Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor

Specification Update

October 2004

Notice: The Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor may contain design defects or errors known as errata that may cause the product to deviate from published specifications. Current characterized errata are documented in this specification update.

Document Number: 252702-006

Contents

This page is intentionally left blank.

Revision History

Date	Version	Description
October 2004	006	Added Non-Core Errata 25 through 35, Core Errata 1 through 8, Specification Clarifications 3 through 6, and Documentation Changes 1 through 6. Also modified Non-Core Errata 13. Updated the Specification Changes section.
July 2004	005	Updated Intel® product branding. Change bars were retained from the previous release of this document (004).
April 2004	004	Added Errata 21 through 24, updated Specification Change 1. Removed Documentation Changes section. Minor edits to Table 1 and Table 2.
November 2003	003	Added Errata 15 through 20 and Documentation Changes 17 through 35.
August 2003	002	Added Errata 11 through 14, Specification Changes 1 and 2, Specification Clarification 2, and Documentation Changes 1 through 17. Updated product-name references. Incorporated the Intel® IXC1100 Control Plane Processor. This version of the document supersedes the *Intel® IXC1100 Control Plane Processor Specification Update* (251806-001).
April 2003	001	Published Errata 1 through 10, Specification Changes 1 through 2, Specification Clarifications 1, and Documentation Changes 1 through 20. This first-version document contains errata previously published in the *Intel® IXP425 Network Processor Based on Intel® XScale™ Microarchitecture Specification Update* (251805) and replaces that other document.

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 errata and specification clarifications and changes. It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools. This document may contain information that was not previously published.

Affected Documents/Related Documents

This document contains errata previously published in the *Intel® IXP425 Network Processor Based on Intel® XScale™ Microarchitecture Specification Update* (251805) and the *Intel® IXC1100 Control Plane Processor Specification Update* (251806). This document supersedes and replaces both documents.

Similarly, the *Intel® IXP425 Network Processor Based on Intel® XScale™ Microarchitecture Datasheet* and the *Intel® IXP425 Network Processor Based on Intel® XScale™ Microarchitecture Component Specification* will no longer be updated. Both documents have been replaced, respectively, by the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* and the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual*.

Title	Document Number
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

Nomenclature

Errata are design defects or errors. These may cause the Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor 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 currently 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, error, or omissions from the current published specifications. These changes will incorporated in the next release of the respective document.

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

Summary Table of Changes

The following table indicates the errata, specification changes, or specification clarifications that apply to the Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor. Intel may fix some of the errata in a future stepping of the component and account for the other outstanding issues through documentation or specification changes as noted. This table uses the following notations:

Codes Used in Summary Table

Stepping

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

Page

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

Status

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

Row

A 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

No.	Steppings IXP425, IXC1100			Steppings IXP420, IXP421, IXP422		Page	Status	Errata
	A0	B0		B0				
1	X					17	Fixed	PCI Doorbell Register Does Not Work Properly (SCR 460).
2	X					17	Fixed	UTOPIA Interface Status Collection Synchronization Issues (SCR 485)
3	X					17	Fixed	No Byte Enables Asserted During a PCI I/O Read of the Intel® IXP425 Network Processor Causes Unpredictable Behavior (SCR 469)
4	X					17	Fixed	Simultaneous AHB Access of the PCI Bus Controller (SCR 499)
5	X					18	Fixed	66-MHz PCI Operation (SCR 543)
6	X	X		X		18	No Fix	Ethernet Control Protocol Frames Transmit Defer Status Bit Error
7	X					18	Fixed	Logic 0 is Driven on Both the USB D+ and D- at Reset (SCR 545)
8	X					18	Fixed	Cannot Generate Watchdog Timer Reset (SCR 641)
9	X					19	Fixed	PCI Non-Prefetch Reads (SCR 1254)
10	X	X		X		19	No Fix	Timer Status Interrupts Get Lost During MMR Writes (SCR 1653)
11	X	X		X		20	No Fix	Character Time-Out Interrupt Sticks Under Certain Software Timing Conditions (SCR 2235)
12	X					21	Fixed	The Intel® IXP425 A-0 Step Processor May Have Problems Working With Some SDRAM Devices (SCR 2411)
13	X	X		X		22	No Fix	PCI DMA Lock-Up Condition (SCR 2372)
14	X	X		X		23	No Fix	PCI Doorbell Register Lock-up Condition When Using Two Products Together That Have Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor (SCR 2379)
15	X	X		X		23	No Fix	PCI Controller Returns Infinite Retries on PCI After AHB Pre-Fetch Error (SCR 1289)
16	X	X		X		23	No Fix	UART Break Indicator (SCR 2929)
17	X	X		X		24	No Fix	Intel XScale® Core Non-Branch Instruction in Vector Table (SCR 2871)
18	X	X		X		24	No Fix	IRQ3 Is Locking the System ‘Disable’ (SCR 2143)
19	X	X		X		24	No Fix	EX_IOWAIT_N Timing (SCR 3051)
20	X					25	Fixed	SOF During Control Read Can Corrupt USB Transfer (SCR 1553)
21	X	X		X		25	No Fix	USB - Invalid 0x81 Value in Endpoint 0 Control Register on SETUP Transfer (SCR 3077)
22	X	X		X		25	No Fix	Ethernet Coprocessors - Ethernet Pad Enable Overrides Append FCS (SCR 299)

Non-Core Errata (Continued)

No.	Steppings IXP425, IXC1100			Steppings IXP420, IXP421, IXP422		Page	Status	Errata
	A0	B0		B0				
23	X	X		X		26	No Fix	Ethernet Coprocessors - Length Errors on Received Frames (SCR 711)
24	X	X		X		26	No Fix	PCI DC Parameter VIH Marginality Issue (SCR 3121)
25	X	X		X		26	No Fix	False PCI DMA Completion Notification Causing Data Corruption (SCR 3910)
26	X	X		X		27	No Fix	Expansion Bus HPI Interface Potential for Contention on Reads with T4=0 (SCR 4117)
27	X	X		X		27	No Fix	PCI Hangs With a Multiple Inbound Error Condition (SCR 4160)
28	X	X		X		27	No Fix	PCI RCOMP Operation if PCI Clock Stops (SCR 4022)
29	X	X		X		27	No Fix	UART - Break Condition Asserted Too Early if Two Stop bits are Used (SCR 4092)
30	X	X		X		28	No Fix	Ethernet Coprocessors - Address Filtering Logic Ignores the Second to Last Nibble of the Destination Address (SCR 4185)
31	X	X		X		28	No Fix	USB DC Parameter VIH Specification Change (SCR 3111)
32	X	X		X		28	No Fix	Start of DMA Operation Close to Start of Non-Prefetch (NP) Operation can Hang NP Operation on AHB (SCR 3939)
33	X	X		X		29	No Fix	PCI Accesses to the Queue Manager During Queue and SRAM Mode (SCR 4076)
34	X	X		X		29	No Fix	Ethernet MACs Detect Late Collision Earlier Than Ethernet 802.3 Specifications (SCR 4062)
35	X	X		X		29	No Fix	Read of PCI Controllers BAR 32'hXXFF_FFFC Rd[N] Corrupts Subsequent Rd[N+1] (SCR 3850)

Core Errata

No.	Steppings IXP425, IXC1100			Steppings IXP420, IXP421, IXP422		Page	Status	Specification Changes
	A0	B0		B0				
1	X	X		X		30	No Fix	Abort is Missed When Lock Command is Outstanding
2	X	X		X		30	No Fix	Aborted Store That Hits the Data Cache May Mark Write-Back Data as ‘Dirty’
3	X	X		X		30	No Fix	Performance Monitor Unit Event 0x1 Can Be Incremented Erroneously by Unrelated Events
4	X	X		X		31	No Fix	In Special Debug State, Back-to-Back Memory Operations — Where the First Instruction Aborts — May Cause a Hang

Core Errata (Continued)

No.	Steppings IXP425, IXC1100			Steppings IXP420, IXP421, IXP422		Page	Status	Specification Changes
	A0	B0		B0				
5	X	X		x		31	No Fix	Accesses to the CP15 ID Register with Opcode2 > 0b001 Returns Unpredictable Values
6	X	X		X		32	No Fix	Disabling and Re-Enabling the MMU can Hang the Core or Cause it to Execute the Wrong Code
7	X	X		X		32	No Fix	Updating the JTAG Parallel Registers Requires an Extra TCK Rising Edge
8	X	X		X		32	No Fix	Non-Branch Instruction in Vector Table May Execute Twice After a Thumb Mode Exception

Specification Changes

No.	Steppings IXP425, IXC1100			Steppings IXP420, IXP421, IXP422		Page	Status	Specification Changes
	A0	B0		B0				
1		X		X		34	Doc	Update to Expansion Bus Setup / Hold Timing Values

Specification Clarifications

No.	Steppings IXP425, IXC1100			Steppings IXP420, IXP421, IXP422		Page	Status	Specification Clarifications
	A0	B0		B0				
1	X	X		X		35	Doc	PCI Byte Enables All Asserted During All Memory Cycle Reads (SCR 1897)
2	X	X		X		35	Doc	Pull-Up Resistor Required on ETH_MDIO Pin (SCR 1225)
3	X	X		X		35	Doc	Ethernet MAC Broadcast Disable bit Needs Further Documentation (SCR 4166))
4	X	X		X		36	Doc	Replace “ICC and Total Average Power” with Commercial Temperature and Extended Temperature Devices Tables (SCR 4009)
5	X	X		X		37	Doc	Update of AC Timings for Expansion Bus (SCR 4199)
6	X	X		X		37	Doc	Update of Expansion Bus Timing Diagrams (SCR 4199)

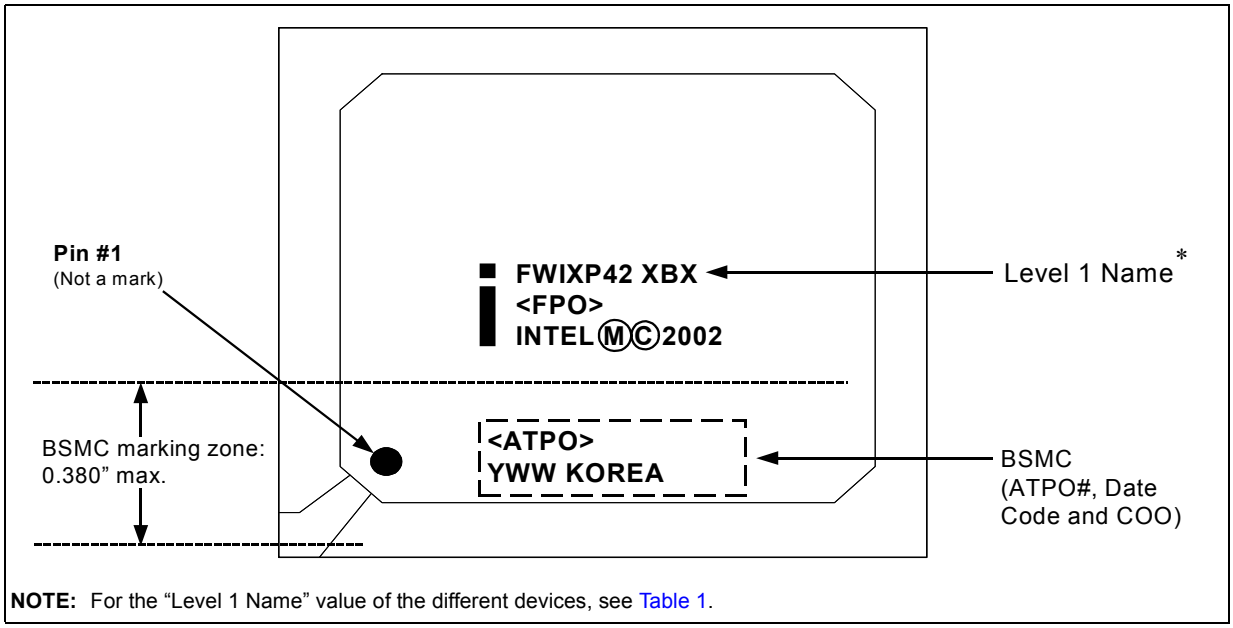
Documentation Changes

No.	Steppings IXP425, IXC1100			Steppings IXP420, IXP421, IXP422		Page	Status	Documentation Changes
	A0	B0		B0				
1	X	X		X		51	Doc	Correction to Expansion Bus Label (SCR 3888)
2	X	X		X		51	Doc	Correction to Input Reference Slew Rate in a Oscillator Configuration (SCR 3890)
3	X	X		X		51	Doc	Update of Management Data Output Register (SCR 4053)
4	X	X		X		51	Doc	Update of Core Clock Speed Expansion Bus Configuration Table (SCR 4086)
5	X	X		X		52	Doc	Correction of PCI Controller Control and Status Register Description (SCR 4206)
6	X	X		X		52	Doc	Correction to Expansion Bus Timing Diagrams

Identification Information

Markings

Figure 1. Package Markings



NOTE: For the “Level 1 Name” value of the different devices, see Table 1.

Table 1. Part Numbers

Device	Stepping	Speed (MHz)	Part #
Intel® IXP425	B-0	533	FWIXP425BD
Intel® IXP425	B-0	400	FWIXP425BC
Intel® IXP425	B-0	266	FWIXP425BB
Intel® IXP425	B-0	533 Extended Temperature	GWIXP425BDT
Intel® IXP425	B-0	400 Extended Temperature	GWIXP425BCT
Intel® IXP425	B-0	266 Extended Temperature	GWIXP425BBT
Intel® IXP422	B-0	266	FWIXP422BB
Intel® IXP421	B-0	266	FWIXP421BB
Intel® IXP420	B-0	533	FWIXP420BD
Intel® IXP420	B-0	400	FWIXP420BC
Intel® IXP420	B-0	266	FWIXP420BB
Intel® IXP420	B-0	266 Extended Temperature	GWIXP420BBT
Intel® IXC1100	B-0	533	FWIXC1100BD
Intel® IXC1100	B-0	400	FWIXC1100BC
Intel® IXC1100	B-0	266	FWIXC1100BB
Intel® IXC1100	B-0	533 Extended Temperature	GWIXC1100BDT
Intel® IXC1100	B-0	400 Extended Temperature	GWIXC1100BCT
Intel® IXC1100	B-0	266 Extended Temperature	GWIXC1100BBT

Change Summary: Intel® IXP425 A0 to B0 Step

Some of the known errata on the Intel® IXP425 Network Processor A0 stepping have been fixed in the IXP425 network processor B0-step processor. For the still-unresolved errata, see "Non-Core Errata" on page 9.

The internal device ID — specific to the IXP425 network processor — has been updated to reflect the B0-step device. For specific information about the device ID, see the *IIntel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual*, Section 3.5.1.1.

The IXP425 B0 stepping and the Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor have redefined pin D25 from "Reserved" to RCOMP. This pin is used to adjust the drive strength used for the PCI interface. For specific information regarding the RCOMP pin, see the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet*.

The B0 stepping and the Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor support 66-MHz PCI operation.

The B0-stepping device has gone through a top-side mark change to reflect the B0 stepping. For specific information on the package top-side mark, see "Identification Information" on page 13.

The A0 stepping supported memory configurations of 32 Mbyte to 256 Mbyte. The B0-stepping device and the Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor support memory configurations from 8 Mbyte to 256 Mbyte. For further information on the SDRAM memory interface, see the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual*.

The IXP425 B0-stepping and IXP422 processors have included capability for Advanced Encryption Standard (AES). For a summary of the IXP42X product line and IXC1100 control plane processors features, see "Processor Features" on page 16

The B0-stepping and the IXP42X product line and IXC1100 control plane processors include support for data or address coherency operation through the use of the Page Attribute bit.

The B0-stepping and IXP42X product line and IXC1100 control plane processors have added a BYTE_SWAP_EN bit located in Expansion Bus Configuration Register 1. The bit location is bit 8 and is defined as BYTE_SWAP_EN. This bit sets byte swapping at the Core Gasket.

The B0 stepping and the IXP42X product line and IXC1100 control plane processors include enhanced support for little-endian mode of operation.

Table 2. Processor Features

Feature	Intel® IXP425 Network Processor B0 Step	Intel® IXP422 Network Processor	Intel® IXP421 Network Processor	Intel® IXP420 Network Processor	Intel® IXC1100 Control Plane Processor
Processor Speed (MHz)	266/400/533	266	266	266/400/533	266/400/533
UTOPIA 2	X		X		
GPIO	X	X	X	X	X
UART 0/1	X	X	X	X	X
HSS 0	X		X		
HSS 1	X		X		
MII 0	X	X	X	X	X
MII 1	X	X		X	X
USB	X	X	X	X	X
PCI	X	X	X	X	X
Expansion Bus	16-bit, 66-MHz	16-bit, 66-MHz	16-bit, 66-MHz	16-bit, 66-MHz	16-bit, 66-MHz
SDRAM	32-bit, 133-MHz	32-bit, 133-MHz	32-bit, 133-MHz	32-bit, 133-MHz	32-bit, 133-MHz
AES / DES / DES3	X	X			
Multi-Channel HDLC	8		8		
SHA-1 / MD-5	X	X			
Commercial Temperature	X	X	X	X	X
Extended Temperature	X			X[1]	X

1. Only the 266-MHz version of the Intel® IXP420 Network Processor supports extended temperature.

Non-Core Errata

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: 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: 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: 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 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: 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: 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 IXP42X product line and IXC1100 control plane processors is unusable.

Workaround: None.

Status: 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: 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: 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: 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 and IXC1100 control plane processors 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 work arounds:

- 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 IXP42X product line 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: No Fix.

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 workaround's 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: No Fix.

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

Problem: Although the Intel® IXP42X product line and IXC1100 control plane processors 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 and IXC1100 control plane processors sets 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 IXP425 A-0 step processor does not work with some SDRAM memory.

Workaround: Use a memory device listed in the preceding, “Memory known to work” section.

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® IXP42X product line and IXC1100 control plane processors.

Status: 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 IXP42X product line and IXC1100 control plane processors. Lock-up may occur when the IXP42X’s DMA controller 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’s inbound transfer queue. However they are not de-queued from this inbound queue for execution in the IXP42X 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, the IXP42X will begin to retry this fourth inbound PCI transfer, as the IXP42X’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 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 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. In this case, the IXP42X product line does a DMA 8-word PCI read, which gets retried by the bridge as the bridge fetches that 8 words of data. The bridge, asynchronously, issues a series of posted PCI writes in quick succession to the IXP42X product line 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 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’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: No Fix.

14. PCI Doorbell Register Lock-up Condition When Using Two Products Together That Have Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor (SCR 2379)

Problem: It is possible that the PCI bus can get in a locked condition when multiple products — using IXP42X product line and IXC1100 control plane processors — are connected in a system and these systems are using the PCI doorbell registers of the IXP42X product line and IXC1100 control plane processors. This lockup only occurs when both of the IXP42X product line and IXC1100 control plane 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 IXP42X product line and IXC1100 control plane 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: 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 IXP42X product line and IXC1100 control plane processors, 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 IXP42X product line and IXC1100 control plane processors 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 IXP42X product line and IXC1100 control plane processors. The initial read completes normally on PCI.

Implication: The IXP42X product line and IXC1100 control plane processors do not support the alias of above 256 Mbyte of SDRAM memory space.

Workaround: None.

Status: No Fix.

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: Under investigation.

Status: No Fix.

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: No Fix.

18. IRQ3 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: 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: 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: 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, then 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: No Fix.

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

Problem: The IXP42X product line and IXC1100 control plane processors have an Ethernet Coprocessor that is configured by Intel XScale core software. [Note that some IXP42X product line and IXC1100 control plane 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: No Fix.

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

Problem: The IXP42X product line and IXC1100 control plane processors have an Ethernet Coprocessor that is configured by Intel XScale core software. [Note that some IXP42X product line and IXC1100 control plane 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: 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 VCCP. This specification is changed to 0.6 VCCP.

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: No Fix.

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:

1. 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.
2. 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.

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 max of 10 ns after the deassertion of DS (which is ex_wr_n). The Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor 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 10ns 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: 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 IXP42X product line and IXC1100 control plane processors.

Status: 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: The IXP42X product line does 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 product line must go into reset first, and then change the PCI CLKIN by pulling up or pulling down the EX_ADDR[4] pin.

Status: 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: No Fix.

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

Problem: The IXP42X product line has an Ethernet coprocessor configured by Intel XScale core software. [Note that some IXP42X product line have two Ethernet coprocessors.] The Ethernet coprocessor logic ignores the second last nibble of the destination address regardless of the packet type (unicast, multicast, broadcast), that is, Destination Address: 11 22 33 44 55 x6. The reason it is the second last is that the address is transmitted on the line with the high nibble first and then the low nibble.

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

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

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

31. USB DC Parameter VIH Specification Change (SCR 3111)

Problem: The input-high voltage (VIH) for the USB signals does not meet the documented specification. In the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet*, Table 26, the USB DC Parameters specifies the VIH minimum value as 2.0 V. This specification is changed to 2.15 V.

Implication: The USB VIH DC parameter does not meet industry-standard specifications.

Workaround: None

Status: No Fix.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (252479-002)

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: No Fix.

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 product line memory space. An external PCI master must use PCI BAR5 when accessing the Queue Manager when in Queue mode.

Status: 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: No Fix.

35. Read of PCI Controllers BAR 32'hXXFF_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. Perhaps change the setup 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.

Core Errata

1. Abort is Missed When Lock Command is Outstanding

Problem: A bus abort occurs on a code fetch while an instruction TLB or I-Cache lock *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: 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 4-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: 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: 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: 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: 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: No Fix.

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

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

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

Status: No Fix

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: No Fix.

Specification Changes

1. Update to Expansion Bus Setup / Hold Timing Values

Issue: In Table 58, “Setup/Hold Timing Values”, of the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet*, the Output Valid timing specification shows a minimum time of 15 ns. The minimum time for Output Valid timing specification should be blank. Instead, the maximum specification column time for Output Valid should be 15 ns.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet*, (252479-004)

Specification Clarifications

1. PCI Byte Enables All Asserted During All Memory Cycle Reads (SCR 1897)

Issue: The PCI controller drives all byte enables low (asserted) during a memory cycle read of non-prefetch memory. However, I/O reads and memory-cycle writes do drive individual byte enables.

Implication: If an external PCI device has non-prefetch memory and requires either a 16-bit or 8-bit read, there is a possibility that the device will not respond correctly to the IXP42X product line and IXC1100 control plane processors' memory reads. This is because the IXP42X product line and IXC1100 control plane processors always perform a 32-bit read to the non-prefetch memory region specified in register PCI_NP_AD.

The 8-bit or 16-bit external device should respond with a "target abort," as per the PCI 2.2 specification, if a 32-bit read is performed to its non-prefetch memory and it requires a 16-bit or 8-bit read.

The IXP42X product line and IXC1100 control plane processors will drive all the byte enables asserted during all memory cycle reads of the external PCI device, no matter what the PCI_NP_CBE register contains in the byte enable bits.

To read non-prefetch memory sub-DWORDS (8-bit or 16-bit), use I/O reads. If it is necessary to use memory cycle reads of sub-DWORDS, a hardware work around may be required. Contact your Intel field application engineer if you require a hardware work around.

Affected Docs: *Intel® IXP425 Network Processor Based on Intel® XScale™ Microarchitecture Component Specification* (11725, v. 1.0) and the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* (252480)

2. Pull-Up Resistor Required on ETH_MDIO Pin (SCR 1225)

Issue: An external pull-up resistor of 1.5K ohm is required on ETH_MDIO to properly quantify the external PHYs used in the system. For specific implementation specifics, see the IEEE 802.3 specification.

Implication: Incorrect number of PHYs will be identified. For example, when interfacing a single Intel® LXT972 Fast Ethernet Transceiver to one of the IXP42X product line and IXC1100 control plane processors — without the external resistor — the processor will see 32 PHYs on the MII interface.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (252479-002)

3. Ethernet MAC Broadcast Disable bit Needs Further Documentation (SCR 4166)

Issue: Section 15.1.5 of the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* states "Determining if the 48-bit destination address of the received frame contains all logic 1s, filters the broadcast frames. Setting bit 7 of Receive Control Register 1 (RXCTRL1) to logic 1 prevents received broadcast frames from being sent to the NPE."

Section 15.2.3 further documents Bit 7, Broadcast Disable. 1 = Prevents broadcast packets from being passed to the application logic.

Clarification: Broadcast packets will only be dropped if broadcast disable is set to "1", the address mask register is NOT 00 00 00 00 00 00 and the address filter is enabled. Setting the address mask register to all 00 (don't care about the address) and setting the Broadcast disable bit to "1" (checking the address) at the same time is a contradiction and will result in broadcast packets still being received and the packet status (MCST_PKT / BCST_PKT) will be read back accordingly.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* (252480-004)

4. Replace "ICC and Total Average Power" with Commercial Temperature and Extended Temperature Devices Tables (SCR 4009)

Issue: The *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* only lists the extended temperature power and current specs (see datasheet Table 66, "I_{CC} and Total Average Power"). The commercial-temperature range power and current consumption specifications are missing. Replace datasheet Table 66, "I_{CC} and Total Average Power", with the following two new tables:

I_{CC} and Total Average Power – Commercial Temperature Range

Speed	Symbol	Description	Typical Current and Power(1)	Max Current (2)	Average Max Power (2)
266 MHz	I_{cc}	Core supply current	0.70A	0.725A	1.0W
	I_{ccp}	I/O supply current	0.17A	0.26A	0.9W
	P_{TOTAL}	Total average power both supplies	1.5W		1.9W
400 MHz	I_{cc}	Core supply current	0.75A	0.800A	1.09W
	I_{ccp}	I/O supply current	0.17A	.26A	0.9W
	P_{TOTAL}	Total average power both supplies	1.57W		2.0W
533 MHz	I_{cc}	Core supply current	0.82A	1.00A	1.4W
	I_{ccp}	I/O supply current	0.17A	0.26A	0.9W
	P_{TOTAL}	Total average power both supplies	1.66W		2.3W

NOTES:
1. Typical current ICC and ICCP are not tested. Typical currents were measured on the Intel® IXDP425 / IXCDP1100 Development Platform at room temperature using typical SKU silicon samples. A SmartBits* tester was used in a router application running Linux on the development board. Two Ethernet NPEs, and two Ethernet controller PCI cards were used in this router application. Typical case power supply voltages VCC =1.327V, VCCP = 3.363 V. Typical operating temperature is room temperature.
2. Maximum voltages: VCC = 1.365 V, VCCP = 3.465 V, VCCosc= 1.365 V, VCCPLL1= 1.365 V, VCCPLL2= 1.365 V, maximum capacitive loading on all I/O pins of 50 pF. Maximum ICC and ICCP are steady state currents. Maximum operating temperature.

Table 3. I_{CC} and Total Average Power – Extended Temperature Range

Speed	Symbol	Description	Typical Current and Power (1)	Max Current (2)	Average Max Power (2)
266 MHz	I_{cc}	Core supply current	0.70A	0.95A	1.3W
	I_{ccp}	I/O supply current	0.17A	0.26A	0.9W
	P_{TOTAL}	Total average power both supplies	1.5W		2.2W
400 MHz	I_{cc}	Core supply current	0.75A	1.05A	1.43W
	I_{ccp}	I/O supply current	0.17A	.26A	0.9W
	P_{TOTAL}	Total average power both supplies	1.57W		2.33W
533 MHz	I_{cc}	Core supply current	0.82A	1.15A	1.57W
	I_{ccp}	I/O supply current	0.17A	0.26A	0.9W
	P_{TOTAL}	Total average power both supplies	1.66W		2.47W

NOTES:
1. Typical current ICC and ICCP are not tested. Typical currents were measured on the Intel® IXDP425 / IXCDP1100 Development Platform at room temperature using typical SKU silicon samples. A SmartBits* tester was used in a router application running Linux on the development board. Two Ethernet NPEs, and two Ethernet controller PCI cards were used in this router application. Typical case power supply voltages VCC =1.327V, VCCP = 3.363 V. Typical operating temperature is room temperature.
2. Maximum voltages: VCC = 1.365 V, VCCP = 3.465 V, VCCosc= 1.365 V, VCCPLL1= 1.365 V, VCCPLL2= 1.365 V, maximum capacitive loading on all I/O pins of 50 pF. Maximum ICC and ICCP are steady state currents.Maximum operating temperature.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (252479-002)

5. Update of AC Timings for Expansion Bus (SCR 4199)

Issue: In the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet*, several AC timings for the expansion bus need to be corrected for both Intel and Motorola modes. The following tables are affected: Table 52, Table 53, Table 54, Table 55.

For the affected tables, Trdsetup was 5.3 ns min. and 14.7 ns max. Change this to: Trdsetup 15 ns min. in all tables (and nothing for max).

Also, Trdhold was 2 ns min. Change this to: Trdhold 0 ns min. in all tables

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (252479-003)

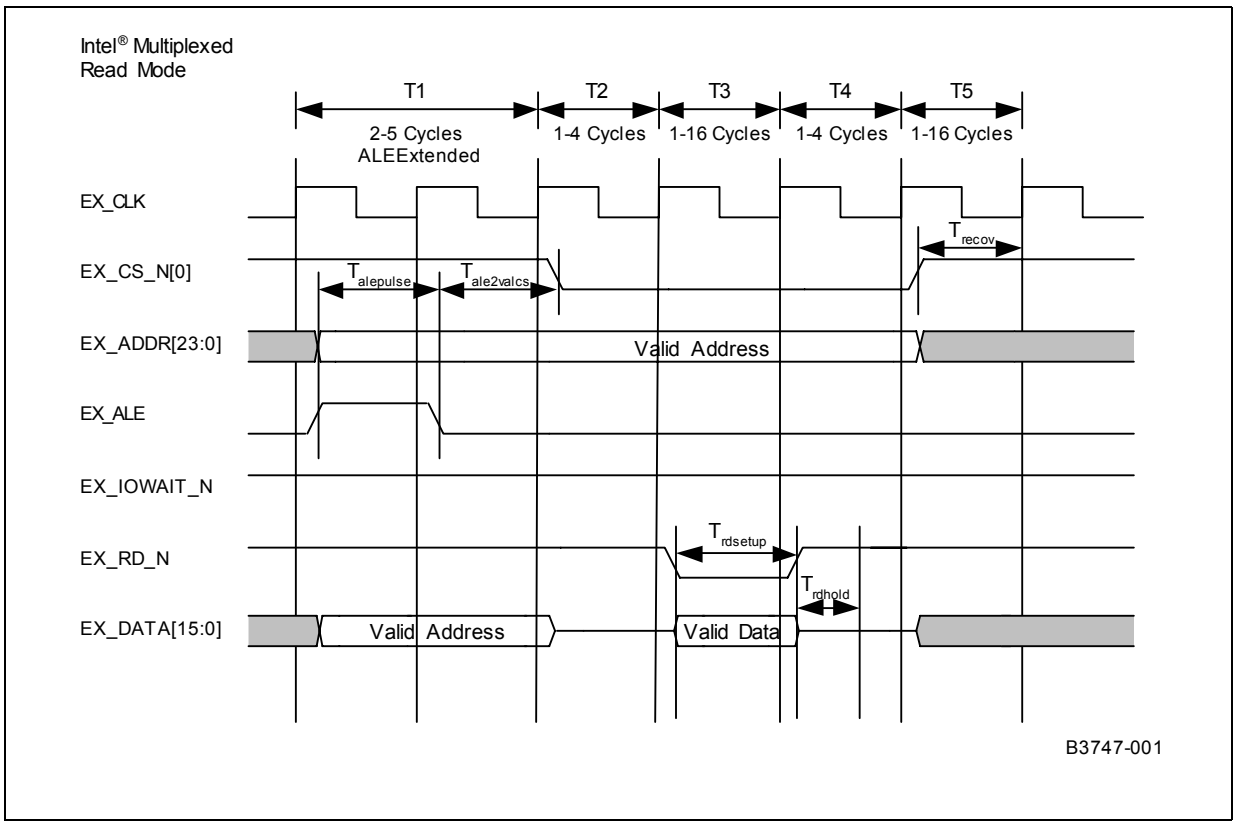
6. Update of Expansion Bus Timing Diagrams (SCR 4199)

Issue: In the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet*, several Expansion Bus timing diagrams are incorrect. Figures 25–33 are affected.

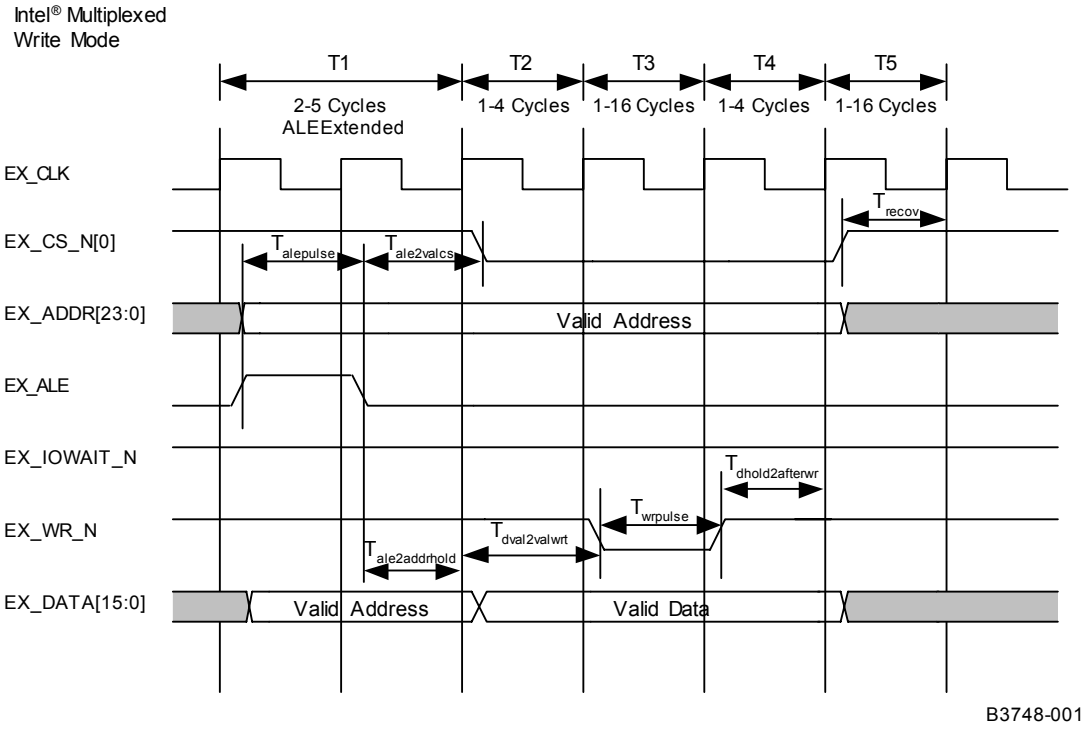
Replace the affected datasheet figures as follows.

Replace datasheet Figure 25, “Intel Multiplexed Mode”, with the following two figures, “Intel® Multiplexed Read Mode”, and “Intel® Multiplexed Write Mode”.

Intel® Multiplexed Read Mode

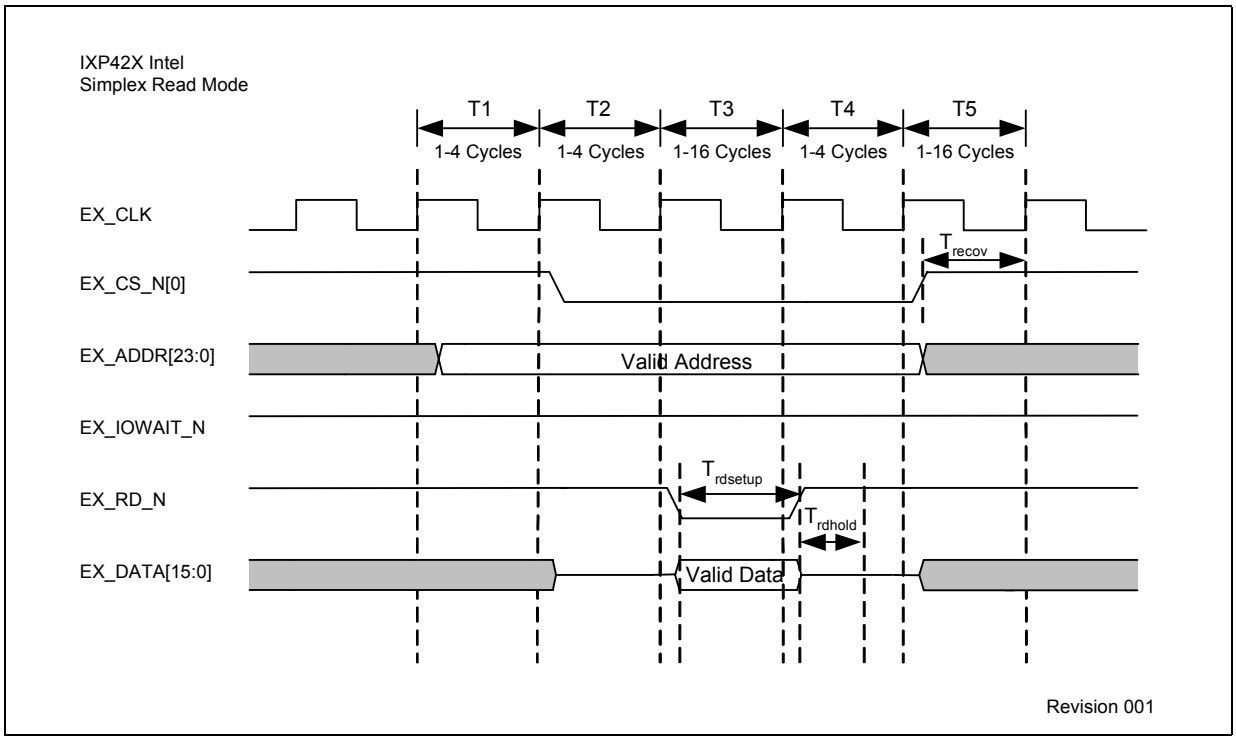


Intel® Multiplexed Write Mode

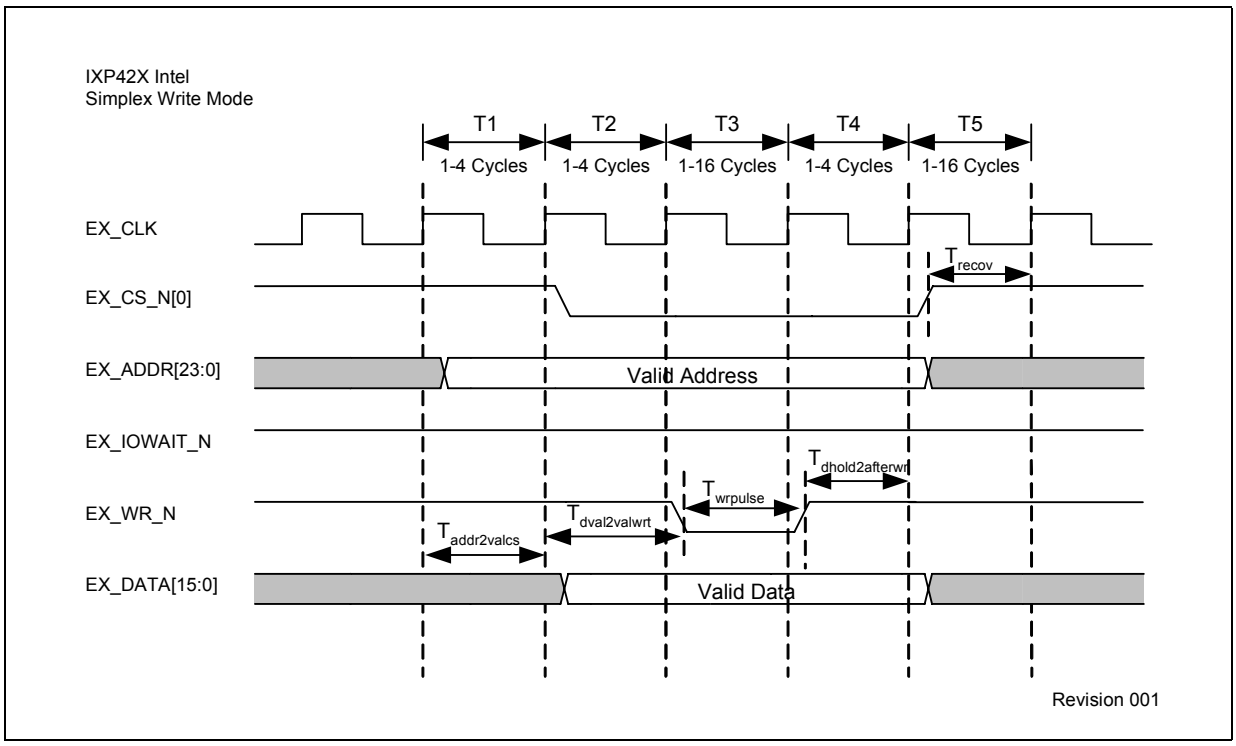


Replace datasheet Figure 26, “Intel® Simplex Mode”, with the following two figures, “Intel® Simplex Read Mode”, and “Intel® Simplex Write Mode”.

Intel® Simplex Read Mode

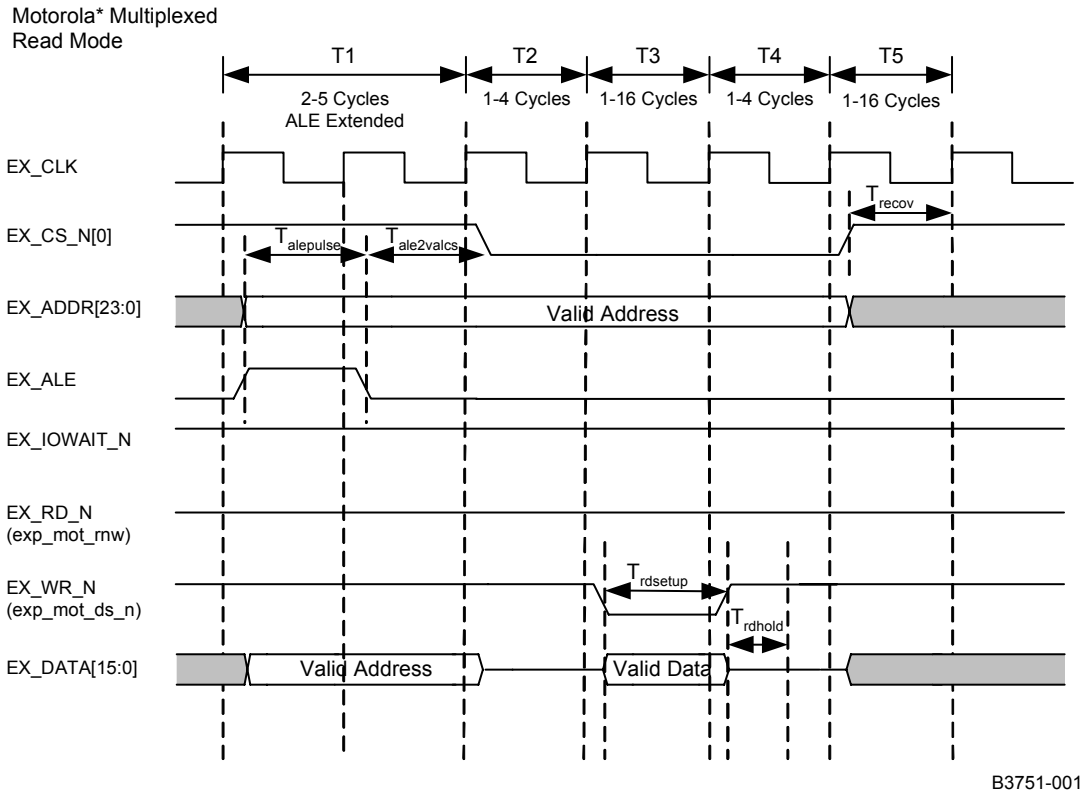


Intel® Simplex Write Mode

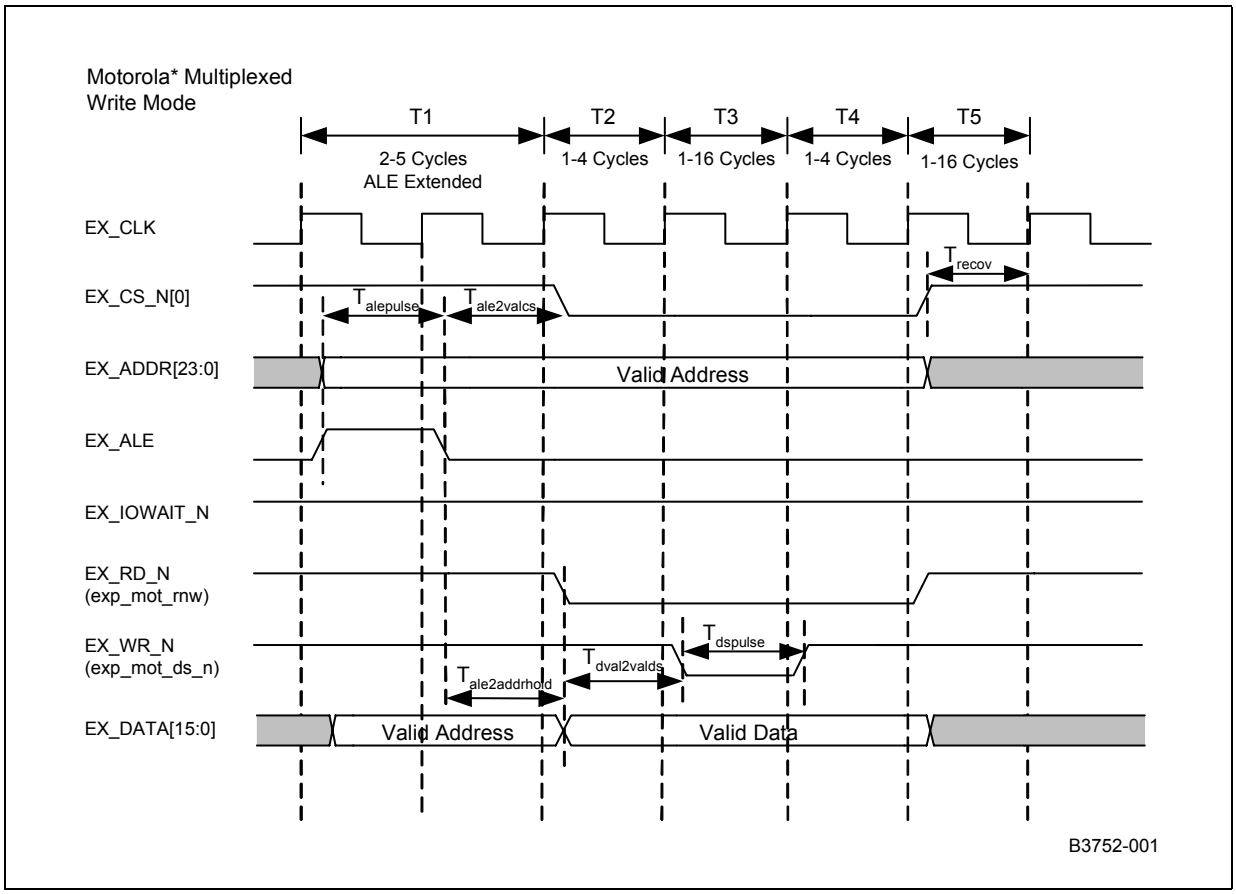


Replace datasheet Figure 27, “Motorola* Multiplexed Mode” with the following two figures, “Motorola* Multiplexed Read Mode”, and “Motorola* Multiplexed Write Mode”.

Motorola* Multiplexed Read Mode

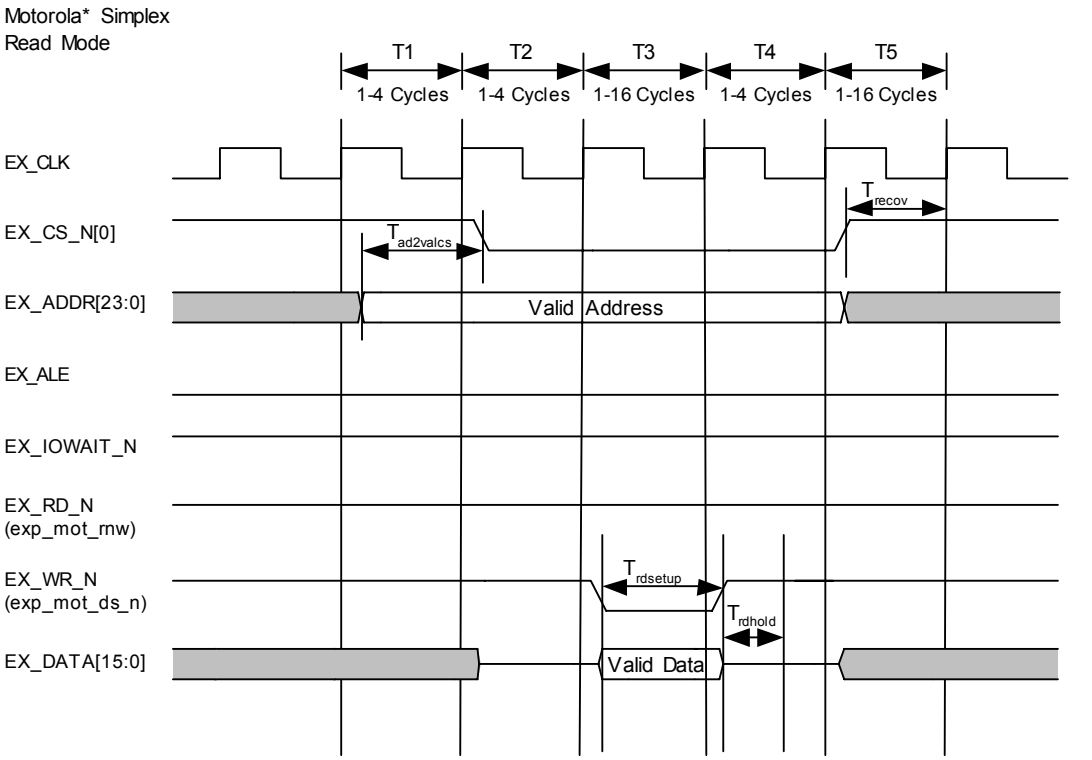


Motorola* Multiplexed Write Mode

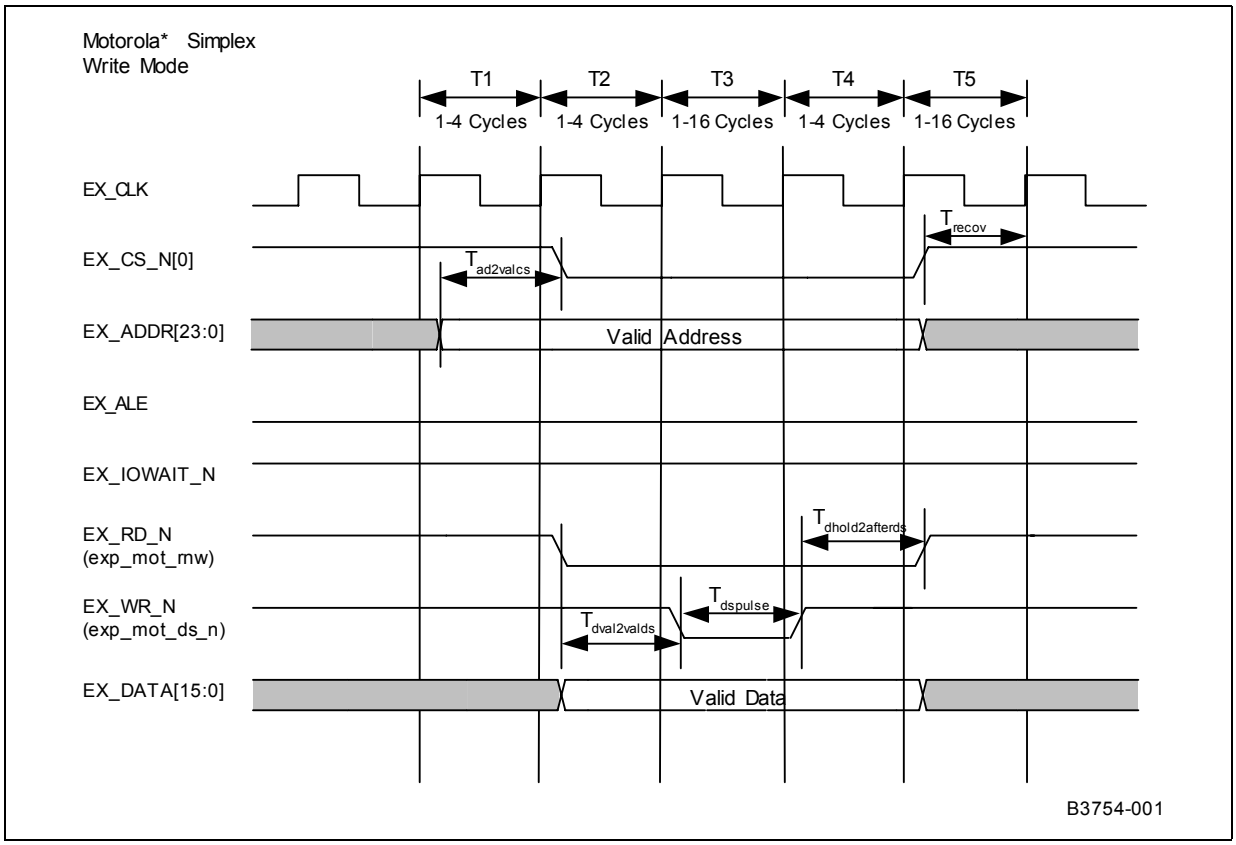


Replace datasheet Figure 28, “Motorola* Simplex Mode”, with the following two figures, “Motorola* Simplex Read Mode”, and “Motorola* Simplex Write Mode”.

Motorola* Simplex Read Mode



Motorola* Simplex Write Mode



Replace datasheet Figure 29, “HPI-8 Mode Write Accesses”, with a new figure, “HPI-8 Mode Write Accesses”, which follows.

Replace datasheet Figure 30, “HPI-8 Mode Read Accesses”, with a new figure, “HPI-8 Mode Read Accesses”, which follows.

Replace datasheet Figure 31, “HPI-16 Multiplex Write Mode”, with a new figure, “HPI-16 Multiplex Write Mode”, which follows.

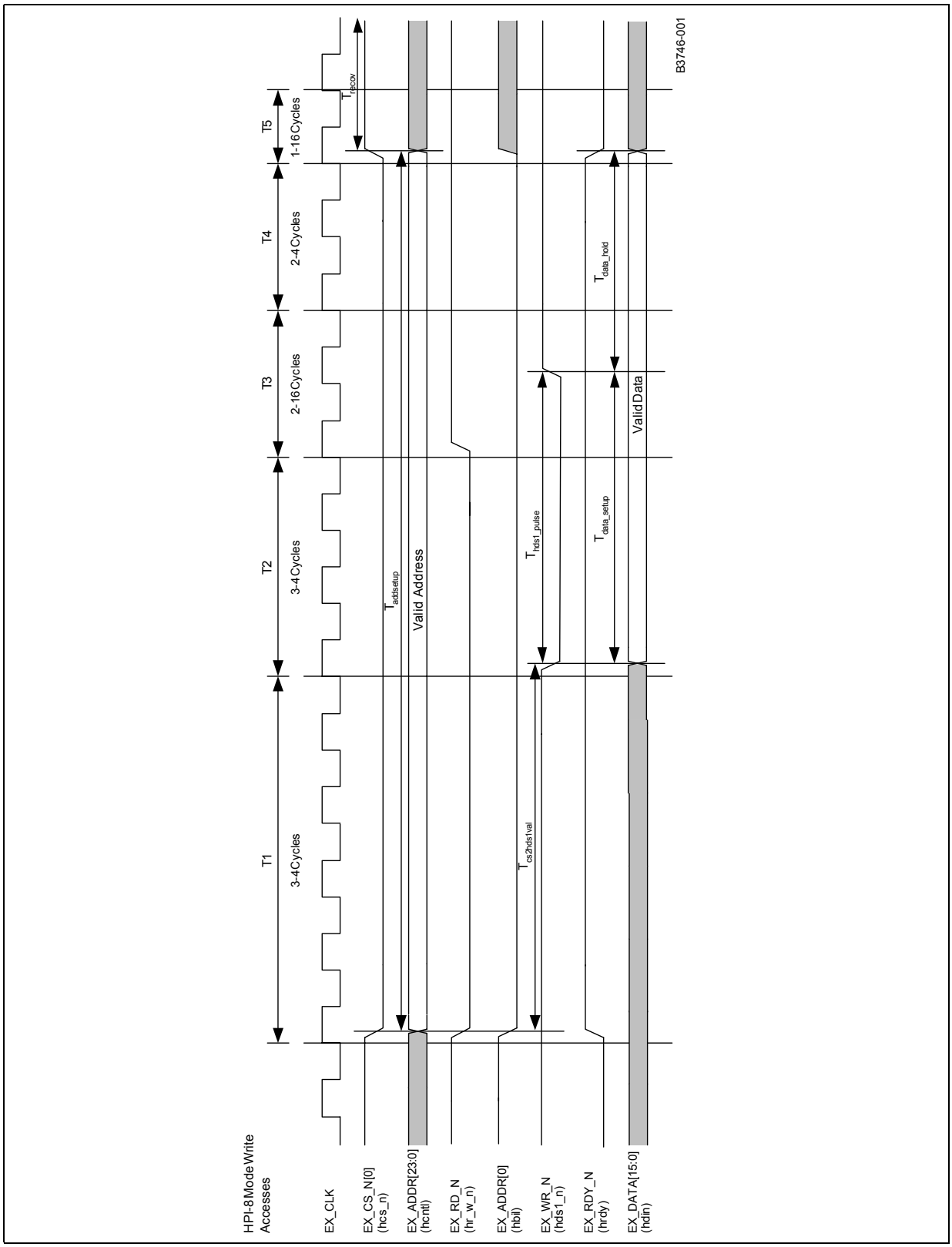
Replace datasheet Figure 32, “HPI-16 Multiplex Read Mode”, with a new figure, “HPI-16 Multiplex Read Mode”, which follows.

Replace datasheet Figure 33, “HPI-16 Simplex Read Mode”, with a new figure, “HPI-16 Simplex Read Mode”, which follows.

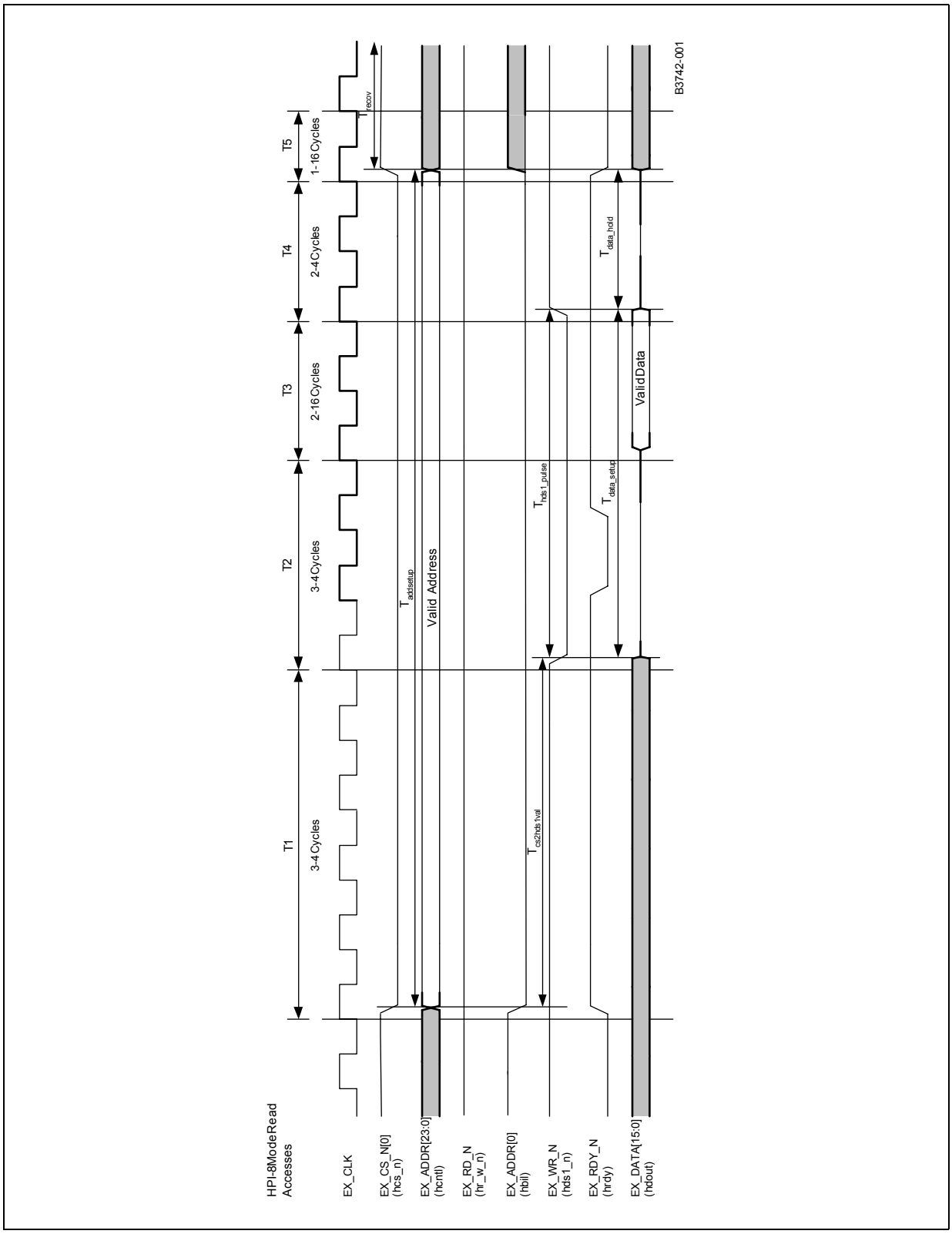
Replace datasheet Figure 34, “HPI-16 Simplex Write Mode”, with a new figure, “HPI-16 Simplex Write Mode”, which follows.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (252479-003)

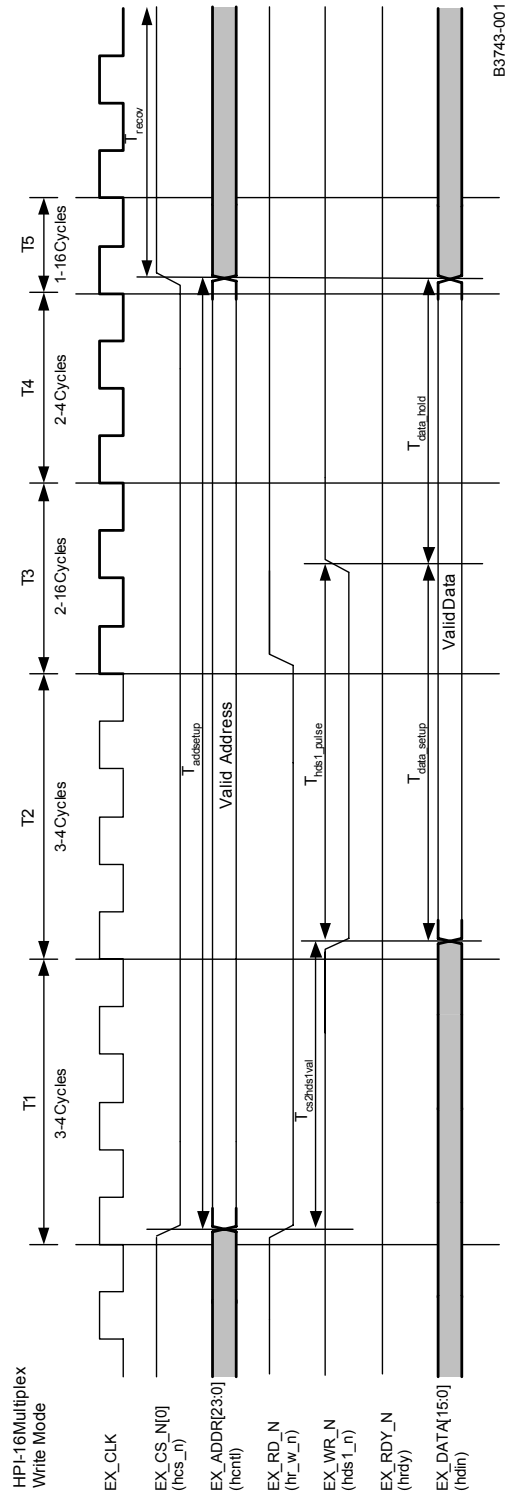
HPI-8 Mode Write Accesses



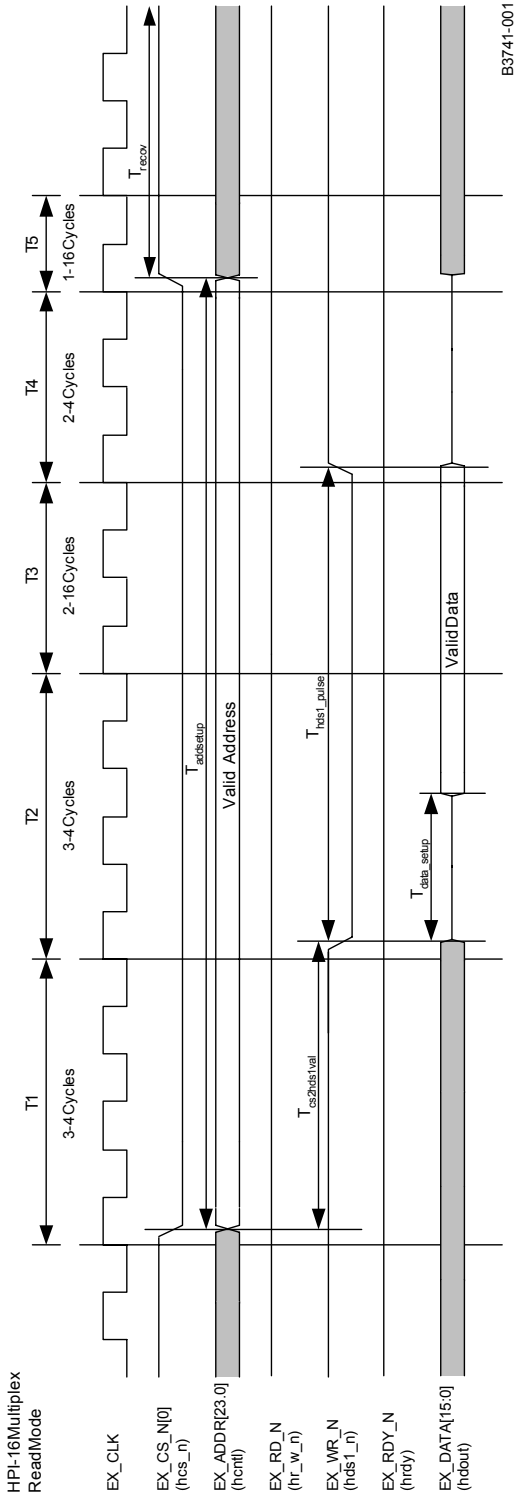
HPI-8 Mode Read Accesses



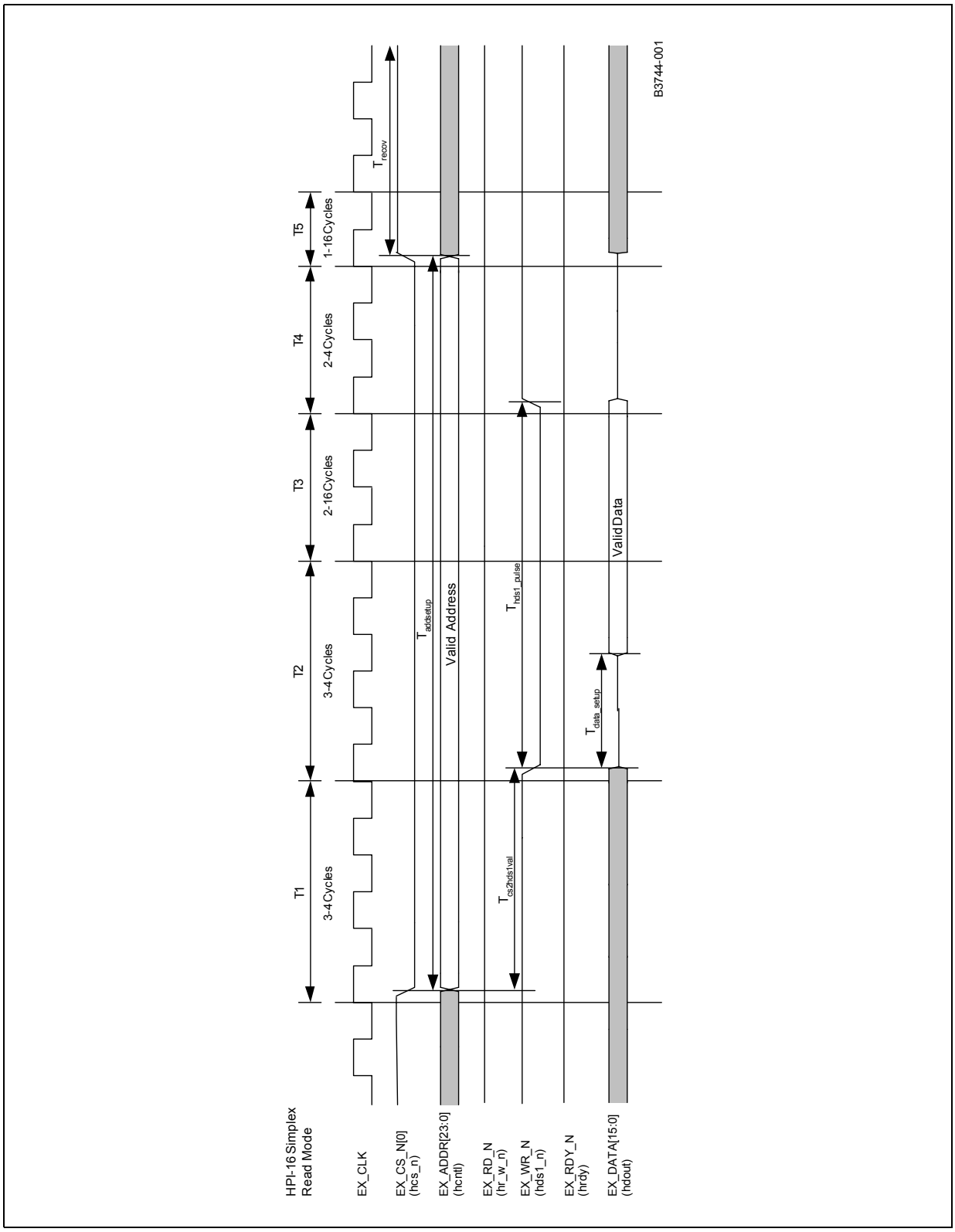
HPI-16 Multiplex Write Mode



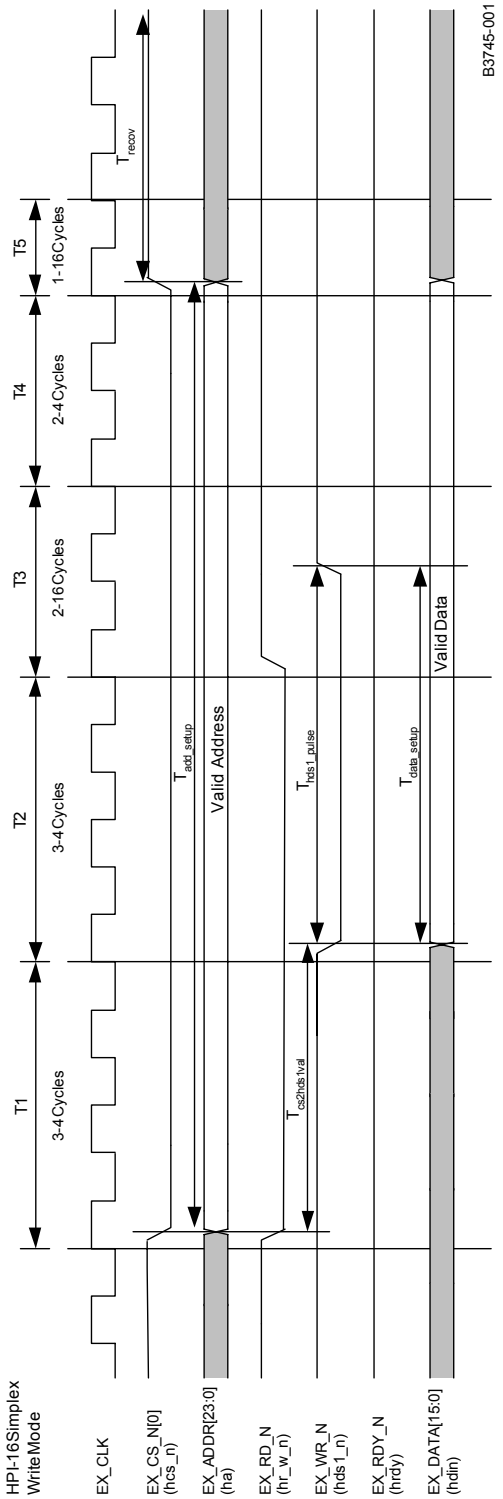
HPI-16 Multiplex Read Mode



HPI-16 Simplex Read Mode



HPI-16 Simplex Write Mode



Documentation Changes

1. Correction to Expansion Bus Label (SCR 3888)

Issue: In the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* (page 300, Figure 59), the Expansion Bus Peripheral Connection diagram incorrectly labels the Expansion Bus on the processor as EX_ADDR[3:0].

EX_ADDR[3:0] should be changed to EX_ADDR[23:0].

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* (252480-003)

2. Correction to Input Reference Slew Rate in a Oscillator Configuration (SCR 3890)

Issue: In the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet*, Figures 38 and 39, the note "Note 4. The reference-clock input slope should not exceed more than 2.5 V/nS to ensure proper PLL operation." is incomplete.

Replace note 4 with the following statement: "Note 4. Where the IXP42X is configured with an input reference-clock, the slew rate should never be faster than 2.5 V/nS to ensure proper PLL operation. To properly guarantee PLL operation at the slower slew rate, the Vih and Vil levels need to be met at the 33.33 MHz frequency."

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (252479-004)

3. Update of Management Data Output Register (SCR 4053)

Issue: In the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet*, the requirement for a 1.5K pull-up resistor on the ETH_MDIO needs to be added to Table 9 on page 36.

The table should read: "ETH_MDIO Management data output. Provides the write data to both PHY devices connected to each MII interface. Requires a 1.5K pull-up resistor.

Should be pulled low through a 10-K resistor when not being utilized in the system."

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Datasheet* (252479-004)

4. Update of Core Clock Speed Expansion Bus Configuration Table (SCR 4086)

Issue: In the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* (page 324), Table 124, "Setting the Intel XScale Core Operation Speed", is incomplete. Not all the possible expansion bus (core clock speed) configurations are listed.

Replace Table 124 with the following table:

Table 124. Intel XScale® Core Speed Expansion Bus Configuration Strappings

Intel XScale® Core Speed (Factory Part Speed)	Cfg1 EX_ADDR(23)	Cfg0 EX_ADDR(22)	Cfg_en_n EX_ADDR(21)	Actual Core Speed (MHz)
533 MHz	1	0	0	533 MHz
533 MHz	0	0	0	533 MHz
533 MHz	0	0	1	400 MHz
533 MHz	0	1	1	266 MHz
400 MHz	1	0	0	400 MHz
400 MHz	0	0	0	400 MHz
400 MHz	0	0	1	400 MHz
400 MHz	0	1	1	266 MHz
266 MHz	1	0	0	266 MHz
266 MHz	0	0	0	266 MHz
266 MHz	0	0	1	266 MHz
266 MHz	0	1	1	266 MHz

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* (252480-004)

5. Correction of PCI Controller Control and Status Register Description (SCR 4206)

Issue: In the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual*, Section 16.4.2.8, the PCI_CSR table incorrectly describes under the IC register "Initialization Complete. When at a logic 0 state, forces the PCI Controller Target Interface to retry PCI Configuration cycles. When set to a 1, configuration cycles will be accepted."

Status: The text should read: "Initialization Complete. When at a logic 0 state, forces the PCI Controller Target Interface to retry PCI cycles. When set to a 1, PCI cycles will be accepted."

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* (252480-004)

6. Correction to Expansion Bus Timing Diagrams

Problem: In the *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* in section 8.8 on page 306, the Expansion Bus timing diagrams are incorrect. Refer to errata 6 Update of Expansion Bus Timing Diagrams (SCR 4199) in the specification clarification section of this document for corrected timing diagrams.

Affected Docs: *Intel® IXP42X Product Line of Network Processors and IXC1100 Control Plane Processor Developer's Manual* (252480-004)