64-bit controller. However, this implementation of the controller only supports 32 bits.

To:

The controller supports the 8 bits of ECC due to the fact that the controller is 64-bits internally. ECC requires 8 bits for implementation over a 64-bit interface and only 7 bits for a 32-bit interface. However, ECC is only implemented in the 32-bit mode of operation.

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Datasheet* (Aug. 2005)

## 22. $T_{POWER\_UP}$ Sequence Needs to be Fixed (SCR 4337)

Issue: In Section 5.7 (Power Sequence), there is a sentence that needs to be changed as follows:

From:

The value for $T_{POWER\_UP}$ must be at least 1 µs before the later of $V_{CCP}$ and $V_{CCM}$.

To:

The value for $T_{POWER\_UP}$ must be at least 1 µs after the later of $V_{CCP}$ and $V_{CCM}$ reaching stable power.

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Datasheet* (Aug. 2005)

# Documentation Changes

## 1. Unused Acronyms Need to be Removed from List (SCR 4274)

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

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

## 2. Memory Controller Cross Reference Error (SCR 4329)

Issue: In Section 11.2.1.2 (Address Decode Blocks), there is a cross-reference that needs to be changed:

- From: (refer to Section 12.5.9, "Configuration Register 0")
- To: (refer to Section 12.5.10, "Configuration Register 1")

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual* (Aug. 2005)

## 3. Memory Controller Text Clarifications in 6 Places (SCR 4330)

Issue: There are several text clarifications required in Section 11 (Memory Controller), as follows:

- Section 11.2.2.2 (DDR1 SDRAM Bank Sizes and Configurations), Table 202:
  Change the 512 Mbit arrangement from "6M x 8" to "64M x 8".
- Section 11.2.2.2 (DDR1 SDRAM Bank Sizes and Configurations), Equations 2 and 3:
  Change Equations 2 and 3 to Equations 1 and 2, because there is no Equation 1.
- Section 11.2.2.11 (DDRI SDRAM Write Cycle), last sentence in this section:
  Change "Figure 116 for random read memory" to "Figure 116 for random write memory".
- Section 11.4 (Interrupts/Error Conditions), Table 213:
  Change the two instances of "SDCR" to "ECCR".
- Section 11.6.9 (ECC Address Registers ECAR0, ECAR1),
  For Bits 01:00, change the default from 000 to 00, because this is only 2 bits.
- Section 11.6.15 (SDRAM Page Registers SDPR0-7):
  For Bit 00, change the default from 00 to 0, because this is only 1 bit.

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual* (Aug. 2005)

## 4. HSS Figure and Table Text Clarifications (SCR 4333)

Issue: There are several text clarifications required in the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual*, Section 17 (HSS Coprocessor), as follows:

- In Section 17.5.2 (Overview of HSS Clock Configuration), Table 156, Note 1 change:
  From: These HSS Tx/Rx clock output frequencies are based on the set-up parameters in 1
  To: These HSS Tx/Rx clock output frequencies are based on the set-up parameters in Table 155.
- In Section 17.6.3.2 (4.096-Mbps Backplane), Figure 90:

The arrows for bit allocation of timeslots 0a and 0b have the middle arrow pointed at the wrong bit, resulting in Timeslot 0a containing 9 bits and timeslot 0b containing 7 bits. The middle arrow should be moved so that timeslot 0b begins at bit 0.

— In Section 17.6.3.2 (4.096-Mbps Backplane), Figure 91:

Timeslot Xa contains 9 bits (instead of 8 bits), therefore, one X in the serial bit stream needs to be deleted from this figure.

Similar text clarifications are also required in the *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual*, Section 13 (HSS Coprocessor), as follows:

— In Section 13.4.2 (Overview of HSS Clock Configuration), Table 238, Note 1 change:

From: These HSS Tx/Rx clock output frequencies are based on the set-up parameters in 1

To: These HSS Tx/Rx clock output frequencies are based on the set-up parameters in Table 237.

— In Section 13.5.4.2 (4.096-Mbps Backplane), Figure 180:

The arrows for bit allocation of timeslots 0a and 0b have the middle arrow pointed at the wrong bit, resulting in Timeslot 0a containing 9 bits and timeslot 0b containing 7 bits. The middle arrow should be moved so that timeslot 0b begins at bit 0.

— In Section 13.5.4.2 (4.096-Mbps Backplane), Figure 181:

Timeslot Xa contains 9 bits (instead of 8 bits), therefore, one X in the serial bit stream needs to be deleted from this figure.

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

Affected Docs: *Intel® IXP45X and Intel® IXP46X Product Line of Network Processors Developer's Manual* (Aug. 2005)