

Intel® IXP43X Product Line of Network Processors

Specification Update

December 2008

Order Number: 316847; Revision: 005US



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

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined.” Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. See http://www.intel.com/products/processor_number for details.

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

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting Intel's Web Site.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino logo, Core Inside, FlashFile, i960, InstantIP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, Itanium, Itanium Inside, MCS, MMX, Oplus, OverDrive, PDCharm, Pentium, Pentium Inside, skoool, Sound Mark, The Journey Inside, VTune, Xeon, and Xeon Inside are trademarks of Intel Corporation in the U.S. and other countries.

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

Copyright © 2008, Intel Corporation. All rights reserved.



Contents

1.0 Preface 5
1.1 Affected Documents/Related Documents 5
1.2 Nomenclature 5

2.0 Summary Table of Changes 6
2.1 Codes Used in Summary Tables 6
2.1.1 Stepping 6
2.1.2 Page 6
2.1.3 Status 6
2.1.4 Row 6
2.2 Non-Intel XScale® Technology Errata 7
2.3 Intel XScale® Technology Errata 8
2.4 Specification Changes 8

3.0 Identification Information 9
3.1 Markings 9
3.2 Part Numbers (Lead free packaging) 10
3.2.1 Intel® IXP43X Product Line of Network Processors 10

4.0 Non-Intel XScale® Technology Errata Descriptions 11

5.0 Intel XScale® Technology Errata Descriptions 18

6.0 Specification Changes 22

Figures

1 Package Markings:
Intel® IXP43X Product Line of Network Processors —
Commercial / Extended Temperature, Lead-Free / Compliant with Standard for Restriction on the Use of Hazardous Substances (RoHS) 9
2 Define DDR_DCALCSR and DDR_DCALADDR Registers 23
3 EMRS Code Example 23

Tables

1 Part Numbers for the Intel® IXP43X Product Line of Network Processors 10
2 Part Numbers for the *Intel® IXP435 Multi-Service Residential Gateway Reference Platform* 10
3 Tested Devices that Pass on the Intel® IXP43X Product Line of Network Processors 16

Revision History

Date	Revision	Description
December 2008	005	• Added Specification Change No. 6 - Updates to DDR2 Setup/Hold Timing Values • Moved Non-Intel XScale® Technology No. 23 and 24 from Section 5.0 to Section 4.0 ***Note:*** Updated IXP43X documents affected by the March 2008 specification update (316847-004); the affected IXP43X documents include: - *Intel® IXP43X Product Line of Network Processors Datasheet* (316842) - *Intel® IXP43X Product Line of Network Processors Developer's Manual* (316843) - *Intel® IXP43X Product Line of Network Processors Design Guidelines* (316844) - *Intel® IXP43X Product Line of Network Processors Design Checklist* (316849)
March 2008	004	• Updated Figure 1 • Added Non-Intel XScale® Technology Errata 22 under Section 2.2 • Updated 4, 5, 6, and 7 specification clarifications (see Section 6.0, "Specification Changes" on page 22) • Added 8-18 specification clarifications (see Section 2.4, "Specification Changes" on page 8)
December 2007	003	• Added Non-Intel XScale® Technology Errata 21 under Section 2.2 • Added Intel XScale® Technology Errata 9 and 10 under Section 2.3 • Added "Specification Changes" on page 22 • Added Table 2 under Section 3.2.1
September 2007	002	Added Non-Intel XScale® Technology Errata 19 and 20 under Section 2.2
April 2007	001	Initial release of the document — The first specification update combining errata for the Intel® IXP43X Product Line of Network Processors

§ §



1.0 Preface

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

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.

Refer to the *Intel® IXP43X Product Line of Network Processors Datasheet* for detailed information on various features listed by processor.

1.1 Affected Documents/Related Documents

Title	Document Number
Intel® IXP43X Product Line of Network Processors Datasheet	316842
Intel® IXP43X Product Line of Network Processors Developer's Manual	316843
Intel® IXP43X Product Line of Network Processors Hardware Design Guidelines	316844
Intel® IXP43X Product Line of Network Processors Design Checklist	316849
Intel® IXP43X Product Line of Network Processors: Migrating from the Intel® IXP42X Product Line Application Note	316845

1.2 Nomenclature

Errata are design defects or errors. These may cause the behavior of Intel® IXP43X Product Line of Network Processors' to deviate from the published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.

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

Specification Clarifications describe a specification in greater detail or further highlight the impact of a specification 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 lifecycle of the product, or until a particular stepping is no longer commercially available. Under these circumstances, errata removed from the specification update are archived and available upon request. Specification changes, specification clarifications and documentation changes are removed from the specification update when the changes are made to the appropriate product specification or user documentation such as datasheets, manuals, and so on.

§ §

2.0 Summary Table of Changes

The following tables indicate the errata, specification changes, specification clarifications, or documentation changes that apply to the Intel® IXP43X Product Line of Network Processors. Intel may fix some of the errata in a future stepping of the component, and account for the other outstanding issues through documentation or specification changes as noted.

The summary tables use the following notations:

2.1 Codes Used in Summary Tables

2.1.1 Stepping

X:	This erratum exists in the stepping indicated. Specification Change or Clarification that applies to this stepping.
(No mark) or (Blank box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.
N/A	This erratum or document update is not applicable to this stepping of the silicon

2.1.2 Page

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

2.1.3 Status

Doc:	Document change or update will be implemented.
Fix:	This erratum is intended to be fixed in a future stepping of the component.
Fixed:	This erratum has been fixed for all steppings that are still being shipped.
No Fix:	This erratum has no plans of being fixed.
Plan Fix:	This erratum may be fixed in a future stepping of the product.
Varies:	This erratum applies to multiple steppings or devices and the erratum status varies between all steppings *that are still being shipped*.

2.1.4 Row

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



2.2 Non-Intel XScale® Technology Errata

Errata No.	Steppings		Page	Status	Errata
	Intel® IXP43X Product Line of Network Processors A0	Intel® IXP43X Product Line of Network Processors A1			
1	X	X	11	No Fix	HDLC Coprocessor Is Unable to Capture the Alignment Error on a Specific Frame Pattern
2	X	X	11	No Fix	PCI Accesses to the Queue Manager During Queue and SRAM Mode
3	X	X	11	No Fix	Ethernet Control Protocol Frames Transmit-Defer Status Bit Error
4	X	X	11	No Fix	PCI Doorbell Register Lock-up Condition When Using Two Products Together that have Intel® IXP4XX Product Line of Network Processors
5	X	X	12	No Fix	Intel XScale® Processor Non-Branch Instruction in Vector Table
6	X	X	12	No Fix	EX_IOWAIT_N Timing
7	X	X	12	No Fix	Ethernet Coprocessors — Ethernet Pad Enable Overrides Append FCS
8	X	X	12	No Fix	Ethernet Coprocessors — Length Errors on Received Frames
9	X	X	13	No Fix	False PCI DMA Completion Notification Causing Data Corruption
10	X	X	13	No Fix	PCI Hangs With a Multiple Inbound Error Condition
11	X	X	13	No Fix	UART — Break Condition Asserted Too Early If Two Stop Bits are Used
12	X	X	14	No Fix	Ethernet Coprocessors — Address Filtering Logic Ignores the Second to Last Nibble of the Destination Address
13	X	X	14	No Fix	Ethernet MACs Detect Late Collision Earlier Than Ethernet 802.3 Specifications
14	X	X	14	No Fix	Read of PCI Controllers BAR 32'h XXFF_FFFC Rd[N] Corrupts Subsequent Rd[N+1]
15	X	X	14	No Fix	UART Operating in Non-FIFO Mode Can Falsely Receive Overrun Error
16	X	X	15	No Fix	Ethernet MAC Does Not Detect Transmit FIFO Under-runs Reliably
17	X		15	Fixed	Expansion Bus Bootup Failure
18	X		15	Fixed	Incorrect Reset Value of GPIO_INR[0] and GPIO_ISR[0]
19	X	X	15	No Fix	USB Under-run errors when both USB ports perform Interrupt OUT transfers.
20	X	X	16	No Fix	Incorrect Inter-Packet Delay on USB interface.
21	X	X	16	No Fix	USB 2.0 Device Reset during Bulk Out Transaction
22	X	X	16	No Fix	RX FIFO Overflow Condition Not Properly Captured in the Rx Status Register
23	X	X	17	No Fix	Incorrect Strapping Configuration during Bootup
24	X	X	17	No Fix	Strong Pull-up Resistor Attached to the JTG_TRST_N

2.3 Intel XScale® Technology Errata

Errata No.	Steppings		Page	Status	Errata
	Intel® IXP43X Product Line of Network Processors A0	Intel® IXP43X Product Line of Network Processors A1			
1	X	X	18	No Fix	Abort is Missed When Lock Command is Outstanding
2	X	X	18	No Fix	Aborted Store that Hits the Data Cache May Mark Write-Back Data as 'Dirty'
3	X	X	18	No Fix	Performance Monitor Unit Event 0x1 Can be Incremented Erroneously by Unrelated Events
4	X	X	19	No Fix	In Special Debug State, Back-to-Back Memory Operations — Where the First Instruction Aborts — May Cause a Hang
5	X	X	19	No Fix	Accesses to the CP15 ID Register With Opcode2 > 0b001 Returns Unpredictable Values
6	X	X	20	No Fix	Disabling and Re-Enabling the MMU Can Hang the Processor or Cause It to Execute the Wrong Code
7	X	X	20	No Fix	Updating the JTAG Parallel Registers Requires an Extra TCK Rising Edge
8	X	X	20	No Fix	Non-Branch Instruction in Vector Table May Execute Twice After a Thumb Mode Exception

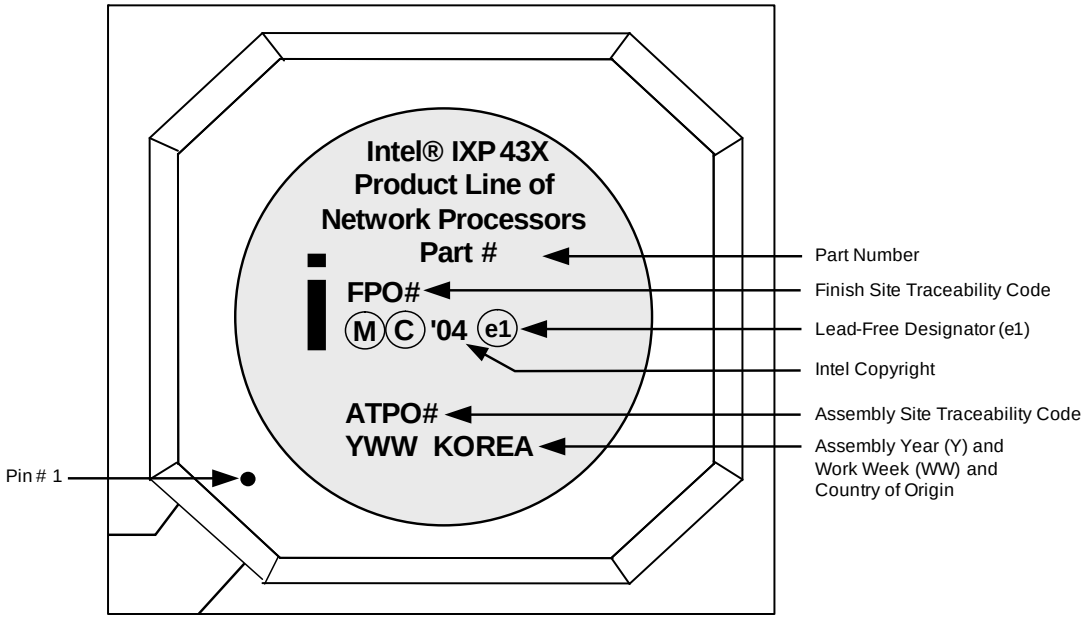
2.4 Specification Changes

Errata No.	Steppings		Page	Status	Errata
	Intel® IXP43X Product Line of Network Processors A0	Intel® IXP43X Product Line of Network Processors A1			
1	X	X	22	Doc	Updates to Pull-Down Strapping resistor value
2	X	X	22	Doc	Updates to Pull-Down Resistor Value on JTG_TRST_N
3	X	X	22	Doc	IEEE* 1588 Hardware Assistance Support
4	X	X	22	Doc	Turbo-MII Mode Support
5	X	X	22	Doc	Add Procedure to Issue EMRS OCD Command during DDR SDRAM Initialization
6	X	X	23	Doc	Updates to DDR2 Setup/Hold Timing Values

3.0 Identification Information

3.1 Markings

Figure 1. Package Markings: Intel® IXP43X Product Line of Network Processors — Commercial / Extended Temperature, Lead-Free / Compliant with Standard for Restriction on the Use of Hazardous Substances (RoHS)[1]



Notes:
1. **Part Number** field — For different part numbers of the Intel® IXP43X Product Line of Network Processors, see Table 1.
2. Package ball counts — Intel® IXP43X Product Line of Network Processors has a ball count of 460.
3. Drawing is not to scale. Marking content is an example.

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

3.2 Part Numbers (Lead free packaging)

3.2.1 Intel® IXP43X Product Line of Network Processors

Table 1. Part Numbers for the Intel® IXP43X Product Line of Network Processors

Device	Stepping	Speed (MHz)	Part Number	Temperature Offering
Intel® IXP435 Network Processor	A1	667	NHIXP435AE	Commercial
Intel® IXP435 Network Processor	A1	667	NHIXP435AET	Extended
Intel® IXP435 Network Processor	A1	533	NHIXP435AD	Commercial
Intel® IXP435 Network Processor	A1	533	NHIXP435ADT	Extended
Intel® IXP435 Network Processor	A1	400	NHIXP435AC	Commercial
Intel® IXP435 Network Processor	A1	400	NHIXP435ACT	Extended
Intel® IXP433 Network Processor	A1	533	NHIXP433AD	Commercial
Intel® IXP432 Network Processor	A1	400	NHIXP432AC	Commercial
Intel® IXP431 Network Processor	A1	400	NHIXP431AC	Commercial
Intel® IXP430 Network Processor	A1	667	NHIXP430AE	Commercial
Intel® IXP430 Network Processor	A1	533	NHIXP430AD	Commercial
Intel® IXP430 Network Processor	A1	400	NHIXP430AC	Commercial
Intel® IXP430 Network Processor	A1	400	NHIXP430ACT	Extended

Table 2. Part Numbers for the *Intel® IXP435 Multi-Service Residential Gateway Reference Platform*

Platform	Revision	Speed (MHz)	Part Number	MM#
Intel® IXP435 Multi-Service Residential Gateway Reference Platform	1.21	667	KIXRP435	884168

§ §



4.0 Non-Intel XScale® Technology Errata Descriptions

1. HDLC Coprocessor Is Unable to Capture the Alignment Error on a Specific Frame Pattern

Problem: Based on ITU-T Q.921, during transmission, a bit stuffing ‘0’ is inserted after five consecutive ‘1’ bits (including the FCS field) and then removed at the receiving end. The bit stuffing ensures that the data does not appear as the ‘end of frame’ flag (01111110). If five ‘1’s are received without a bit stuffing ‘0’, a ‘byte alignment error’ will be issued. The HDLC controller of the IXP43X product line of network processors will not issue a ‘byte alignment error’ when the following boundary condition occurs:

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

   AND

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

Implication: An HDLC frame received by the IXP43X product line of network processors HDLC controller from a Q.921 compliant HDLC controller with the same pattern described above will not generate a ‘byte alignment error’; an FCS error will be issued by the HDLC controller.

Workaround: None

Status: No Fix.

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

Problem: Under certain data traffic, the PCI controller may generate spurious write transfers and may return incorrect data on reads while 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 as these accesses cause pre-fetches during reads.

Implication: Pre-fetches will cause loss of queue data.

Workaround: Do not use the Queue Manager's SRAM mode during PCI accesses. Instead use the DDRII/DDRI SDRAM memory space while generating PCI accesses to the memory space of the IXP43X network processors. An external PCI master must use PCI BAR5 while accessing the Queue Manager in Queue mode.

Status: No Fix.

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

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

Implication: The Transmit-Defer Status bit in the IXP43X network processors is unusable.

Workaround: None.

Status: No Fix.

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

Problem: It is possible that the PCI bus can get in a locked condition when multiple products using the 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 occurs only when both the IXP4XX product line processors attempt to access each other’s PCI doorbell register at a particular instance. This error occurs only on reads of the of the doorbell register.

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

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

Status: No Fix.

5. Intel XScale® Processor Non-Branch Instruction in Vector Table

Problem: If an exception occurs in thumb mode and a non-branch instruction is executed at the corresponding exception vector, that instruction may execute twice. 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: Instructions 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 there will not be any incorrect behavior. If a branch instruction is placed at the beginning of the handler, it will not be executed twice.

Status: No Fix.

6. EX_IOWAIT_N Timing

Problem: There are two problems with the functionality of the expansion bus IOWAIT protocol. If both T2 and T3 are programmed to be 0 (normal timing), the expansion bus controller will not extend the T3 data state as described in Section 12.4.1.5 “Using I/O Wait” on page 572, of the *Intel® IXP43X Product Line of Network Processors Developer’s Manual - Rev. 001* (April 2007). This occurs because there is a synchronizer on the EX_IOWAIT_N signal that causes the expansion bus controller to transition to the T4 state before the EX_IOWAIT_N is detected.

Additionally, the *Intel® IXP43X Product Line of Network Processors 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 it waits for the T3 count to expire before proceeding to T4.

Implication: The expansion bus will not extend the T3 data state as shown in Figure 141 of the *Intel® IXP43X Product Line of Network Processors Developer’s Manual - Rev. 001* (April 2007).

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

Status: No Fix.

7. Ethernet Coprocessors — Ethernet Pad Enable Overrides Append FCS

Problem: The Intel® IXP43X Product Line of Network Processors has an Ethernet coprocessor that is configured by the Intel® IXP400 Software. 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: No Fix.

8. Ethernet Coprocessors — Length Errors on Received Frames

Workaround: The Intel® IXP43X Product Line of Network Processors has an Ethernet coprocessor that is configured by the Intel® IXP400 Software. 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 (Receive Control 1 Register, bit-1) 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: No Fix.

9. False PCI DMA Completion Notification Causing Data Corruption

Problem: The PADC1, PADC0, APDC1, and APDC0 complete bits in the PCI_DMACTRL register will not be cleared under certain conditions when the Intel XScale® Processor performs a write-1-to-clear (W1C) 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 not have been cleared).

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

Workaround: Following two workarounds are 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: No Fix.

10. PCI Hangs With a Multiple Inbound Error Condition

Problem: The PCI controller may lock up if there are multiple errors occurring around two different inbound PCI transactions.

A lock-up will occur when:

1. An inbound PCI read that targets an internal slave such as the expansion bus or Queue Manager results in an AHB error that occurs due to the PCI controller generating an illegal AHB transfer type on the target.

AND

2. 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.

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 is logged (PCI_SRCR.DPE = 1), and the PCI controller retries every inbound transaction; the system board must reset the IXP43X network processors.

Status: No Fix.

11. UART — Break Condition Asserted Too Early If Two Stop Bits are Used

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

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

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

Workaround: Do not use two stop bits.

Status: No Fix.

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

Problem: The Intel® IXP43X Product Line of Network Processors has an Ethernet coprocessor configured by the Intel® IXP400 Software. 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 high nibble first and then the low nibble.

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

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

Status: No Fix.

13. Ethernet MACs Detect Late Collision Earlier Than Ethernet 802.3 Specifications

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

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

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

Status: No Fix.

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

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

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

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

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

Status: No Fix.

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

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

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

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



Status: No Fix.

16. Ethernet MAC Does Not Detect Transmit FIFO Under-runs Reliably

Problem: There is a transmit race condition in the 10/100 Ethernet MAC. The race condition happens when the 10/100 Ethernet MACs transmit FIFO is filled on the same transmit clock as the transmit FIFO under-run occurs on the line side. If the race condition were to happen, the 10/100 MAC may end up appending four more bytes of indeterminate data to the Ethernet frame that 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.

Currently no customers 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 the transmission of corrupt frames. This problem is detectable in applications that run 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: No Fix.

17. Expansion Bus Bootup Failure

Problem: EX_DATA is switching between input and output during bootup sequence. As the output buffer is always enabled, the bootup code cannot be driven into the Intel XScale processor successfully.

Implication: After reset, the Intel® IXP43X Product Line of Network Processors performs read transaction through Expansion Bus. The IXP43X product line of network processors continues to drive the EX_DATA[15:0] even when EX_RD_N is asserted. This causes a bus contention with the target device on the Expansion Bus. The IXP43X product line of network processors also drives the EX_DATA[15:0] with values from EX_ADDR[15:0] appearing as if it is attempting to perform a multiplexed mode read transaction. However, the EX_ALE were not asserted, and it does not release the EX_DATA for the target device.

Workaround: None.

Status: Fixed.

18. Incorrect Reset Value of GPIO_INR[0] and GPIO_ISR[0]

Problem: GPIO[0] has a weak pull-up to 3.3V, if the GPIO0 is configured as input. The weak pull-up will appear internally and propagate all the way to the interrupt status register, INTR_ST. It will interrupt only the Intel XScale processor via either IRQ or FIQ (INTR_EN bit 6) that corresponds to the enabled GPIO0.

Implication: The value of the GPIO_INR[15:0] and GPIO_ISR[15:0] should be zero when configured as input; though the value read back from the Intel® IXP43X Product Line of Network Processors does not indicate all zero. The read back value of the GPIO_INR[15:1] and GPIO_ISR[15:1] are zeros, while the GPIO_INR[0] and GPIO_ISR[0] are ones.

Workaround: None.

Status: Fixed.

19. USB Under-run errors when both USB ports perform Interrupt OUT transfers.

Problem: Under-run errors are noticed on USB ports when both the ports are performing Interrupt OUT transfers. Single port will not report any error. The under-run error notifies that the host is ready to transmit, but the data is not available in transmit buffer.

Implication: Interrupt OUT transfer is not supported on both the USB ports simultaneously.

Workaround: Do not perform Interrupt OUT transfer on both ports simultaneously. Use single port instead. The Interrupt OUT pipe is optional. If no Interrupt OUT endpoint is declared

then output reports are transmitted to a device through the Control endpoint.

Status: No Fix.

20. Incorrect Inter-Packet Delay on USB interface.

Problem: The Inter-Packet Delay (IPD) between two consecutive High-Speed transmit packets from the USB ports does not meet the USB2.0 Compliant Test Specification of minimum 88 to maximum 192 bit time. As a result, unable to pass and obtain USB2.0 compliant logo. The IPD is the time between the end of one packet to the start of the subsequent packet.

Table 3. Tested Devices that Pass on the Intel® IXP43X Product Line of Network Processors

Device	Capacity	External Vendor	Note
Storage: USB2.0 Flash Disk	512 MB	Sandisk*	-
Storage: USB2.0 Flash Disk	1 GB	Kingston*	-
Storage: USB2.0 Flash Disk	128 MB	Transcend*	-
Storage: USB1.1 Flash Disk	128 MB	HP*	-
Storage: USB2.0 2.5” HDD	80 GB	Samsung*	USB enclosure: Polar
Storage: USB2.0 3.5” HDD	80 GB	Seagate*	USB enclosure: Polar
Printer: USB2.0 HP Photosmart D5160	-	Hewlett Packard*	-
Printer: USB1.1 Epson Color Photo Stylus 790	-	Epson*	-
Webcam: USB2.0 Creative Webcam Live	-	Creative*	

Implication: Devices that connect to the USB interface may not be recognized when using High Speed mode.

Workaround: None.

Status: No Fix.

21. USB 2.0 Device Reset during Bulk Out Transaction

Problem: In High Speed mode, the bit stuffing mechanism fails to insert a zero on the seventh bit position followed by six consecutive ones. As a result, the USB host fails to send the entire MaxPacketSize, and illegal packets are detected at the device (for example MDATA and Split Token in BULK OUT transfer). The device will be reset and packets are re-transmitted.

Implication: Failure signature is intermittent. The USB device will be reset in between the transmission cycles. Eventually the entire data size (or file) will successfully get to the device. Meaning that the data could be delayed and cause a performance impact.

Workaround: None.

Status: No Fix.

22. RX FIFO Overflow Condition Not Properly Captured in the Rx Status Register

Problem: Whenever Rx FIFO overflows, it generates a rx_fifo_error signal to the Receive State Machine (RSM). The RSM will transit to a state where it will report a status valid for NPE to read the Rx Status register (via ECP_RdRxSts instruction). During this state, the state machine fails to capture the rx_fifo_error signal properly and transits back to idle state. This results in the Rx FIFO overflow condition not properly captured in the Rx Status Register.

Implication: Causes no further packets to be received in the Rx FIFO. The issue is observed only when TMII co-exist with HSS interface. Will not happen when the NPE runs in standard MII mode.

Workaround: Workaround has been implemented in the Intel® IXP400 Software Version 3.0.1. The Ethernet NPE measures the Rx idle time, and issues Ethernet MAC coprocessor instruction (ECP_ClearRxError) to flush MAC Rx Engine FIFO to correct the lock-up



condition. In addition, the Ethernet NPE uses internal control state to prevent repetitive flushing of the MAC Rx Engine FIFO.

Status: No Fix.

23. Incorrect Strapping Configuration during Bootup

Problem: The EX_ADDR[23:0] has a strong pull-up resistor attached internally. During bootup, the strapping value will be read incorrectly when the EX_ADDR[23:0] is attached to 1K ohm pull-down resistor.

Implication: Incorrect strapping configuration will be read during bootup.

Workaround: A 470 ohm pull-down is required to override the internal pull-up resistor.

Status: No Fix.

24. Strong Pull-up Resistor Attached to the JTG_TRST_N

Problem: The JTG_RST_N has a strong pull-up resistor attached internally. During bootup, the IEEE 1149.1 JTAG interface is unable to detect a ‘low’ at the JTG_TRST_N pin when it is attached to a 10K ohm pull-down resistor.

Implication: Intermittent bootup problem.

Workaround: A 100 ohm pull-down is required to override the internal pull-up resistor.

Status: No Fix.

5.0 Intel XScale® Technology Errata Descriptions

1. Abort is Missed When Lock Command is Outstanding

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

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

Status: No Fix.

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

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

The boundary condition where this might occur:

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

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

Status: 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 cycles counted should be only those due to an instruction cache miss or an instruction TLB miss. The following unrelated events in the processor also causes 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 that 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 can also 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: No Fix.

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

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

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

Status: No Fix.

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

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

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

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

Workaround: None.

Status: No Fix.

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

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

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

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

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

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

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

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

mov r0, r0

sub pc, pc, #4

b 1f @ branch to aligned code

.align 5

1:

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

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

mov r0, r0

sub pc, pc, #4
```

Status: No Fix.

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

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

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

Status: 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. The 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. Due to



this bug, the first instruction of such an FIQ handler may be executed twice when it is not a branch instruction.

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

Status: No Fix.

6.0 Specification Changes

1. Updates to Pull-Down Strapping resistor value

Issue: The EX_ADDR[23:0] has a strong pull-up resistor attached internally. A 470 ohm pull-down is required to override these pull-up resistors.

Affected Docs: *Intel® IXP43X Product Line of Network Processors Datasheet*, *Intel® IXP43X Product Line of Network Processors Hardware Design Guidelines* and *Intel® IXDP435 Multi-Service Residential Gateway Reference Platform Schematics*.

2. Updates to Pull-Down Resistor Value on JTG_TRST_N

Issue: The JTG_TRST_N has a strong pull-up resistor attached internally. A 100 ohm pull-down is required to override this pull-up resistor.

Affected Docs: *Intel® IXP43X Product Line of Network Processors Datasheet*, *Intel® IXP43X Product Line of Network Processors Hardware Design Guidelines* and *Intel® IXDP435 Multi-Service Residential Gateway Reference Platform Schematics*.

3. IEEE* 1588 Hardware Assistance Support

Issue: The IEEE* 1588 standard defines a precision clock synchronization protocol for Networked Measurement and Control Systems. It provides a kind of correction mechanism by implementing several message exchange timing information to synchronize individual clocks of the multiple systems. The Intel® IXP43X Product Line of Network Processors provides hardware assist block to achieve this purpose. The hardware assist block is called “Time Synchronization Hardware Assist” (TSYNC). Refer to the *Enabling Time Synchronization (IEEE1588) Hardware on Intel® IXP43X Network Processors Application Notes*.

The logic also supports one non-MII “auxiliary’ interface via GPIO[8:7] pins. The GPIO[8:7] must be tied with a 10K-ohm pull-down while using the IEEE* 1588.

Affected Docs: *Intel® IXP43X Product Line of Network Processors Developer’s Manual* and *Intel® IXP43X Product Line of Network Processors Datasheet*.

4. Turbo-MII Mode Support

Issue: Added new feature. The Intel® IXP43X Product Line of Network Processors support Turbo MII (TMII) mode of operation. It is the standard MII supporting over clock up to 50 Mhz (instead of 25 Mhz) to provide throughput of 200 Mbps full-duplex. For details, see the latest version of the *Intel® IXP43X Product Line of Network Processors Datasheet*.

Affected Docs: *Intel® IXP43X Product Line of Network Processors Datasheet*.

5. Add Procedure to Issue EMRS OCD Command during DDR SDRAM Initialization

Issue: The Intel® IXP43X Product Line does not support OCD calibration. According to the JEDEC Standard for DDR2 SDRAM Specification, when OCD calibration is not used, EMRS OCD Default command (A9=A8=A7=1) followed by EMRS OCD Calibration Mode Exit command (A9=A8=A7=0) must be issued with other operating parameters of EMRS.

Add the following procedures as step 16 in Section 11.2.2.10 DDR SDRAM Initialization of the *Intel® IXP43X Product Line of Network Processors Developer’s Manual*:

EMRS OCD Default command (A9=A8=A7=1) followed by EMRS OCD Calibration Mode Exit command (A9=A8=A7=0) must be issued. The ‘OCD Program’ field is mapped to DCALADDR register bit[9:7].

1) Configure the DCALCSR register (Hex offset: CC00 F500) with bit[2:0]=’011’

2) Configure the DCALADDR register (Hex offset: CC00 F504) with bit[1:0]=’01’, and bit[9:7]=’111’ to issue a OCD Default command



3) Software must wait for at least 2 cycles. Then configure the DCALADDR register (Hex offset: CC00 F504) with bit[1:0]='01', and bit[9:7]='111'. (To issue a OCD Exit command)

Note: Other fields in the DCALCSR and DCALADDR registers should be left as 'zero' (default value after reset). The 'Qoff', 'RDQS', and 'DQS' fields are mapped to DCALADDR register bit[12:10] respectively.

The DLL enabled is configured through writing '0100' to the SDIR register. The Rtt is configured by writing to SDCR0 register bit[5:4].

Procedure to modify the bootloader (for EMRS OCD instruction):

Solution for Redboot* platform *(packages/hal/arm/xscale/ixdp435/current/include/hal_ixp425.h)* as illustrated in Figure 2.

Figure 2. Define DDR_DCALCSR and DDR_DCALADDR Registers

```
#define IXP_DDR_SDPR4        REG32(IXP_DDR_CFG_BASE1,0x60)
#define IXP_DDR_SDPR5        REG32(IXP_DDR_CFG_BASE1,0x64)
#define IXP_DDR_SDPR6        REG32(IXP_DDR_CFG_BASE1,0x68)
#define IXP_DDR_SDPR7        REG32(IXP_DDR_CFG_BASE1,0x6C)

#define IXP_DDR_DCALCSR      REG32(IXP_DDR_CFG_BASE2,0x00)
#define IXP_DDR_DCALADDR     REG32(IXP_DDR_CFG_BASE2,0x04)
#define IXP_DDR_RCVDLY       REG32(IXP_DDR_CFG_BASE2,0x50)
#define IXP_DDR_SLVLMIX0     REG32(IXP_DDR_CFG_BASE2,0x54)
```

Solution for Redboot* platform *(packages/hal/arm/xscale/ixdp435/current/include/hal_platform_setup.h)*. Add in the EMRS code (bold) in between Step 14 and Step 15, as illustrated in Figure 3.

Figure 3. EMRS Code Example

```
// Step 14. Issue mode register set w/o DLL reset
        mov     r1, #DDR_SDIR_MODE_SET_NO_RESET
        str     r1, [r0, #IXP_DDR_SDIR]
        DELAY   0x100000, r1
//*********Added steps:EMRS OCD Calibration Default********
        ldr     r2, [r0, #IXP_DDR_DCALADDR] //save DCALADDR value in r2 before modifying
        ldr     r1, = 0x01000003    //set bit24=1 for SDRAM Enable
                                    //set bits[0:1] for EMRS OCD Default
        str     r1, [r0, #IXP_DDR_DCALCSR]
        ldr     r1, =0x00000381     //set bits [9:7] and bit 0 for EMRS OCD Default
        str     r1, [r0, #IXP_DDR_DCALADDR]
        DELAY   0x100000, r1
//*********Added steps:EMRS OCD Calibration Mode Exit******
        ldr     r1, =0x00000001     //set bit 0 only for exit
        str     r1, [r0, #IXP_DDR_DCALADDR]
        DELAY   0x100000, r1
        str     r2, [r0, #IXP_DDR_DCALADDR] //restore original DCALADDR value

// Step 15. Start normal operation
```

Affected Docs: *Intel® IXP43X Product Line of Network Processors Developer's Manual*.

Guidelines, the TVA4 timing specification shows a minimum time of 2020 ps. The minimum timing for TVA4 should be 1650 ps. The tables will be replaced as below.

Table: DDRII-400MHz Interface - Signal Timings

Symbol	Parameter	Min.	Nominal	Max.	Units	Notes
T_{VB4}	DQ, CB, and DM read input valid time before DQS rising or falling edges	-480 ps			ps	2
T_{VA4}	DQ, CB, and DM read input valid time after DQS rising or falling edges	1650 ps			ps	2

Affected Docs: *Intel® IXP43X Product Line of Network Processors Datasheet* and *Intel® IXP43X Product Line of Network Processors Hardware Design Guidelines*.