Intel® IXP4XX Product Line of Network Processors

Datasheet

Product Features

For a complete list of product features, see "Product Features" on page 9

- Intel® XScale™ Core
- Three Network Processor Engines
- PCI Interface
- 2-MII/RMII Interfaces
- UTOPIA-2 Interface
- USB v 1.1 Device Controller
- Two High-Speed, Serial Interfaces
- SDRAM Interface
- Encryption/Authentication
- High-Speed UART
- Console UART
- Internal Bus Performance Monitoring Unit
- 16 GPIO
- Four Internal Timers
- Packaging
  - —492-pin PBGA
  - —Commercial/Extended Temperature

Typical Applications

- High-Performance DSL Modem
- High-Performance Cable Modem
- Residential Gateway
- SME Router
- Network Printers
- Control Plane
- Integrated Access Device (IAD)
- Set-Top Box
- Access Points (802.11a/b/g)
- Industrial Controllers
- DSLAM

Document Number: 252479-001
February 2003

Contents

Figures

Tables

Revision History

Date	Revision	Description
February 2003	001	Initial release of this document. Document reissued, without “Confidential” marking.

1.0 Product Features

1.1 Product Line Features

- Intel® XScale™ Core (compliant with Intel® StrongARM* architecture)
  - — High-performance processor based on Intel® XScale™ Microarchitecture
  - — Seven/eight-stage Intel® Super-Pipelined RISC Technology
  - — Management unit
    - 32-entry, data memory management unit
    - 32-entry, instruction memory management unit
    - 32-KByte, 32-way, set associative instruction cache
    - 32-KByte, 32-way, set associative data cache
    - 2 KByte, two-way, set associative mini-data cache
    - 128-entry, branch target buffer
    - Eight-entry write buffer
    - Four-entry fill and pend buffers
  - — Clock speeds:
    - 266 MHz
    - 400 MHz
    - 533 MHz
  - — Intel® StrongARM* Version 5TE Compliant
  - — Intel® Media Processing Technology
    Multiply-accumulate coprocessor
  - — Debug unit
    Accessible through JTAG port
- Three network processor engines (NPEs)
  Used to off load typical Layer-2 networking functions like:
  - — Ethernet filtering
  - — ATM SARing
  - — HDLC
- PCI interface
  - — 32-bit interface
  - — Selectable clock
    - 33-MHz clock output
    - 0- to 66-MHz clock input
  - — *PCI Local Bus Specification*, Revision 1.1 compatible
  - — PCI arbiter supporting up to four external PCI devices (four REQ/GNT pairs)
  - — Host/option capable

  - — Master/target capable
  - — Two DMA channels
  - — High-performance support for 264-MBps peak data transfers
- 2-MII/RMII interfaces
  - — 802.3 MII interfaces that additionally support RMII interfaces
  - — Single MDIO interface to control both MII/RMII interfaces
- UTOPIA-2 Interface
  - — Eight-bit interface
  - — Up to 33 MHz clock speed
  - — Five transmit and five receive address lines
- USB v 1.1 device controller
  - — Full-speed capable
  - — Embedded transceiver
  - — 16 endpoints
- Two high-speed, serial interfaces
  - — Six-wire
  - — Supports speeds up to 8.192 MHz
  - — Supports connection to T1/E1 framers
  - — Supports connection to CODEC/SLICs
  - — Eight HDLC Channels
- SDRAM interface
  - — 32-bit data
  - — 13-bit address
  - — 133 MHz
  - — Up to eight open pages simultaneously maintained
  - — Programmable auto-refresh
  - — Programmable CAS/data delay
  - — 8 MB minimum up to 256 MB maximum supported
- Expansion interface
  - — 24-bit address
  - — 16-bit data
  - — Eight programmable chip selects
  - — Supports Intel/Motorola* microprocessors
    - Multiplexed-style bus cycles
    - Simplex-style bus cycles
- Encryption/Authentication

  - DES
  - DES3
  - AES 128-bit and 256-bit
- DSP support for:
  - Texas Instruments* DSPs supporting HPI-8 bus cycles
  - Texas Instruments* DSPs supporting HPI-16 bus cycles
- High-speed UART
  - 1,200 Baud to 921 Kbaud
  - 16550 compliant
  - 64-Byte Tx and Rx FIFOs
  - CTS and RTS modem control signals
- Console UART
  - 1,200 Baud to 921 Kbaud
  - 16550 compliant
  - 64-byte Tx and Rx FIFOs
  - CTS and RTS modem control signals
- Internal bus performance monitoring unit
  - Seven 27-bit event counters
  - Monitors internal bus occurrence and duration events
- 16 GPIOs
- Four internal timers
- Packaging
  - 492-pin PBGA
  - Commercial temperature (0° to +70° C)
  - Extended temperature (-40° to +85° C)

1.2 Specific-Model Features

Table 1. Intel® IXP4XX Product Line Features

	Intel® IXP425 B Step	Intel® IXP422	Intel® IXP421	Intel® IXP420
Processor Speed (MHz)	266 / 400 / 533	266	266	266
UTOPIA 2	X		X	
GPIO	X	X	X	X
UART 0/1	X	X	X	X
HSS 0	X		X	
HSS 1	X		X	
MII 0	X	X	X	X
MII 1	X	X		X
USB	X	X	X	X
PCI	X	X	X	X
Expansion Bus	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
AES / DES / DES3	X	X		
Multi-Channel HDLC	8		8	
SHA-1 / MD-5	X	X		
Commercial Temperature	X	X	X	X
Extended Temperature	X			

1.3 Typical Applications

- High-performance DSL modem
- High-performance cable modem
- Residential gateway
- SME router
- Integrated access device (IAD)
- Set-top box
- DSLAM
- Access Points 802.11a/b/g
- Industrial Controllers
- Network Printers
- Control Plane

2.0 About this Document

This datasheet contains a functional overview of the Intel® IXP4XX Product Line of Network Processors, as well as mechanical data (package signal locations and simulated thermal characteristics), targeted electrical specifications, and some bus functional wave forms for the device. Detailed functional descriptions — other than parametric performance — are published in the *Intel® IXP4XX Product Line Developer's Manual*.

Other related documents are shown in Table 2.

Table 2. Related Documents

Document Title	Document #
Intel® IXP4XX Product Line of Network Processors Specification Update	252702
Intel® IXP4XX Product Line of Network Processors Developer's Manual	252480
Intel® IXP4XX Product Line Programmer's Guide (Version 1.1)	252539
Intel® XScale™ Core Developer's Manual	273473
Intel® XScale™ Microarchitecture Technical Summary	—
PCI Local Bus Specification, Revision 1.1	N/A
Universal Serial Bus Specification, Revision 2.0	N/A

3.0 Functional Overview

The Intel® IXP4XX Product Line of Network Processors is compliant with the Intel® StrongARM* Version 5TE instruction-set architecture (ISA). The Intel® IXP4XX Product Line of Network Processors is designed with Intel state-of-the-art 0.18-µ production semiconductor process technology. This process technology — along with the compactness of the Intel® StrongARM* RISC ISA, simultaneous processing of three integrated Network Processing Engines, and numerous dedicated function peripheral interfaces — enables the Intel® IXP4XX Product Line of Network Processors to operate over a wide range of low-cost networking applications, with industry-leading performance.

As indicated in Figure 1 through Figure 4, the Intel® IXP4XX Product Line of Network Processors combine many features with the Intel® XScale™ core to create a highly integrated processor applicable to LAN/WAN-based networking applications in addition to other embedded networking applications.

This section briefly describes the main features of the product. For detailed functional descriptions, see the *Intel® IXP4XX Product Line Developer's Manual* (252480).

Figure 1. Intel® IXP425 Network Processor: Block Diagram

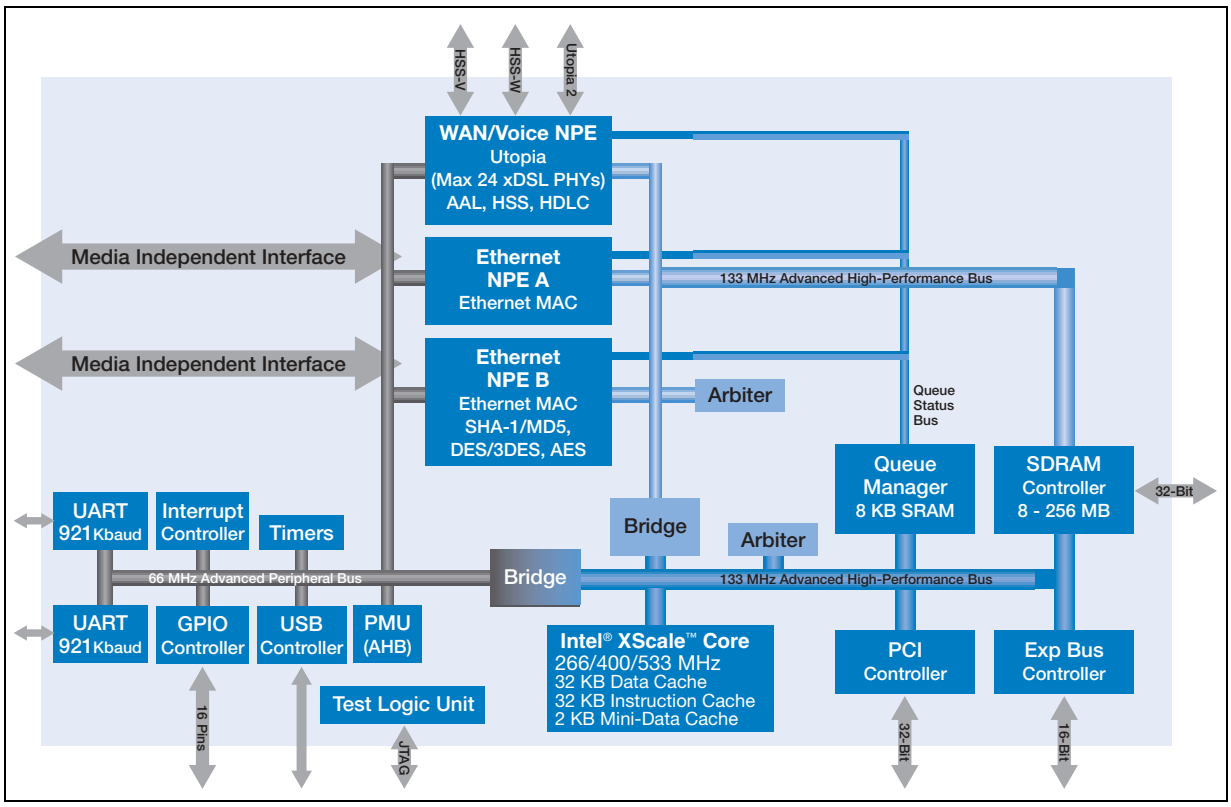


Figure 2. Intel® IXP422 Network Processor: Block Diagram

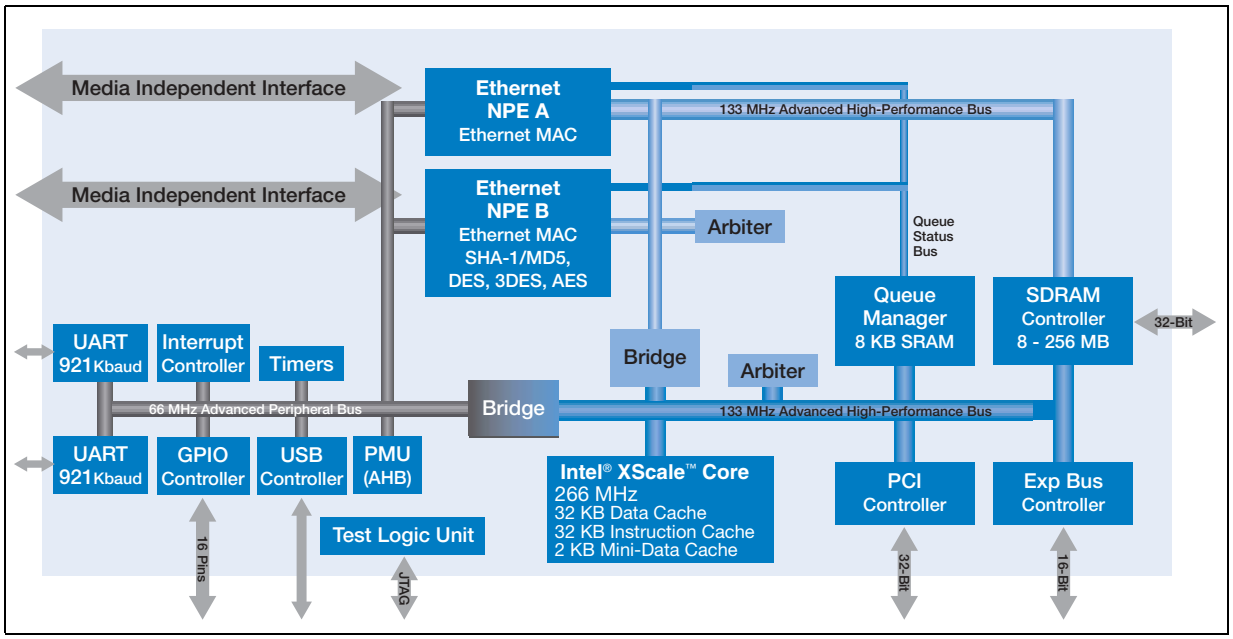


Figure 3. Intel® IXP421 Network Processor: Block Diagram

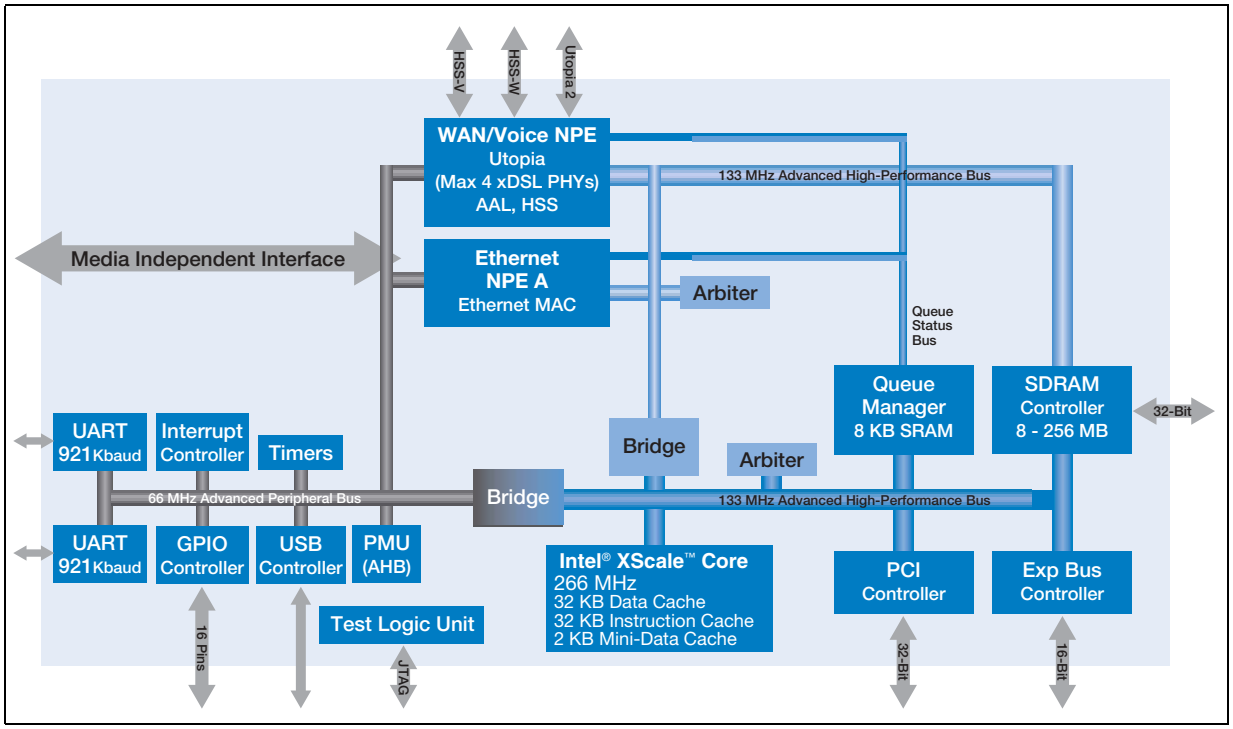
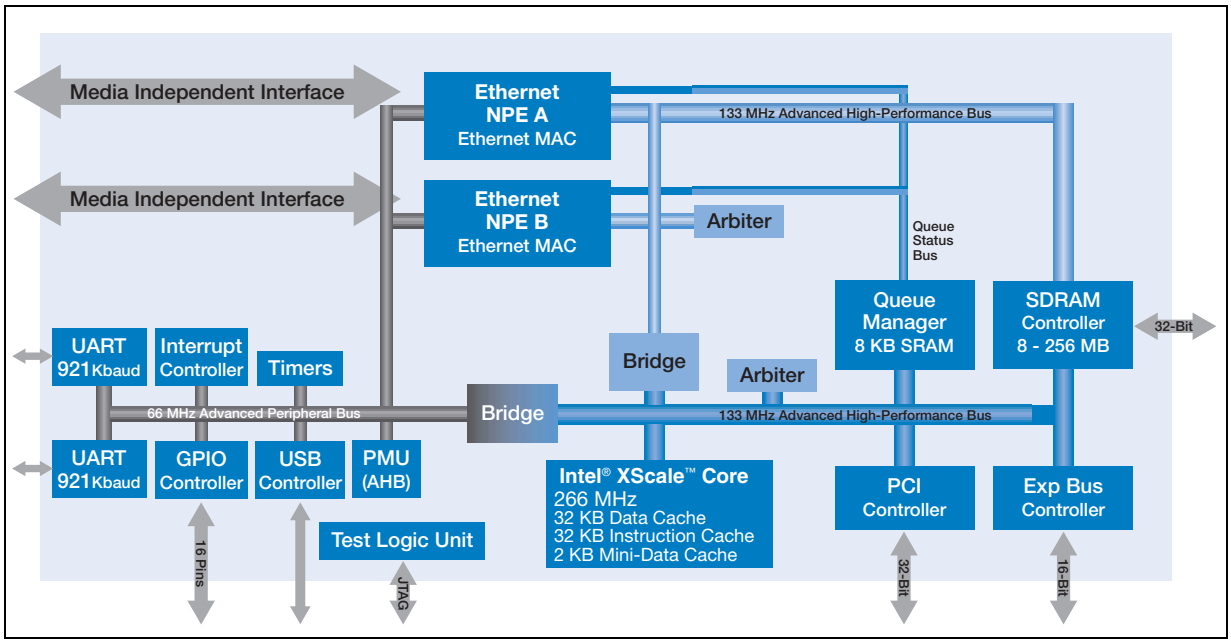


Figure 4. Intel® IXP420 Network Processor: Block Diagram



3.1 Key Functional Units

The following sections briefly describe the functional units and their interaction in the system. For more detailed information, refer to the *Intel® IXP4XX Product Line Developer's Manual* (252480).

Unless otherwise specified, the functional descriptions apply to all processors in the Intel® IXP4XX Product Line. Refer to Table 1 on page 12 and Figure 1 on page 14 through Figure 4 for specific information on supported interfaces

3.1.1 Network Processor Engines (NPEs)

The network processor engines (NPEs) are dedicated-function processors containing hardware coprocessors integrated into the Intel® IXP4XX Product Line of Network Processors. The network processor engines are used to off load processing function required by the Intel® XScale™ core.

These network-processor engines are high-performance, hardware-multi-threaded processors with additional local-hardware-assist functionality used to off load highly processor-intensive functions such as MII (MAC), CRC checking/generation, AAL 2, AES, DES, SHA-1, and MD5. All instruction code for the NPE processors are stored locally with a dedicated instruction memory bus and dedicated data memory bus.

These engines support processing of the dedicated peripherals that can include:

- A Universal Test and Operation PHY Interface for ATM (UTOPIA) 2 interface
- Two High-Speed Serial (HSS) interfaces
- Two Media-Independent Interface (MII) / Reduced Media Independent Interface (RMII) interfaces

Table 3 specifies which devices, in the Intel® IXP4XX Product Line, have which of these capabilities.

Table 3. Network Processor Functions of Intel® IXP4XX Product Line of Network Processors

Device	UTOPIA	HSS	MII 0	MII 1	AES / DES / DES3	Multi-Channel HDLC	SHA-1 / MD-5
Intel® IXP425, B-Step	X	X	X	X	X	8	X
Intel® IXP422			X	X	X		X
Intel® IXP421	X	X	X			8	
Intel® IXP420			X	X			

The NPE core is a hardware-multi-threaded processor engine that is used to accelerate functions that are difficult to achieve high performance in a standard RISC processor. Each NPE core is a 133-MHz processor core that has self-contained instruction memory and self-contained data memory that operate in parallel.

In addition to having separate instruction/data memory and local-code store, the NPE core supports hardware multi-threading with support for multiple contexts. The support of hardware multi-threading creates an efficient processor engine with minimal processor stalls due to the ability of the processor core to switch contexts in a single clock cycle, based on a prioritized/preemptive basis. The prioritized/preemptive nature of the context switching allows time-critical applications to be implemented in a low-latency fashion — which is required when processing multi-media applications.

The NPE core also connects several hardware-based coprocessors that are used to implement functions that are difficult for a processor to implement. These functions include:

- Serialization/De-serialization
- DES/3DES/AES
- MD5
- CRC checking/generation
- SHA-1
- HDLC bit stuffing/de-stuffing

These coprocessors are implemented in hardware, enabling the coprocessors and the NPE processor core to operate in parallel.

The combined forces of the hardware multi-threading, local-code store, independent instruction memory, independent data memory, and parallel processing allows the Intel® XScale™ core to be utilized for application purposes. The multi-processing capability of the peripheral interface functions allows unparalleled performance to be achieved by the application running on the Intel® XScale™ core.

3.1.2 Internal Bus

The internal bus architecture of the Intel® IXP4XX Product Line of Network Processors is designed to allow parallel processing to occur and to isolate bus utilization, based on particular traffic patterns. The bus is segmented into three major buses, the North AHB, South AHB, and APB.

3.1.2.1 North AHB

The North AHB is a 133-MHz, 32-bit bus that can be mastered by the WAN/Voice NPE or both of the Ethernet NPEs. The targets of the North AHB can be the SDRAM or the AHB/AHB bridge. The AHB/AHB bridge allows the NPEs to access the peripherals and internal targets on the South AHB.

Data transfers by the NPEs on the North AHB to the South AHB are targeted predominately to the queue manager. Transfers to the AHB/AHB bridge may be “posted,” when writing, or “split,” when reading.

When a transaction is “posted,” a master on the North AHB requests a write to a peripheral on the South AHB. If the AHB/AHB Bridge has a free FIFO location, the write request will be transferred from the master on the North AHB to the AHB/AHB bridge. The AHB/AHB bridge will complete the write on the South AHB, when it can obtain access to the peripheral on the South AHB. The North AHB is released to complete another transaction.

When a transaction is “split,” a master on the North AHB requests a read of a peripheral on the South AHB. If the AHB/AHB bridge has a free FIFO location, the read request will be transferred from the master on the North AHB to the AHB/AHB bridge. The AHB/AHB bridge will complete the read on the South AHB, when it can obtain access to the peripheral on the South AHB.

Once the AHB/AHB bridge has obtained the read information from the peripheral on the South AHB, the AHB/AHB bridge notifies the arbiter, on the North AHB, that the AHB/AHB bridge has the data for the master that requested the “split” transfer. The master on the North AHB — that requested the split transfer — will arbitrate for the North AHB and transfer the read data from the AHB/AHB bridge. The North AHB is released to complete another transaction while the North AHB master — that requested the “split” transfer — waits for the data to arrive.

These “posting” and “splitting” transfers allow control of the North AHB to be given to another master on the North AHB — enabling the North AHB to achieve maximum efficiency. Transfers to the AHB/AHB bridge are considered to be small and infrequent, relative to the traffic passed between the NPEs on the North AHB and the SDRAM.

3.1.2.2 South AHB

The South AHB is a 133-MHz, 32-bit bus that can be mastered by the Intel® XScale™ core, PCI controller, and the AHB/AHB bridge. The targets of the South AHB Bus can be the SDRAM, PCI interface, queue manager, expansion bus, or the APB/AHB bridge.

Accessing across the APB/AHB bridge allows interfacing to peripherals attached to the APB.

3.1.2.3 APB Bus

The APB Bus is a 66-MHz, 32-bit bus that can be mastered by the AHB/APB bridge only. The targets of the APB bus can be:

- The high-speed UART interface
- USB v 1.1 interface
- The internal bus performance monitoring unit (IBPMU)
- GPIO
- Console UART interface
- All NPEs
- Interrupt controller
- Timers

The APB interface is also used as an alternate-path interface to the NPEs and is used for NPE code download and configuration.

3.1.3 MII/RMII Interfaces

Two industry-standard, media-independent interface (MII) interfaces are integrated into most of the Intel® IXP4XX Product Line of Network Processors with separate media-access controllers and independent network processing engines. (See Table 3 on page 17.)

The independent NPEs and MACs allow parallel processing of data traffic on the MII interfaces and off loading of processing required by the Intel® XScale™ core. The Intel® IXP4XX Product Line of Network Processors are compliant with the IEEE, 802.3 specification.

In addition to two MII interfaces, the Intel® IXP4XX Product Line of Network Processors include a single management data interface that is used to configure and control PHY devices that are connected to the MII interface. The Intel® IXP4XX Product Line of Network Processors provides support for the reduced media independent interface (RMII) interfaces that is shared with the pins of the MII interface.

3.1.4 UTOPIA 2

The integrated, UTOPIA-2 interface works with a network processing engine, for several of the Intel® IXP4XX Product Line of Network Processors. (See Table 3 on page 17.)

The UTOPIA-2 interface supports a single- or a multiple-physical-interface configuration with cell-level or octet-level handshaking. The network processing engine handles segmentation and reassembly of ATM cells, CRC checking/generation, and transfer of data to/from memory. This allows parallel processing of data traffic on the UTOPIA-2 interface, off loading processor overhead required by the Intel® XScale™ core.

The Intel® IXP4XX Product Line of Network Processors are compliant with the ATM Forum, *UTOPIA Level-2 Specification*, Revision 1.0.

3.1.5 USB v 1.1 Interface

The integrated USB v 1.1 interface is a device-only controller. The interface supports full-speed operation and 16 endpoints and includes an integrated transceiver.

There are:

- Six isochronous endpoints (three input and three output)
- One control endpoints
- Three interrupt endpoints
- Six bulk endpoints (three input and three output)

3.1.6 PCI Controller

The PCI bus is an industry-standard, high-performance, low-latency system bus that operates up to 264 Mbps.

The Intel® IXP4XX Product Line of Network Processors are compatible with the *PCI Local Bus Specification*, Revision 1.1.

3.1.7 SDRAM Controller

The memory controller manages an interface to external SDRAM memory chips. The interface:

- Operates at 133 MHz
- Supports eight open pages simultaneously
- Has two banks to support memory configurations from 8 Mbyte to 256 Mbyte

The memory controller interface is required to support 32-bit memory only. If a x16 memory chip is used, a minimum of two memory chips would be required to facilitate the 32-bit interface required by the Intel® IXP4XX Product Line of Network Processors. A maximum of four SDRAM memory chips may be attached to the Intel® IXP4XX Product Line of Network Processors. For more information on SDRAM support and configuration see the *Intel® IXP4XX Product Line of Network Processors Developer's Manual* (252480).

The memory controller internally interfaces to the North AHB and South AHB with independent interfaces. This architecture allows SDRAM transfers to be interleaved and pipelined to achieve maximum possible efficiency.

The maximum burst size supported to the SDRAM interface is eight 32-bit words. This burst size allows the best efficiency/fairness performance between accesses from the North AHB and the South AHB.

3.1.8 Expansion Interface

The expansion interface allows easy and — in most cases — glue-less connection to slow-speed peripheral devices. It also provides input information for device configuration after reset. Some of the peripheral device types are flash, ATM control interfaces, and DSPs used for voice applications. (Some voice configurations can be supported by the HSS interfaces and the Intel® XScale™ core, implementing voice-compression algorithms.)

The expansion interface is a 16-bit interface that allows an address range of 512 bytes to 16 Mbytes, using 24 address lines for each of the eight independent chip selects.

Accesses to the expansion-bus interface consists of five phases. Each of the five phases can be lengthened or shortened by setting various configuration registers on a per-chip-select basis. This feature allows the Intel® IXP4XX Product Line of Network Processors to connect to a wide variety of peripheral devices with varying speeds.

The expansion interface supports Intel or Motorola microprocessor-style bus cycles. The bus cycles can be configured to be multiplexed address/data cycles or separate address/data cycles for each of the eight chip-selects.

Additionally, Chip Selects 4 through 7 can be configured to support Texas Instruments HPI-8 or HPI-16 style accesses for DSPs.

The expansion interface is an asynchronous interface to externally connected chips. However, a clock must be supplied to the Intel® IXP4XX Product Line of Network Processors' expansion interface for the interface to operate. This clock can be driven from GPIO 15 or an external source. The maximum clock rate that the expansion interface can accept is 33 MHz.

At the de-assertion of reset, the 24-bit address bus is used to capture configuration information from the levels that are applied to the pins at this time. External pull-up/pull-down resistors are used to tie the signals to particular logic levels. (For additional details, see "Package Information" on page 30.)

3.1.9 High-Speed, Serial Interfaces

The high-speed, serial interfaces are six-signal interfaces that supports serial transfer speeds from 512 KHz to 8.192 MHz, for some models of the Intel® IXP4XX Product Line of Network Processors. (See Table 3 on page 17.)

Each interface allows direct connection of up to four T1/E1 framers and CODEC/SLICs to the Intel® IXP4XX Product Line of Network Processors. The high-speed, serial interfaces are capable of supporting various protocols, based on the implementation of the code developed for the network processor engine core. For a list of supported protocols, see the *Intel® IXP4XX Product Line Programmer's Guide (Version 1.1)* (252539).

3.1.10 High-Speed UART

The high-speed UART interface is a 16550-compliant UART with the exception of transmit and receive buffers. Transmit and receive buffers are 64 bytes-deep versus the 16 bytes required by the 16550 UART specification.

The interface can be configured to support speeds from 1,200 Baud to 921 Kbaud. The interface support configurations of:

- Five, six, seven, or eight data-bit transfers
- One or two stop bits
- Even, odd, or no parity

The request-to-send (RTS_N) and clear-to-send (CTS_N) modem control signals also are available with the interface.

3.1.11 Console UART

The console UART interface exhibits the same features as the high-speed UART.

3.1.12 GPIO

There are 16 GPIO pins supported by the Intel® IXP4XX Product Line of Network Processors. GPIO pins 0 through 13 can be configured to be general-purpose input or general-purpose output. Additionally, GPIO pins 0 through 12 can be configured to be an interrupt input.

GPIO Pin 14 can be configured the same as GPIO pin 13 or as a clock output. The output-clock configuration can be set at various speeds, up to 33 MHz, with various duty cycles. GPIO Pin 14 is configured as an input, upon reset.

GPIO Pin 15 can be configured the same as GPIO pin 13 or as a clock output. The output-clock configuration can be set at various speeds, up to 33 MHz, with various duty cycles. GPIO Pin 15 is configured as an output, upon reset. GPIO Pin 15 can be used to clock the expansion interface, after reset.

3.1.13 Internal Bus Performance Monitoring Unit (IBPMU)

The Intel® IXP4XX Product Line of Network Processors consists of seven 27-bit counters that may be used to capture predefined durations or occurrence events on the North AHB, South AHB, or SDRAM controller page hits/misses.

3.1.14 Interrupt Controller

The Intel® IXP4XX Product Line of Network Processors consists of 32 interrupt sources to allow an extension of the Intel® XScale™ core FIQ and IRQ interrupt sources. These sources can originate from some external GPIO pins or internal peripheral interfaces.

The interrupt controller can configure each interrupt source as an FIQ, IRQ, or disabled. The interrupt sources tied to Interrupt 0 to 7 can be prioritized. The remaining interrupts are prioritized in ascending order. For example, Interrupt 8 has a higher priority than 9, 9 has a higher priority than 10, and 30 has a higher priority that 31.

3.1.15 Timers

The Intel® IXP4XX Product Line of Network Processors consists of four internal timers operating at 66MHz to allow task scheduling and prevent software lock-ups. The device has four 32-bit counters named:

- Watch-Dog Timer
- Timestamp Timer
- Two general-purpose timers

3.2 Intel® XScale™ Core

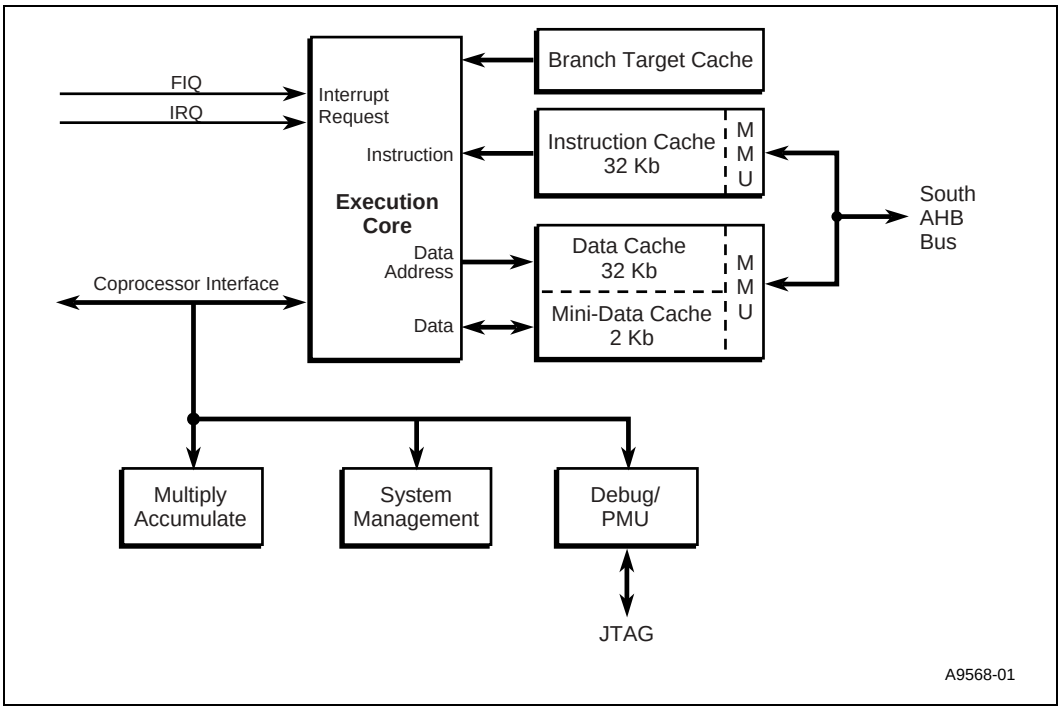
The Intel® XScale™ core technology is compliant with the Intel® StrongARM* Version 5TE instruction-set architecture (ISA). The Intel® XScale™ core — shown in Figure 5 — is designed with Intel state-of-the-art, 0.18-µ-production semiconductor process technology. This process technology — with the compactness of the Intel® StrongARM* RISC ISA — enables the Intel® XScale™ core to operate over a wide speed and power range, producing industry-leading mW/MIPS performance.

Intel® XScale™ core features include:

- Seven/eight-stage super-pipeline promotes high-speed, efficient core performance
- 128-entry branch target buffer keeps pipeline filled with statistically correct branch choices
- 32-entry instruction memory-management unit for logical-to-physical address translation, access permissions, I-cache attributes
- 32-entry data-memory management unit for logical-to-physical address translation, access permissions, D-cache attributes
- 32-Kbyte instruction cache can hold entire programs, preventing core stalls caused by multi-cycle memory accesses
- 32-Kbyte data cache reduces core stalls caused by multi-cycle memory accesses
- 2-Kbyte mini-data cache for frequently changing data streams avoids “thrashing” of the D-cache
- Four-entry fill-and-pend buffers to promote core efficiency by allowing “hit-under-miss” operation with data caches
- Eight-entry write buffer allows the core to continue execution while data is written to memory
- Multiple-accumulate coprocessor that can do two simultaneous, 16-bit, SIMD multiplies with 40-bit accumulation for efficient, high-quality media and signal processing
- Performance monitoring unit (PMU) furnishing two 32-bit event counters and one 32-bit cycle counter for analysis of hit rates, etc.

  This PMU is for the Intel® XScale™ core only. An additional PMU is supplied for monitoring of internal bus performance.
- JTAG debug unit that uses hardware break points and 256-entry trace history buffer (for flow-change messages) to debug programs

Figure 5. Intel® XScale™ Core Block Diagram



3.2.1 Super Pipeline

The super pipeline is composed of integer, multiply-accumulate (MAC), and memory pipes.

The integer pipe has seven stages:

- Branch Target Buffer (BTB)/Fetch 1
- Fetch 2
- Decode
- Register File/Shift
- ALU Execute
- State Execute
- Integer Writeback

The memory pipe has eight stages:

- The first five stages of the Integer pipe (BTB/Fetch 1 through ALU Execute)
  . . . then finish with the following memory stages
- Data Cache 1
- Data Cache 2
- Data Cache Writeback

The MAC pipe has six to nine stages:

- The first four stages of the Integer pipe (BTB/Fetch 1 through Register File/ Shift) . . . then finish with the following MAC stages
- MAC 1
- MAC 2
- MAC 3
- MAC 4
- Data Cache Writeback

The MAC pipe supports a data-dependent early terminate where stages MAC 2, MAC 3, and/or MAC 4 are bypassed.

Deep pipes promote high instruction execution rates only when a means exists to successfully predict the outcome of branch instructions. The branch target buffer provides such a means.

3.2.2 Branch Target Buffer (BTB)

Each entry of the 128-entry BTB contains the address of a branch instruction, the target address associated with the branch instruction, and a previous history of the branch being taken or not taken. The history is recorded as one of four states:

- Strongly taken
- Weakly taken
- Weakly not taken
- Strongly not taken

The BTB can be enabled or disabled via Coprocessor 15, Register 1.

When the address of the branch instruction hits in the BTB and its history is strongly or weakly taken, the instruction at the branch target address is fetched. When its history is strongly or weakly not-taken, the next sequential instruction is fetched. In either case the history is updated.

Data associated with a branch instruction enters the BTB the first time the branch is taken. This data enters the BTB in a slot with a history of strongly not-taken (overwriting previous data when present).

Successfully predicted branches avoid any branch-latency penalties in the super pipeline. Unsuccessfully predicted branches result in a four to five cycle branch-latency penalty in the super-pipeline.

3.2.3 Instruction Memory Management Unit (IMMU)

For instruction pre-fetches, the IMMU controls logical-to-physical address translation, memory access permissions, memory-domain identifications, and attributes (governing operation of the instruction cache). The IMMU contains a 32-entry, fully associative instruction-translation, look-aside buffer (ITLB) that has a round-robin replacement policy. ITLB entries zero through 30 can be locked.

When an instruction pre-fetch misses in the ITLB, the IMMU invokes an automatic table-walk mechanism that fetches an associated descriptor from memory and loads it into the ITLB. The descriptor contains information for logical-to-physical address translation, memory-access permissions, memory-domain identifications, and attributes governing operation of the I-cache.

The IMMU then continues the instruction pre-fetch by using the address translation just entered into the ITLB. When an instruction pre-fetch hits in the ITLB, the IMMU continues the pre-fetch using the address translation already resident in the ITLB.

Access permissions for each of up to 16 memory domains can be programmed. When an instruction pre-fetch is attempted to an area of memory in violation of access permissions, the attempt is aborted and a pre-fetch abort is sent to the core for exception processing. The IMMU and DMMU can be enabled or disabled together.

3.2.4 Data Memory Management Unit (DMMU)

For data fetches, the DMMU controls logical-to-physical address translation, memory-access permissions, memory-domain identifications, and attributes (governing operation of the data cache or mini-data cache and write buffer). The DMMU contains a 32-entry, fully associative data-translation, look-aside buffer (DTLB) that has a round-robin replacement policy. DTLB entries 0 through 30 can be locked.

When a data fetch misses in the DTLB, the DMMU invokes an automatic table-walk mechanism that fetches an associated descriptor from memory and loads it into the DTLB. The descriptor contains information for logical-to-physical address translation, memory-access permissions, memory-domain identifications, and attributes (governing operation of the D-cache or mini-data cache and write buffer).

The DMMU continues the data fetch by using the address translation just entered into the DTLB. When a data fetch hits in the DTLB, the DMMU continues the fetch using the address translation already resident in the DTLB.

Access permissions for each of up to 16 memory domains can be programmed. When a data fetch is attempted to an area of memory in violation of access permissions, the attempt is aborted and a data abort is sent to the core for exception processing.

The IMMU and DMMU can be enabled or disabled together.

3.2.5 Instruction Cache (I-Cache)

The I-cache can contain high-use, multiple-code segments or entire programs, allowing the core access to instructions at core frequencies. This prevents core stalls caused by multi-cycle accesses to external memory.

The 32-Kbyte I-cache is 32-set/32-way associative, where each set contains 32 ways and each way contains a tag address, a cache line of instructions (eight 32-bit words and one parity bit per word), and a line-valid bit. For each of the 32 sets, 0 through 28 ways can be locked. Unlocked ways are replaceable via a round-robin policy.

The I-cache can be enabled or disabled. Attribute bits within the descriptors — contained in the ITLB of the IMMU — provide some control over an enabled I-cache.

When a needed line (eight 32-bit words) is not present in the I-cache, the line is fetched (critical word first) from memory via a two-level, deep-fetch queue. The fetch queue allows the next instruction to be accessed from the I-cache, but only when its data operands do not depend on the execution results of the instruction being fetched via the queue.

3.2.6 Data Cache (D-Cache)

The D-cache can contain high-use data such as lookup tables and filter coefficients, allowing the core access to data at core frequencies. This prevents core stalls caused by multi-cycle accesses to external memory.

The 32-Kbyte D-cache is 32-set/32-way associative, where each set contains 32 ways and each way contains a tag address, a cache line (32 bytes with one parity bit per byte) of data, two dirty bits (one for each of two eight-byte groupings in a line), and one valid bit. For each of the 32 sets, zero through 28 ways can be locked, unlocked, or used as local SRAM. Unlocked ways are replaceable via a round-robin policy.

The D-cache (together with the mini-data cache) can be enabled or disabled. Attribute bits within the descriptors, contained in the DTLB of the DMMU, provide significant control over an enabled D-cache. These bits specify cache operating modes such as read and write allocate, write-back, write-through, and D-cache versus mini-data cache targeting.

The D-cache (and mini-data cache) work with the load buffer and pend buffer to provide “hit-under-miss” capability that allows the core to access other data in the cache after a “miss” is encountered. The D-cache (and mini-data cache) works in conjunction with the write buffer for data that is to be stored to memory.

3.2.7 Mini-Data Cache

The mini-data cache can contain frequently changing data streams such as MPEG video, allowing the core access to data streams at core frequencies. This prevents core stalls caused by multi-cycle accesses to external memory. The mini-data cache relieves the D-cache of data “thrashing” caused by frequently changing data streams.

The 2-Kbyte, mini-data cache is 32-set/two-way associative, where each set contains two ways and each way contains a tag address, a cache line (32 bytes with one parity bit per byte) of data, two dirty bits (one for each of two eight-byte groupings in a line), and a valid bit. The mini-data cache uses a round-robin replacement policy, and cannot be locked.

The mini-data cache (together with the D-cache) can be enabled or disabled. Attribute bits contained within a coprocessor register specify operating modes write and/or read allocate, write-back, and write-through.

The mini-data cache (and D-cache) work with the load buffer and pend buffer to provide “hit-under-miss” capability that allows the core to access other data in the cache after a “miss” is encountered. The mini-data cache (and D-cache) works in conjunction with the write buffer for data that is to be stored to memory.

3.2.8 Fill Buffer (FB) and Pend Buffer (PB)

The four-entry fill buffer (FB) works with the core to hold non-cacheable loads until the bus controller can act on them. The FB and the four-entry pend buffer (PB) work with the D-cache and mini-data cache to provide “hit under-miss” capability, allowing the core to seek other data in the caches while “miss” data is being fetched from memory.

The FB can contain up to four unique “miss” addresses (logical), allowing four “misses” before the core is stalled. The PB holds up to four addresses (logical) for additional “misses” to those addresses that are already in the FB. A coprocessor register can specify draining of the fill and pend (write) buffers.

3.2.9 Write Buffer (WB)

The write buffer (WB) holds data for storage to memory until the bus controller can act on it. The WB is eight entries deep, where each entry holds 16 bytes. The WB is constantly enabled and accepts data from the core, D-cache, or mini-data cache.

Coprocessor 15, Register 1 specifies whether WB coalescing is enabled or disabled. When coalescing is disabled, stores to memory occur in program order regardless of the attribute bits within the descriptors located in the DTLB. When coalescing is enabled, the attribute bits within the descriptors located in the DTLB are examined to determine when coalescing is enabled for the destination region of memory. When coalescing is enabled in both CP15, R1 and the DTLB, data entering the WB can coalesce with any of the eight entries (16 bytes) and be stored to the destination memory region, but possibly out of program order.

Stores to a memory region specified to be non-cacheable and non-bufferable by the attribute bits within the descriptors located in the DTLB causes the core to stall until the store completes. A coprocessor register can specify draining of the write buffer.

3.2.10 Multiply-Accumulate Coprocessor (CP0)

For efficient processing of high-quality, media-and-signal-processing algorithms, CP0 provides 40-bit accumulation of 16 x 16, dual-16 x 16 (SIMD), and 32 x 32 signed multiplies. Special MAR and MRA instructions are implemented to move the 40-bit accumulator to two core-general registers (MAR) and move two core-general registers to the 40-bit accumulator (MRA). The 40-bit accumulator can be stored or loaded to or from D-cache, mini-data cache, or memory using two STC or LDC instructions.

The 16 x 16 signed multiply-accumulates (MIAxy) multiply either the high/high, low/low, high/low, or low/high 16 bits of a 32-bit core general register (multiplier) and another 32-bit core general register (multiplicand) to produce a full, 32-bit product that is sign-extended to 40 bits and added to the 40-bit accumulator.

Dual-signed, 16 x 16 (SIMD) multiply-accumulates (MIAPH) multiply the high/high and low/low 16-bits of a packed 32-bit, core-general register (multiplier) and another packed 32-bit, core-general register (multiplicand) to produce two 16-bits products that are both sign-extended to 40 bits and added to the 40-bit accumulator.

The 32 x 32 signed multiply-accumulates (MIA) multiply a 32-bit, core-general register (multiplier) and another 32-bit, core-general register (multiplicand) to produce a 64-bit product where the 40 LSBs are added to the 40-bit accumulator. The 16 x 32 versions of the 32 x 32 multiply-accumulate instructions complete in a single cycle.

3.2.11 Performance Monitoring Unit (PMU)

The performance monitoring unit contains two 32-bit, event counters and one 32-bit, clock counter. The event counters can be programmed to monitor I-cache hit rate, data caches hit rate, ITLB hit rate, DTLB hit rate, pipeline stalls, BTB prediction hit rate, and instruction execution count.

3.2.12 Debug Unit

The debug unit is accessed through the JTAG port. The industry-standard, IEEE 1149.1 JTAG port consists of a test access port (TAP) controller, boundary-scan register, instruction and data registers, and dedicated signals TDI, TDO, TCK, TMS, and TRST#.

The debug unit — when used with debugger application code running on a host system outside of the Intel® XScale™ core — allows a program, running on the Intel® XScale™ core, to be debugged. It allows the debugger application code or a debug exception to stop program execution and redirect execution to a debug handling routine.

Debug exceptions are instruction breakpoint, data breakpoint, software breakpoint, external debug breakpoint, exception vector trap, and trace buffer full breakpoint. Once execution has stopped, the debugger application code can examine or modify the core's state, coprocessor state, or memory. The debugger application code can then restart program execution.

The debug unit has two hardware-instruction, break point registers; two hardware, data-breakpoint registers; and a hardware, data-breakpoint control register. The second data-breakpoint register can be alternatively used as a mask register for the first data-breakpoint register.

A 256-entry trace buffer provides the ability to capture control flow messages or addresses. A JTAG instruction (LDIC) can be used to download a debug handler via the JTAG port to the mini-instruction cache (the I-cache has a 2-Kbyte, mini-instruction cache, like the mini-data cache, that is used only to hold a debug handler).

4.0 Package Information

The Intel® IXP4XX Product Line of Network Processors have a 492-ball, plastic ball grid array (PBGA) package for commercial-temperature applications and a pin-for-pin, compatible 492-ball, plastic ball grid array with a drop-in heat spreader (H) for extended-temperature applications.

4.1 Functional Signal Definitions

Listed in the signal definition tables — starting at Section 5, “SDRAM Interface” on page 31 — are optional pull-up an pull-down resistor recommendations that are required when the particular *enabled* interface is not being used in the application. These external resistor requirements are only needed if the particular model of Intel® IXP4XX Product Line processor has the particular interface *enabled* and the interface is not required in the application.

Warning: All Intel® IXP4XX Product Line of Network Processors I/O pins are *not* 5-V tolerant.

Disabled features, within the Intel® IXP4XX Product Line of Network Processors, do not require external resistors as the processor will have internal pull-up or pull-down resistors enabled as part of the *disabled* interface.

Table 4 presents the legend for interpreting the **Type** field in the other tables in this section of the document.

See Table 1 on page 12 to determine which interfaces are not enabled within the Intel® IXP4XX Product Line of Network Processors.

Table 4. Signal Type Definitions

Symbol	Description
I	Input pin only
O	Output pin only
I/O	Pin can be either an input or output
OD	Open Drain pin
PWR	Power pin
GND	Ground pin
N/C	No Connect
-	Pin must be connected as described

This section’s other tables include:

- Table 5 — SDRAM Interface signals
- Table 6 — PCI Controller signals
- Table 7 — High-Speed, Serial Interface 0 signals
- Table 8 — High-Speed, Serial Interface 1 signals
- Table 9 — MII/RMII Interfaces signals
- Table 10 — UTOPIA-2 Interface signals

- Table 11 — Expansion Bus Interface signals
- Table 12 — UART Interfaces signals
- Table 13 — USB Interface signals
- Table 14 — Oscillator Interface signals
- Table 15 — GPIO Interface signals
- Table 16 — JTAG Interface signals
- Table 17 — System Interface signals
- Table 18 — Power Interface signals

Table 5. SDRAM Interface

Name	Type†	Description
SDM_ADDR[12:0]	O	SDRAM Address: A0-A12 signals are output during the READ/WRITE commands and ACTIVE commands to select a location in memory to act upon.
SDM_DATA[31:0]	I/O	SDRAM Data: Bidirectional data bus used to transfer data to and from the SDRAM
SDM_CLKOUT	O	SDRAM Clock: All SDRAM input signals are sampled on the rising edge of SDM_CLKOUT. All output signals are driven with respect to the rising edge of SDM_CLKOUT.
SDM_BA[1:0]	O	SDRAM Bank Address: SDM_BA0 and SDM_BA1 define the bank the current command is attempting to access.
SDM_RAS_N	O	SDRAM Row Address strobe/select (active low): Along with SDM_CAS_N, SDM_WE_N, and SDM_CS_N signals determines the current command to be executed.
SDM_CAS_N	O	SDRAM Column Address strobe/select (active low): Along with SDM_RAS_N, SDM_WE_N, and SDM_CS_N signals determines the current command to be executed.
SDM_CS_N[1:0]	O	SDRAM Chip select (active low): CS# enables the command decoder in the external SDRAM when logic low and disables the command decoder in the external SDRAM when logic high.
SDM_WE_N	O	SDRAM Write enable (active low): Along with SDM_CAS_N, SDM_RAS_N, and SDM_CS_N signals determines the current command to be executed.
SDM_CKE	O	SDRAM Clock Enable: CKE is driving high to activate the clock to an external SDRAM and driven low to de-activate the CLK to an external SDRAM.
SDM_DQM[3:0]	O	SDRAM Data bus mask: DQM is used to byte select data during read/write access to an external SDRAM.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 6. PCI Controller (Sheet 1 of 2)

Name	Type†	Description
PCI_AD[31:0]	I/O	PCI Address/Data bus used to transfer address and bidirectional data to and from multiple PCI devices. Should be pulled low with a 10-KΩ resistor, when the PCI bus is not being utilized in the system.
PCI_CBE_N[3:0]	I/O	PCI Command/Byte Enables is used as a command word during PCI address cycles and as byte enables for data cycles. Should be pulled high with a 10-KΩ resistor, when the PCI bus is not being utilized in the system.
PCI_PAR	I/O	PCI Parity used to check parity across the 32 bits of PCI_AD and the four bits of PCI_CBE_N. Should be pulled low with a 10-KΩ resistor, when the PCI bus is not being utilized in the system.
PCI_FRAME_N	I/O	PCI Cycle Frame used to signify the beginning and duration of a transaction. The signal will be inactive prior to or during the final data phase of a given transaction. Should be pulled high with a 10-KΩ resistor.
PCI_TRDY_N	I/O	PCI Target Ready informs that the target of the PCI bus is ready to complete the current data phase of a given transaction. Should be pulled high with a 10-KΩ resistor.
PCI_IRDY_N	I/O	PCI Initiator Ready informs the PCI bus that the initiator is ready to complete the transaction. Should be pulled high with a 10-KΩ resistor.
PCI_STOP_N	I/O	PCI Stop indicates that the current target is requesting the current initiator to stop the current transaction. Should be pulled high with a 10-KΩ resistor.
PCI_PERR_N	I/O	PCI Parity Error asserted when a PCI parity error is detected — between the PCI_PAR and associated information on the PCI_AD bus and PCI_CBE_N — during all PCI transactions, except for Special Cycles. The agent receiving data will drive this signal. Should be pulled high with a 10-KΩ resistor.
PCI_SERR_N	I/OD	PCI System Error asserted when a parity error occurs on special cycles or any other error that will cause the PCI bus not to function properly. This signal can function as an input or an open drain output. Should be pulled high with a 10-KΩ resistor.
PCI_DEVSEL_N	I/O	PCI Device Select: • When used as an output, PCI_DEVSEL_N indicates that device has decoded that address as the target of the requested transaction. • When used as an input, PCI_DEVSEL_N indicates if any device on the PCI bus exists with the given address. Should be pulled high with a 10-KΩ resistor.
PCI_IDSEL	I	PCI Initialization Device Select is a chip select during configuration reads and writes. Should be pulled low with a 10-KΩ resistor, when the PCI bus is not being utilized in the system.
PCI_REQ_N[3:1]	I	PCI arbitration request: Used by the internal PCI arbiter to allow an agent to request the PCI bus. Should be pulled high with a 10-KΩ resistor, when the PCI bus is not being utilized in the system.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 6. PCI Controller (Sheet 2 of 2)

Name	Type†	Description
PCI_REQ_N[0]	I/O	PCI arbitration request: • When configured as an input (PCI arbiter enabled), the internal PCI arbiter will allow an agent to request the PCI bus. • When configured as an output (PCI arbiter disabled), the pin will be used to request access to the PCI bus from an external arbiter. Should be pulled high with a 10-KΩ resistor, when the PCI bus is not being utilized in the system.
PCI_GNT_N[3:1]	O	PCI arbitration grant: Generated by the internal PCI arbiter to allow an agent to claim control of the PCI bus.
PCI_GNT_N[0]	I/O	PCI arbitration grant: • When configured as an output (PCI arbiter enabled), the internal PCI arbiter to allow an agent to claim control of the PCI bus. • When configured as an input (PCI arbiter disabled), the pin will be used to claim access of the PCI bus from an external arbiter. Should be pulled high with a 10-KΩ resistor, when the PCI bus is not being utilized in the system.
PCI_INTA_N	O/D	PCI interrupt: Used to request an interrupt. Should be pulled high with a 10-KΩ resistor.
PCI_CLKIN	I	PCI Clock: lock provides timing for all transactions on PCI. All PCI signals — except INTA#, INTB#, INTC#, and INTD# — are sampled on the rising edge of CLK and timing parameters are defined with respect to this edge. The PCI clock rate can operate at up to 66 MHz. Should be pulled low with a 10-KΩ resistor, when the PCI bus is not being utilized in the system.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 7. High-Speed, Serial Interface 0

Name	Type†	Description
HSS_TXFRAME0	I/O	The High-Speed Serial (HSS) transmit frame signal can be configured as an input or an output to allow an external source become synchronized with the transmitted data. Often known as a Frame Sync signal. Configured as an input upon reset.
HSS_TXDATA0	O	Transmit data out. Open Drain output. Must be pulled up with a 10-KΩ resistor to V_{CCP}.
HSS_TXCLK0	I/O	The High-Speed Serial (HSS) transmit clock signal can be configured as an input or an output. The clock can be a frequency ranging from 512 KHz to 8.192 MHz. Used to clock out the transmitted data. Configured as an input upon reset. Frame sync and data can be selected to be generated on the rising or falling edge of the transmit clock.
HSS_RXFRAME0	I/O	The High-Speed Serial (HSS) receive frame signal can be configured as an input or an output to allow an external source to become synchronized with the received data. Often known as a Frame Sync signal. Configured as an input upon reset.
HSS_RXDATA0	I	Receive data input. Can be sampled on the rising or falling edge of the receive clock. Should be tied low through a 10-KΩ resistor, when not being utilized in the system.
HSS_RXCLK0	I/O	The High-Speed Serial (HSS) receive clock signal can be configured as an input or an output. The clock can be from 512 KHz to 8.192 MHz. Used to sample the received data. Configured as an input upon reset.

Table 8. High-Speed, Serial Interface 1

Name	Type †	Description
HSS_TXFRAME1	I/O	The High-Speed Serial (HSS) transmit frame signal can be configured as an input or an output to allow an external source to be synchronized with the transmitted data. Often known as a Frame Sync signal. Configured as an input upon reset.
HSS_TXDATA1	O	Transmit data out. Open Drain output. Must be pulled up with a 10-KΩ resistor to V_{CCP}.
HSS_TXCLK1	I/O	The High-Speed Serial (HSS) transmit clock signal can be configured as an input or an output. The clock can be a frequency ranging from 512 KHz to 8.192 MHz. Used to clock out the transmitted data. Configured as an input upon reset. Frame sync and Data can be selected to be generated on the rising or falling edge of the transmit clock.
HSS_RXFRAME1	I/O	The High-Speed Serial (HSS) receive frame signal can be configured as an input or an output to allow an external source to be synchronized with the received data. Often known as a Frame Sync signal. Configured as an input upon reset.
HSS_RXDATA1	I	Receive data input. Can be sampled on the rising or falling edge of the receive clock. Should be tied low through a 10-KΩ resistor when not being utilized in the system.
HSS_RXCLK1	I/O	The High-Speed Serial (HSS) receive clock signal can be configured as an input or an output. The clock can be from 512 KHz to 8.192 MHz. Used to sample the received data. Configured as an input upon reset.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 9. MII/RMII Interfaces (Sheet 1 of 3)

Name	Type†	Description
ETH_TXCLK0	I	Externally supplied transmit clock. • 25 MHz for 100 Mbps operation • 2.5 MHz for 10 Mbps • 50 MHz for RMII 100 Mbps operation, originating from the physical interface This MAC interface does not contain hardware hashing capabilities. Should be pulled low through a 10-KΩ resistor, when not connected.
ETH_TXDATA0[3:0]	O	Transmit data bus to PHY, asserted synchronously with respect to ETH_TXCLK0. This MAC interface does not contain hardware hashing capabilities. When configured to RMII mode of operation, ETH_TXDATA0[1:0] are used to transmit the data. ETH_TXDATA0[3:2] are not connected.
ETH_TXEN0	O	Indicates that the PHY is being presented with nibbles on the MII interface. Asserted synchronously, with respect to ETH_TXCLK0, at the first nibble of the preamble and remains asserted until all the nibbles of a frame are presented. This MAC interface does not contain hardware hashing capabilities.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 9. MII/RMII Interfaces (Sheet 2 of 3)

Name	Type†	Description
ETH_RXCLK0	I	Externally supplied receive clock. • 25 MHz for 100 Mbps operation • 2.5 MHz for 10 Mbps • 50 MHz for RMII 100 Mbps operation originating from the physical interface. This MAC interface does not contain hardware hashing capabilities. Should be pulled low through a 10-KΩ resistor, when not connected.
ETH_RXDATA0[3:0]	I	Receive data bus from PHY, data sampled synchronously with respect to ETH_RXCLK0. This MAC interface does not contain hardware hashing capabilities. • When configured to RMII mode of operation, ETH_RXDATA0[1:0] are used to receive data from the PHY. ETH_RXDATA0[3:2] are not connected and should be pulled low through a 10-KΩ resistor. • Should be pulled low through a 10-KΩ resistor, when not connected.
ETH_RXDV0	I	Receive data valid, used to inform the MII interface that the Ethernet PHY is sending data. This MAC interface does not contain hardware hashing capabilities. Should be pulled low through a 10-KΩ resistor, when not connected.
ETH_COL0	I	Asserted by the PHY when a collision is detected by the PHY. This MAC interface does not contain hardware hashing capabilities. • When configured in RMII mode of operation, ETH_COL0 must be pulled low through a 10-KΩ resistor. • Should be pulled low through a 10-KΩ resistor, when not connected.
ETH_CRS0	I	Asserted by the PHY when the transmit medium or receive medium is active. De-asserted when both the transmit and receive medium are idle. Remains asserted throughout the duration of a collision condition. PHY asserts CRS asynchronously and de-asserts synchronously, with respect to ETH_RXCLK0. This MAC interface does not contain hardware hashing capabilities. Should be pulled low through a 10-KΩ resistor, when not connected.
ETH_MDIO	I/O	Management data output. Provides the write data to both PHY devices connected to each MII interface. Should be pulled low through a 10-KΩ resistor, when not connected.
ETH_MDC	O	Management data clock. Management data interface clock is used to clock the MDIO signal as an output and sample the MDIO as an input. The ETH_MDC is an input on power up and can be configured to be an output through an Intel® API as documented in the *Intel® IXP4XX Product Line Programmer's Guide (Version 1.1)*.
ETH_TXCLK1	I	Externally supplied transmit clock. • 25 MHz for 100 Mbps operation • 2.5 MHz for 10 Mbps • 50 MHz for RMII 100 Mbps operation originating from the physical interface This MAC contains hardware hashing capabilities. Should be pulled low through a 10-KΩ resistor when not connected.
ETH_TXDATA1[3:0]	O	Transmit data bus to PHY, asserted synchronously with respect to ETH_TXCLK1. This MAC contains hardware hashing capabilities. When configured to RMII mode of operation, ETH_TXDATA1[1:0] are used to transmit the data. ETH_TXDATA1[3:2] are not connected.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 9. MII/RMII Interfaces (Sheet 3 of 3)

Name	Type†	Description
ETH_TXEN1	O	Indicates that the PHY is being presented with nibbles on the MII interface. Asserted synchronously, with respect to ETH_TXCLK1, at the first nibble of the preamble, and remains asserted until all the nibbles of a frame are presented. This MAC contains hardware hashing capabilities.
ETH_RXCLK1	I	Externally supplied receive clock. • 25 MHz for 100 Mbps operation • 2.5 MHz for 10 Mbps • 50 MHz for RMII 100 Mbps operation originating from the physical interface This MAC contains hardware hashing capabilities. Should be pulled low through a 10-KΩ resistor when not connected.
ETH_RXDATA1[3:0]	I	Receive data bus from PHY, data sampled synchronously, with respect to ETH_RXCLK1. This MAC contains hardware hashing capabilities. • When configured to RMII mode of operation, ETH_RXDATA1[1:0] are used to receive data from the PHY. ETH_RXDATA1[3:2] are not connected and should be pulled low through a 10-KΩ resistor. • Should be pulled low through a 10-KΩ resistor, when not connected.
ETH_RXDV1	I	Receive data valid, used to inform the MII interface that the Ethernet PHY is sending data. This MAC contains hardware hashing capabilities. Should be pulled low through a 10-KΩ resistor when not connected.
ETH_COL1	I	Asserted by the PHY when a collision is detected by the PHY. This MAC contains hardware hashing capabilities. • When configured in RMII mode of operation, ETH_COL1 must be pulled low through a 10-KΩ resistor. • Should be pulled low through a 10-KΩ resistor, when not connected.
ETH_CRS1	I	Asserted by the PHY when the transmit medium or receive medium are active. De-asserted when both the transmit and receive medium are idle. Remains asserted throughout the duration of collision condition. PHY asserts CRS asynchronously and de-asserts synchronously with respect to ETH_RXCLK1. This MAC contains hardware hashing capabilities. Should be pulled low through a 10-KΩ resistor when not connected.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 10. UTOPIA-2 Interface (Sheet 1 of 2)

Name	Type†	Description
UTP_OP_CLK	I	UTOPIA Transmit clock input. Also known as UTP_TX_CLK. This signal is used to synchronize all UTOPIA-transmit outputs to the rising edge of the UTP_OP_CLK. This signal should be tied low through a 10-KΩ resistor, if not being used.
UTP_OP_FCO	O	UTOPIA flow control output signal. Also known as the TXENB_N signal. Used to inform the selected PHY that data is being transmitted to the PHY. Placing the PHY's address on the UTP_OP_ADDR — and bringing UTP_OP_FCO to logic 1, during the current clock — followed by the UTP_OP_FCO going to a logic 0, on the next clock cycle, selects which PHY is active in MPHY mode. In SPHY configurations, UTP_OP_FCO is used to inform the PHY that the Intel® IXP4XX Product Line processor is ready to send data.
UTP_OP_SOC	O	Start of Cell. Also known as TX_SOC. Active high signal is asserted when UTP_OP_DATA contains the first valid byte of a transmitted cell.
UTP_OP_DATA[7:0]	O	UTOPIA output data. Also known as UTP_TX_DATA. Used to send data from the Intel® IXP4XX Product Line processor to an ATM UTOPIA-Level-2-compliant PHY.
UTP_OP_ADDR[4:0]	O	Transmit PHY address bus. Used by the Intel® IXP4XX Product Line processor when operating in MPHY mode to poll and select a single PHY at any given time.
UTP_OP_FCI	I	UTOPIA Output data flow control input: Also known as the TXFULL/CLAV signal. Used to inform the Intel® IXP4XX Product Line processor of the ability of each polled PHY to receive a complete cell. For cell-level flow control in an MPHY environment, TxClav is an active high tri-stateable signal from the MPHY to ATM layer. The UTP_OP_FCI, which is connected to multiple MPHY devices, will see logic high generated by the PHY, one clock after the given PHY address is asserted — when a full cell can be received by the PHY. The UTP_OP_FCI will see a logic low generated by the PHY one clock cycle, after the PHY address is asserted — if a full cell cannot be received by the PHY. This signal should be tied low through a 10-KΩ resistor if not being used.
UTP_IP_CLK	I	UTOPIA Receive clock input. Also known as UTP_RX_CLK. This signal is used to synchronize all UTOPIA-received inputs to the rising edge of the UTP_IP_CLK. This signal should be tied low through a 10-KΩ resistor, if not being used.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 10. UTOPIA-2 Interface (Sheet 2 of 2)

Name	Type†	Description
UTP_IP_FCI	I	UTOPIA Input Data flow control input signal. Also known as RXEMPTY/CLAV. Used to inform the Intel® IXP4XX Product Line processor of the ability of each polled PHY to send a complete cell. For cell-level flow control in an MPHY environment, RxClav is an active high tri-stateable signal from the MPHY to ATM layer. The UTP_IP_FCI, which is connected to multiple MPHY devices, will see logic high generated by the PHY, one clock after the given PHY address is asserted, when a full cell can be received by the PHY. The UTP_IP_FCI will see a logic low generated by the PHY, one clock cycle after the PHY address is asserted if a full cell cannot be received by the PHY. In SPHY mode, this signal is used to indicate to the Intel® IXP4XX Product Line processor that the PHY has an octet or cell available to be transferred to the Intel® IXP4XX Product Line processor. Should be tied low through a 10-KΩ resistor when not being utilized in the system.
UTP_IP_SOC	I	Start of Cell. RX_SOC Active-high signal that is asserted when UTP_IP_DATA contains the first valid byte of a transmitted cell. Should be tied low through a 10-KΩ resistor, when not being utilized in the system.
UTP_IP_DATA[7:0]	I	UTOPIA input data. Also known as RX_DATA. Used by to the Intel® IXP4XX Product Line processor to receive data from an ATM UTOPIA-Level-2-compliant PHY. Should be tied low through a 10-KΩ resistor, when not being utilized in the system.
UTP_IP_ADDR[4:0]	O	Receive PHY address bus. Used by the Intel® IXP4XX Product Line processor when operating in MPHY mode to poll and select a single PHY at any one given time.
UTP_IP_FCO	O	UTOPIA Input Data Flow Control Output signal: Also known as the RX_ENB_N. In SPHY configurations, UTP_IP_FCO is used to inform the PHY that the Intel® IXP4XX Product Line processor is ready to accept data. In MPHY configurations, UTP_IP_FCO is used to select which PHY will drive the UTP_RX_DATA and UTP_RX_SOC signals. The PHY is selected by placing the PHY's address on the UTP_IP_ADDR and bringing UTP_OP_FCO to logic 1 during the current clock, followed by the UTP_OP_FCO going to a logic 0 on the next clock cycle.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 11. Expansion Bus Interface

Name	Type†	Description
EX_CLK	I	Input clock signal used to sample all expansion interface inputs and clock all expansion interface outputs.
EX_ALE	O	Address-latch enable used for multiplexed address/data bus accesses. Used in Intel and Motorola multiplexed modes of operation.
EX_ADDR[23:0]	I/O	Expansion-bus address used as an output for data accesses over the expansion bus. Also, used as an input during reset to capture device configuration. These signals have a weak pull-up resistor attached internally. Based on the desired configuration, various address signals must be tied low in order for the device to operate in the desired mode. (See the *Intel® IXP4XX Product Line of Network Processors Developer's Manual* for details.)
EX_WR_N	O	Intel-mode write strobe / Motorola-mode data strobe (EXP_MOT_DS_N) / TI mode data strobe (TI_HDS1_N).
EX_RD_N	O	Intel-mode read strobe / Motorola-mode read-not-write (EXPB_MOT_RNW) / TI mode read-not-write (TI_HR_W_N).
EX_CS_N[7:0]	O	External chip selects for expansion bus. • Chip selects 0 through 7 can be configured to support Intel or Motorola bus cycles. • Chip selects 4 through 7 can be configured to support TI HPI bus cycles.
EX_DATA[15:0]	I/O	Expansion-bus, bidirectional data
EX_IOWAIT_N	I	Data ready/acknowledge from expansion-bus devices. Expansion-bus access is halted when an external device sets EX_IOWAIT_N to logic 0 and resume from the halted location once the external device sets EX_IOWAIT_N to logic 1. This signal affects accesses that use EX_CS_N[7:0] when the chip select is configured in Intel- or Motorola-mode of operation. Should be pulled high through a 10-KΩ resistor, when not connected.
EX_RDY[3:0]	I	HPI interface ready signals. Can be configured to be active high or active low. These signals are used to halt accesses using chip selects 7 through 4 when the chip selects are configured to operate in HPI mode. There is one RDY signal per chip select. This signal only affects accesses that use EX_CS_N[7:4]. Should be tied low though a 10-KΩ resistor when not being used in the system.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 12. UART Interfaces (Sheet 1 of 2)

Name	Type†	Description
RXDATA0	I	UART serial data input to High-Speed UART Pins. Should be tied low through a 10-KΩ resistor, if not being used.
TXDATA0	O	UART serial data output. The TXD signal is set to the MARKING (logic 1) state upon a reset operation. High-Speed Serial UART Pins.
CTS0_N	I	UART CLEAR-TO-SEND input to High-Speed UART Pins. When logic 0, this pin indicates that the modem or data set connected to the UART interface of the Intel® IXP4XX Product Line processor is ready to exchange data. The CTS_N signal is a modem status input whose condition can be tested by the processor. Should be tied high through a 10-KΩ resistor if not being used.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 12. UART Interfaces (Sheet 2 of 2)

Name	Type†	Description
RTS0_N	O	UART REQUEST-TO-SEND output: When logic 0, this informs the modem or the data set connected to the UART interface of the Intel® IXP4XX Product Line processor that the UART is ready to exchange data. A reset sets the request to send signal to logic 1. LOOP-mode operation holds this signal in its inactive state (logic 1). High-Speed UART Pins.
RXDATA1	I	UART serial data input. Should be tied low through a 10-KΩ resistor if not being used.
TXDATA1	O	UART serial data output. The TXD signal is set to the MARKING (logic 1) state upon a Reset operation. Console UART Pins.
CTS1_N	I	UART CLEAR-TO-SEND input to Console UART pins. When logic 0, this pin indicates that the modem or data set connected to the UART interface of the Intel® IXP4XX Product Line processor is ready to exchange data. The CTS_N signal is a modem status input whose condition can be tested by the processor. Should be tied high through a 10-KΩ resistor if not being used.
RTS1_N	O	UART REQUEST-TO-SEND output: When logic 0, this informs the modem or the data set connected to the UART interface of the Intel® IXP4XX Product Line processor that the UART is ready to exchange data. A reset sets the request to send signal to logic 1. LOOP-mode operation holds this signal in its inactive state (logic 1). Console UART Pins.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 13. USB Interface

Name	Type†	Description
USB_DPOS	I/O	Positive signal of the differential USB receiver/driver.
USB_DNEG	I/O	Negative signal of the differential USB receiver/driver.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 14. Oscillator Interface

Name	Type†	Description
OSC_IN	I	33.33-MHz, sinusoidal crystal input signal. Can be driven by an oscillator.
OSC_OUT	O	33.33-MHz, sinusoidal crystal output signal. Left disconnected when being driven by an oscillator.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 15. GPIO Interface

Name	Type†	Description
GPIO(12:0)	I/O	General purpose Input/Output pins. May be configured as an input or an output. As an input, each signal may be configured a processor interrupt. Default after reset is to be configured as inputs. Should be tied low using a 10-KΩ resistor, when not being used in the system.
GPIO(13)	I/O	General purpose input/output pins. May be configured as an input or an output. Default after reset is to be configured as inputs. Should be tied low using a 10-KΩ resistor, when not being used in the system.
GPIO(14)	I/O	Can be configured the same as GPIO Pin 13 or as a clock output. Configuration as an output clock can be set at various speeds of up to 33 MHz with various duty cycles. Configured as an input, upon reset. Should be tied low though a 10-KΩ resistor, when not being used in the system.
GPIO(15)	I/O	Can be configured the same as GPIO Pin 13 or as a clock output. Configuration as an output clock can be set at various speeds of up to 33 MHz with various duty cycles. Configured as an output, upon reset. Can be used to clock the expansion interface, after reset. Should be tied low though a 10-KΩ resistor, when not being used in the system.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 16. JTAG Interface

Name	Type†	Description
JTG_TMS	I	Test mode select for the IEEE 1149.1 JTAG interface.
JTG_TDI	I	Input data for the IEEE 1149.1 JTAG interface.
JTG_TDO	O	Output data for the IEEE 1149.1 JTAG interface.
JTG_TRST_N	I	Used to reset the IEEE 1149.1 JTAG interface. When the JTAG interface is not being used, the signal should be tied low using a 10-KΩ resistor.
JTG_TCK	I	Used as the clock for the IEEE 1149.1 JTAG interface.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 17. System Interface (Sheet 1 of 2)

Name	Type†	Description
BYPASS_CLK	I	Used for test purposes only. Must be pulled high for normal operation.
SCANTESTMODE_N	I	Used for test purposes only. Must be pulled high for normal operation.
RESET_IN_N	I	Used as a reset input to the device after power up conditions have been met. Power up conditions include the power supplies reaching a safe stable condition and the PLL achieving a locked state and the PWRON_RESET_N coming to an active state prior to the RESET_IN_N coming to an active state.

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 17. System Interface (Sheet 2 of 2)

Name	Type†	Description
PWRON_RESET_N	I	Signal used at power up to reset all internal logic to a known state after the PLL has achieved a locked state. The PWRON_RESET_N signal is a 1.3V signal.
HIGHZ_N	I	Used for test purposes only. Must be pulled high for normal operation.
PLL_LOCK	O	Signal used to inform external reset logic that the internal PLL has achieved a locked state.
RCOMP	I	Signal used to control PCI drive strength characteristics. Drive strength is varied on PCI address, data and control signals. Pin requires a 34-Ω +/- 1% tolerance resistor to ground. Refer to Figure 12 on page 77

† For a legend of the **Type** codes, see Table 4 on page 30.

Table 18. Power Interface

Name	Type†	Description
VCC	I	1.3-V power supply input pins used for the internal logic of the Intel® IXP4XX Product Line processor.
VCCP	I	3.3-V power supply input pins used for the peripheral (I/O) logic of the Intel® IXP4XX Product Line processor.
VSS		Ground power supply input pins used for both the 3.3-V and the 1.3-V power supplies.
VCCOSCP	I	3.3-V power supply input pins used for the peripheral (I/O) logic of the analog oscillator circuitry on the Intel® IXP4XX Product Line processor. Require special power filtering circuitry. Refer to Figure 10 on page 76
VSSOSCP	I	Ground input pins used for the peripheral (I/O) logic of the analog oscillator circuitry on the Intel® IXP4XX Product Line processor. Used in conjunction with the VCCOSCP pins. Requires special power filtering circuitry. Refer to Figure 10 on page 76
VCCOSC	I	1.3-V power supply input pins used for the internal logic of the analog oscillator circuitry on the Intel® IXP4XX Product Line processor. Requires special power filtering circuitry. Refer to Figure 11 on page 76
VSSOSC	I	Ground power supply input pins used for the internal logic of the analog oscillator circuitry on the Intel® IXP4XX Product Line processor. Used in conjunction with the VCCOSC pins. Requires special power filtering circuitry. Refer to Figure 11 on page 76
VCCPLL1	I	1.3-V power supply input pins used for the internal logic of the analog phase lock-loop circuitry on the Intel® IXP4XX Product Line processor. Requires special power filtering circuitry.Refer to Figure 8 on page 75
VCCPLL2	I	1.3-V power supply input pins used for the internal logic of the analog phase lock-loop circuitry on the Intel® IXP4XX Product Line processor. Requires special power filtering circuitry.Refer to Figure 9 on page 75

† For a legend of the **Type** codes, see Table 4 on page 30.

4.2 Package Description

Figure 6. 492-Pin Lead PBGA Package

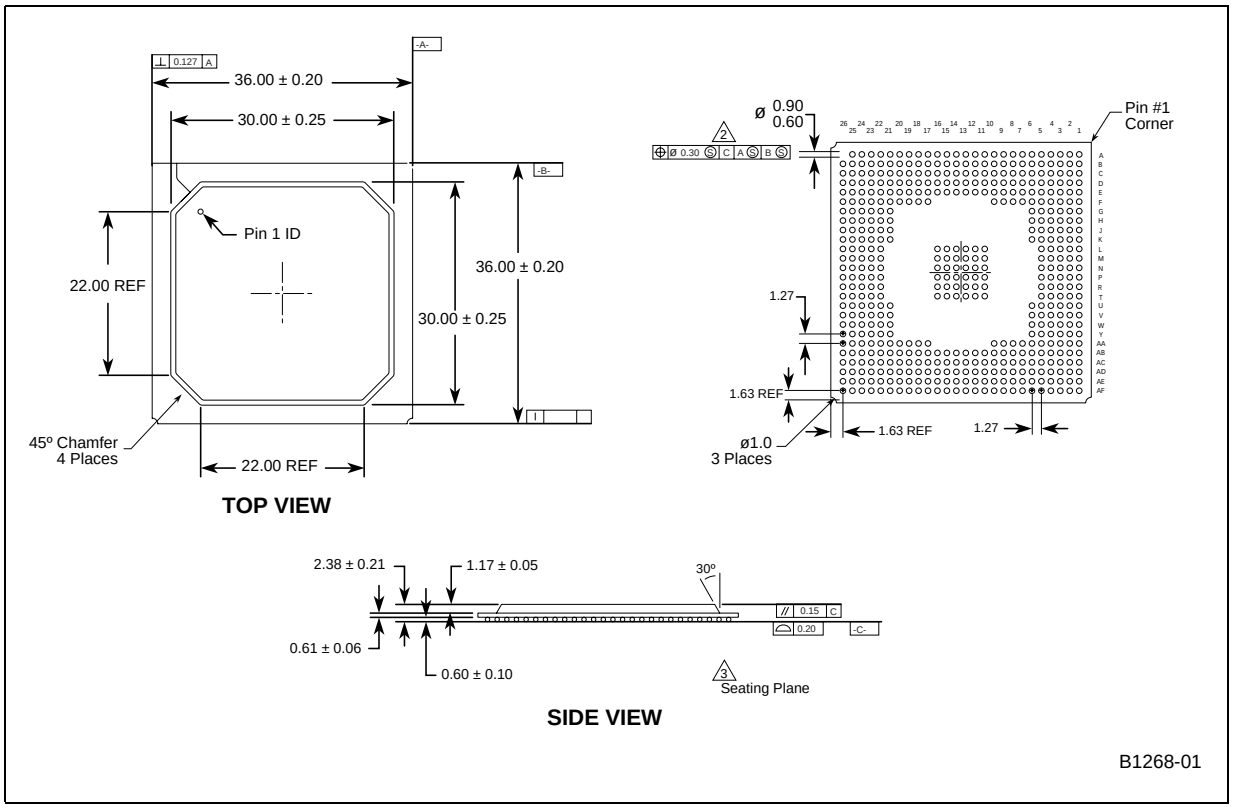
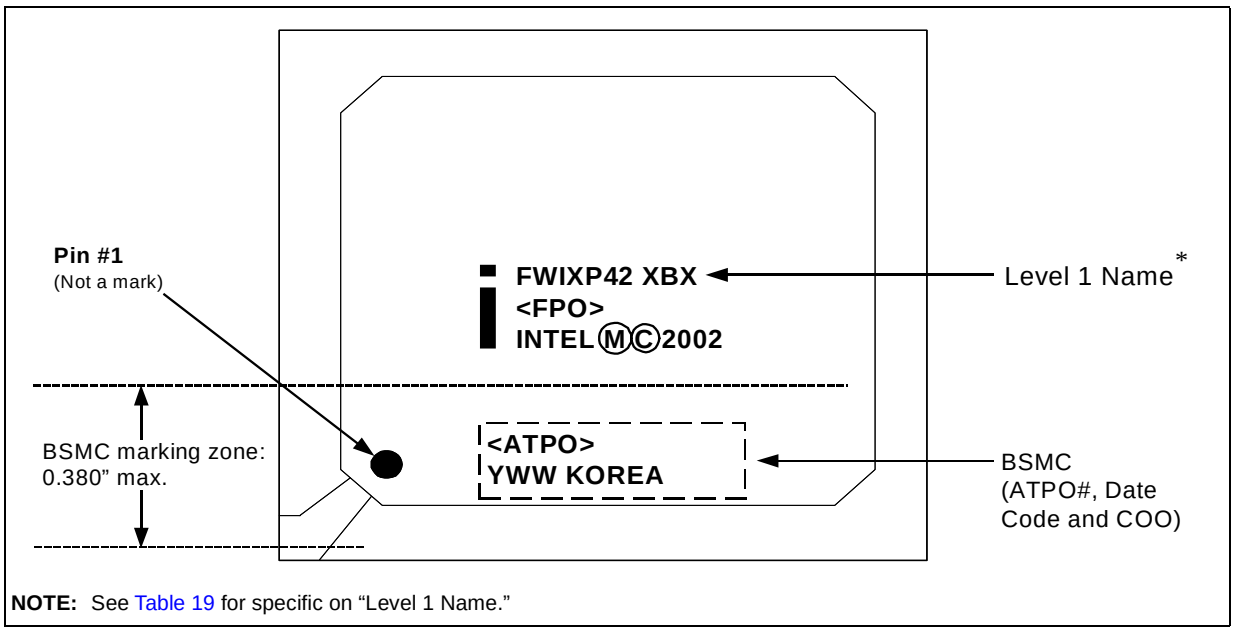


Figure 7. Package Markings



NOTE: See Table 19 for specific on "Level 1 Name."

Table 19. 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® IXP422	B-0	266	FWIXP422BB
Intel® IXP421	B-0	266	FWIXP421BB
Intel® IXP420	B-0	266	FWIXP420BB

4.3 Signal-Pin Descriptions

Separate ball-map-assignment tables are given for each model of the Intel® IXP4XX Product Line of Network Processors. Those tables include:

Device	Table #	Starting Page
Intel® IXP425	20	45
Intel® IXP422	21	52
Intel® IXP421	22	59
Intel® IXP420	23	66

Table 20. Ball Map Assignment for Intel® IXP425 Network Processor (Sheet 1 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
A1	PCI_AD[27]	B1	PCI_AD[28]	C1	PCI_AD[26]	D1	PCI_AD[25]
A2	PCI_GNT_N[1]	B2	VCCP	C2	PCI_AD[30]	D2	VSS
A3	PCI_GNT_N[3]	B3	PCI_GNT_N[2]	C3	VSS	D3	PCI_AD[31]
A4	SDM_DATA[19]	B4	VCCP	C4	PCI_INTA_N	D4	VCC
A5	SDM_DATA[27]	B5	SDM_DATA[28]	C5	VSS	D5	PCI_SERR_N
A6	SDM_DATA[26]	B6	VCCP	C6	SDM_DATA[18]	D6	VCC
A7	SDM_DATA[25]	B7	SDM_DATA[21]	C7	VSS	D7	SDM_DATA[29]
A8	SDM_DATA[23]	B8	VSS	C8	VCCP	D8	SDM_DATA[20]
A9	SDM_DATA[14]	B9	SDM_DATA[0]	C9	SDM_DATA[24]	D9	VCC
A10	SDM_DATA[13]	B10	VCCP	C10	VSS	D10	SDM_DATA[15]
A11	SDM_DATA[11]	B11	SDM_DATA[12]	C11	SDM_DATA[2]	D11	SDM_DATA[1]
A12	SDM_DATA[10]	B12	VSS	C12	SDM_DATA[4]	D12	VCC
A13	SDM_DATA[6]	B13	SDM_DATA[9]	C13	VSS	D13	SDM_DATA[5]
A14	SDM_DATA[8]	B14	VCCP	C14	SDM_DATA[7]	D14	VCC
A15	SDM_DQM[1]	B15	SDM_DQM[2]	C15	SDM_DQM[3]	D15	SDM_WE_N
A16	SDM_CS_N[0]	B16	VSS	C16	VCCP	D16	SDM_CS_N[1]
A17	SDM_CLKOUT	B17	SDM_CKE	C17	SDM_CAS_N	D17	SDM_BA[1]
A18	SDM_RAS_N	B18	VCCP	C18	SDM_ADDR[11]	D18	VCC
A19	SDM_ADDR[12]	B19	SDM_ADDR[10]	C19	VSS	D19	SDM_ADDR[0]
A20	SDM_ADDR[9]	B20	VSS	C20	SDM_ADDR[6]	D20	VSS
A21	SDM_ADDR[8]	B21	SDM_ADDR[1]	C21	SDM_ADDR[2]	D21	VCC
A22	SDM_ADDR[5]	B22	VCCP	C22	VSS	D22	EX_ALE
A23	EX_RD_N	B23	EX_IOWAIT_N	C23	EX_ADDR[0]	D23	VCC
A24	EX_ADDR[1]	B24	VSS	C24	EX_ADDR[4]	D24	EX_ADDR[6]
A25	EX_ADDR[3]	B25	VCCP	C25	EX_ADDR[7]	D25	RCOMP
A26	EX_ADDR[5]	B26	EX_ADDR[9]	C26	EX_ADDR[13]	D26	EX_ADDR[17]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, "Functional Signal Definitions" on page 30.

Table 20. Ball Map Assignment for Intel® IXP425 Network Processor (Sheet 2 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
E1	PCI_AD[23]	F1	PCI_AD[20]	G1	PCI_AD[21]	H1	PCI_AD[16]
E2	VCCP	F2	PCI_IDSEL	G2	VCCP	H2	PCI_AD[18]
E3	PCI_REQ_N[2]	F3	VCC	G3	PCI_AD[24]	H3	VCC
E4	VSS	F4	PCI_REQ_N[0]	G4	VSS	H4	PCI_CBE_N[3]
E5	PCI_GNT_N[0]	F5	VCCP	G5	PCI_REQ_N[1]	H5	VCC
E6	SDM_DATA[16]	F6	VCC	G6	VSS	H6	PCI_REQ_N[3]
E7	VCCP	F7	SDM_DATA[31]				
E8	SDM_DATA[30]	F8	VSS				
E9	VSS	F9	SDM_DATA[17]				
E10	SDM_DATA[22]	F10	VCC				
E11	VCCP						
E12	SDM_DATA[3]						
E13	VSS						
E14	SDM_DQM[0]						
E15	VCCP						
E16	SDM_BA[0]						
E17	VSS	F17	VCC				
E18	SDM_ADDR[7]	F18	SDM_ADDR[4]				
E19	VCCP	F19	VSS				
E20	SDM_ADDR[3]	F20	USB_DPOS				
E21	USB_DNEG	F21	VCC	G21	EX_ADDR[2]	H21	VSS
E22	VCCP	F22	EX_WR_N	G22	VSS	H22	EX_ADDR[11]
E23	VSS	F23	VCC	G23	EX_ADDR[12]	H23	EX_ADDR[18]
E24	EX_ADDR[10]	F24	EX_ADDR[14]	G24	VSS	H24	VCCP
E25	EX_ADDR[15]	F25	VCCP	G25	EX_ADDR[20]	H25	VSS
E26	EX_ADDR[19]	F26	EX_ADDR[21]	G26	EX_ADDR[22]	H26	EX_CS_N[1]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 20. Ball Map Assignment for Intel® IXP425 Network Processor (Sheet 3 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
J1	PCI_CLKIN	K1	PCI_CBE_N[2]	L1	PCI_DEVSEL_N	M1	PCI_CBE_N[1]
J2	VCCP	K2	VSS	L2	VCCP	M2	PCI_PAR
J3	VSS	K3	PCI_AD[17]	L3	PCI_STOP_N	M3	VSS
J4	PCI_AD[22]	K4	VCCP	L4	VCC	M4	PCI_IRDY_N
J5	VSS	K5	PCI_AD[19]	L5	PCI_FRAME_N	M5	VCCP
J6	PCI_AD[29]	K6	VCC				
				L11	VSS	M11	VSS
				L12	VSS	M12	VSS
				L13	VSS	M13	VSS
				L14	VSS	M14	VSS
				L15	VSS	M15	VSS
				L16	VSS	M16	VSS
J21	EX_ADDR[8]	K21	VCC				
J22	EX_ADDR[16]	K22	VSS	L22	VCCP	M22	EX_CS_N[5]
J23	VCC	K23	EX_CS_N[0]	L23	VCC	M23	EX_CLK
J24	EX_ADDR[23]	K24	EX_CS_N[3]	L24	EX_CS_N[6]	M24	EX_DATA[2]
J25	EX_CS_N[2]	K25	VCCP	L25	EX_DATA[0]	M25	VSS
J26	EX_CS_N[4]	K26	EX_CS_N[7]	L26	EX_DATA[1]	M26	EX_DATA[3]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 20. Ball Map Assignment for Intel® IXP425 Network Processor (Sheet 4 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
N1	PCI_AD[11]	P1	PCI_CBE_N[0]	R1	PCI_AD[10]	T1	PCI_AD[6]
N2	VCCP	P2	PCI_AD[14]	R2	VSS	T2	PCI_TRDY_N
N3	VCC	P3	PCI_AD[13]	R3	PCI_AD[9]	T3	VSS
N4	PCI_PERR_N	P4	VSS	R4	VCC	T4	PCI_AD[2]
N5	PCI_AD[15]	P5	PCI_AD[12]	R5	PCI_AD[4]	T5	VCCP
N11	VSS	P11	VSS	R11	VSS	T11	VSS
N12	VSS	P12	VSS	R12	VSS	T12	VSS
N13	VSS	P13	VSS	R13	VSS	T13	VSS
N14	VSS	P14	VSS	R14	VSS	T14	VSS
N15	VSS	P15	VSS	R15	VSS	T15	VSS
N16	VSS	P16	VSS	R16	VSS	T16	VSS
N22	VCC	P22	EX_DATA[6]	R22	VCCP	T22	EX_RDY_N[0]
N23	VSS	P23	EX_DATA[7]	R23	VCC	T23	VSS
N24	VCC	P24	EX_DATA[8]	R24	EX_DATA[12]	T24	EX_DATA[14]
N25	EX_DATA[4]	P25	VCCP	R25	EX_DATA[11]	T25	VSS
N26	EX_DATA[5]	P26	EX_DATA[9]	R26	EX_DATA[10]	T26	EX_DATA[13]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 20. Ball Map Assignment for Intel® IXP425 Network Processor (Sheet 5 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
U1	PCI_AD[8]	V1	PCI_AD[5]	W1	PCI_AD[1]	Y1	HSS_TXCLK0
U2	VCCP	V2	VSS	W2	VCCP	Y2	HSS_RXCLK0
U3	PCI_AD[0]	V3	PCI_AD[3]	W3	HSS_RXFRAME0	Y3	HSS_TXFRAME1
U4	PCI_AD[7]	V4	VCC	W4	VSS	Y4	VCC
U5	HSS_TXDATA0	V5	HSS_TXFRAME0	W5	HSS_TXCLK1	Y5	VCCP
U6	VCC	V6	VSS	W6	HSS_RXFRAME1	Y6	ETH_TXEN0
U21	VCC	V21	GPIO[6]	W21	GPIO[1]	Y21	RXDATA1
U22	GPIO[14]	V22	GPIO[9]	W22	VCCP	Y22	GPIO[0]
U23	EX_RDY_N[1]	V23	VCC	W23	GPIO[8]	Y23	VCC
U24	EX_RDY_N[2]	V24	GPIO[13]	W24	VSS	Y24	GPIO[5]
U25	GPIO[15]	V25	VCCP	W25	GPIO[11]	Y25	VCCP
U26	EX_DATA[15]	V26	EX_RDY_N[3]	W26	GPIO[12]	Y26	GPIO[10]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 20. Ball Map Assignment for Intel® IXP425 Network Processor (Sheet 6 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
AA1	HSS_RXDATA0	AB1	HSS_TXDATA1	AC1	VSS	AD1	ETH_TXCLK0
AA2	VCCP	AB2	HSS_RXDATA1	AC2	ETH_TXDATA0[0]	AD2	ETH_RXDV0
AA3	VSS	AB3	ETH_TXDATA0[3]	AC3	VCCP	AD3	VSS
AA4	HSS_RXCLK1	AB4	ETH_TXDATA0[1]	AC4	VCC	AD4	ETH_CRS0
AA5	ETH_TXDATA0[2]	AB5	VSS	AC5	ETH_RXDATA0[0]	AD5	ETH_MDC
AA6	VCC	AB6	ETH_RXCLK0	AC6	VSS	AD6	ETH_TXDATA1[0]
AA7	ETH_RXDATA0[1]	AB7	VCCP	AC7	VCC	AD7	ETH_RXDATA1[3]
AA8	VSS	AB8	ETH_TXDATA1[2]	AC8	ETH_RXDATA1[2]	AD8	ETH_RXCLK1
AA9	ETH_TXDATA1[1]	AB9	ETH_RXDATA1[1]	AC9	VCC	AD9	VSS
AA10	VCC	AB10	VCCP	AC10	VCC	AD10	VSSOSCP
		AB11	VCCP	AC11	VCCOSCP	AD11	VCCP
		AB12	VSS	AC12	VCC	AD12	PLL_LOCK
		AB13	UTP_OP_DATA[7]	AC13	RESET_IN_N	AD13	PWRON_RESET_N
		AB14	VCCP	AC14	VCC	AD14	UTP_OP_DATA[4]
		AB15	UTP_OP_SOC	AC15	UTP_OP_DATA[1]	AD15	UTP_OP_DATA[2]
		AB16	VSS	AC16	UTP_OP_FCI	AD16	VSS
AA17	VCC	AB17	UTP_IP_DATA[6]	AC17	UTP_OP_ADDR[1]	AD17	UTP_OP_ADDR[3]
AA18	UTP_IP_FCI	AB18	VCCP	AC18	VCC	AD18	UTP_IP_DATA[7]
AA19	UTP_IP_ADDR[0]	AB19	UTP_IP_CLK	AC19	UTP_IP_DATA[2]	AD19	VCCP
AA20	VSS	AB20	UTP_IP_ADDR[1]	AC20	UTP_IP_SOC	AD20	UTP_IP_DATA[1]
AA21	VCC	AB21	SCANTESTMODE_N	AC21	VCC	AD21	UTP_IP_ADDR[4]
AA22	TXDATA1	AB22	VCCP	AC22	JTG_TRST_N	AD22	VSS
AA23	VSS	AB23	CTS0_N	AC23	VCC	AD23	JTG_TDO
AA24	GPIO[3]	AB24	CTS1_N	AC24	RXDATA0	AD24	VSS
AA25	VSS	AB25	VCCP	AC25	RTS1_N	AD25	TXDATA0
AA26	GPIO[7]	AB26	GPIO[4]	AC26	GPIO[2]	AD26	RTS0_N

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 20. Ball Map Assignment for Intel® IXP425 Network Processor (Sheet 7 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
AE1	ETH_RXDATA0[3]	AF1	ETH_RXDATA0[2]				
AE2	VCCP	AF2	ETH_MDIO				
AE3	ETH_COL0	AF3	(Reserved)				
AE4	ETH_TXEN1	AF4	ETH_TXDATA1[3]				
AE5	VCCP	AF5	ETH_TXCLK1				
AE6	ETH_RXDV1	AF6	ETH_RXDATA1[0]				
AE7	VSS	AF7	ETH_CRS1				
AE8	ETH_COL1	AF8	VSSOSC				
AE9	VCCP	AF9	OSC_IN				
AE10	VCCPLL1	AF10	VSSOSCP				
AE11	VSS	AF11	OSC_OUT				
AE12	VCCPLL2	AF12	VCCOSC				
AE13	VCCP	AF13	BYPASS_CLK				
AE14	UTP_OP_DATA[5]	AF14	UTP_OP_DATA[6]				
AE15	VSS	AF15	UTP_OP_DATA[3]				
AE16	UTP_OP_FCO	AF16	UTP_OP_DATA[0]				
AE17	VCCP	AF17	UTP_OP_CLK				
AE18	UTP_OP_ADDR[2]	AF18	UTP_OP_ADDR[4]				
AE19	VSS	AF19	UTP_OP_ADDR[0]				
AE20	UTP_IP_DATA[4]	AF20	UTP_IP_DATA[5]				
AE21	VCCP	AF21	UTP_IP_DATA[3]				
AE22	UTP_IP_FCO	AF22	UTP_IP_DATA[0]				
AE23	VCCP	AF23	UTP_IP_ADDR[3]				
AE24	JTG_TDI	AF24	UTP_IP_ADDR[2]				
AE25	VCCP	AF25	JTG_TMS				
AE26	HIGHZ_N	AF26	JTG_TCK				

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 21. Ball Map Assignment for Intel® IXP422 Network Processor (Sheet 1 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
A1	PCI_AD[27]	B1	PCI_AD[28]	C1	PCI_AD[26]	D1	PCI_AD[25]
A2	PCI_GNT_N[1]	B2	VCCP	C2	PCI_AD[30]	D2	VSS
A3	PCI_GNT_N[3]	B3	PCI_GNT_N[2]	C3	VSS	D3	PCI_AD[31]
A4	SDM_DATA[19]	B4	VCCP	C4	PCI_INTA_N	D4	VCC
A5	SDM_DATA[27]	B5	SDM_DATA[28]	C5	VSS	D5	PCI_SERR_N
A6	SDM_DATA[26]	B6	VCCP	C6	SDM_DATA[18]	D6	VCC
A7	SDM_DATA[25]	B7	SDM_DATA[21]	C7	VSS	D7	SDM_DATA[29]
A8	SDM_DATA[23]	B8	VSS	C8	VCCP	D8	SDM_DATA[20]
A9	SDM_DATA[14]	B9	SDM_DATA[0]	C9	SDM_DATA[24]	D9	VCC
A10	SDM_DATA[13]	B10	VCCP	C10	VSS	D10	SDM_DATA[15]
A11	SDM_DATA[11]	B11	SDM_DATA[12]	C11	SDM_DATA[2]	D11	SDM_DATA[1]
A12	SDM_DATA[10]	B12	VSS	C12	SDM_DATA[4]	D12	VCC
A13	SDM_DATA[6]	B13	SDM_DATA[9]	C13	VSS	D13	SDM_DATA[5]
A14	SDM_DATA[8]	B14	VCCP	C14	SDM_DATA[7]	D14	VCC
A15	SDM_DQM[1]	B15	SDM_DQM[2]	C15	SDM_DQM[3]	D15	SDM_WE_N
A16	SDM_CS_N[0]	B16	VSS	C16	VCCP	D16	SDM_CS_N[1]
A17	SDM_CLKOUT	B17	SDM_CKE	C17	SDM_CAS_N	D17	SDM_BA[1]
A18	SDM_RAS_N	B18	VCCP	C18	SDM_ADDR[11]	D18	VCC
A19	SDM_ADDR[12]	B19	SDM_ADDR[10]	C19	VSS	D19	SDM_ADDR[0]
A20	SDM_ADDR[9]	B20	VSS	C20	SDM_ADDR[6]	D20	VSS
A21	SDM_ADDR[8]	B21	SDM_ADDR[1]	C21	SDM_ADDR[2]	D21	VCC
A22	SDM_ADDR[5]	B22	VCCP	C22	VSS	D22	EX_ALE
A23	EX_RD_N	B23	EX_IOWAIT_N	C23	EX_ADDR[0]	D23	VCC
A24	EX_ADDR[1]	B24	VSS	C24	EX_ADDR[4]	D24	EX_ADDR[6]
A25	EX_ADDR[3]	B25	VCCP	C25	EX_ADDR[7]	D25	RCOMP
A26	EX_ADDR[5]	B26	EX_ADDR[9]	C26	EX_ADDR[13]	D26	EX_ADDR[17]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 21. Ball Map Assignment for Intel® IXP422 Network Processor (Sheet 2 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
E1	PCI_AD[23]	F1	PCI_AD[20]	G1	PCI_AD[21]	H1	PCI_AD[16]
E2	VCCP	F2	PCI_IDSEL	G2	VCCP	H2	PCI_AD[18]
E3	PCI_REQ_N[2]	F3	VCC	G3	PCI_AD[24]	H3	VCC
E4	VSS	F4	PCI_REQ_N[0]	G4	VSS	H4	PCI_CBE_N[3]
E5	PCI_GNT_N[0]	F5	VCCP	G5	PCI_REQ_N[1]	H5	VCC
E6	SDM_DATA[16]	F6	VCC	G6	VSS	H6	PCI_REQ_N[3]
E7	VCCP	F7	SDM_DATA[31]				
E8	SDM_DATA[30]	F8	VSS				
E9	VSS	F9	SDM_DATA[17]				
E10	SDM_DATA[22]	F10	VCC				
E11	VCCP						
E12	SDM_DATA[3]						
E13	VSS						
E14	SDM_DQM[0]						
E15	VCCP						
E16	SDM_BA[0]						
E17	VSS	F17	VCC				
E18	SDM_ADDR[7]	F18	SDM_ADDR[4]				
E19	VCCP	F19	VSS				
E20	SDM_ADDR[3]	F20	USB_DPOS				
E21	USB_DNEG	F21	VCC	G21	EX_ADDR[2]	H21	VSS
E22	VCCP	F22	EX_WR_N	G22	VSS	H22	EX_ADDR[11]
E23	VSS	F23	VCC	G23	EX_ADDR[12]	H23	EX_ADDR[18]
E24	EX_ADDR[10]	F24	EX_ADDR[14]	G24	VSS	H24	VCCP
E25	EX_ADDR[15]	F25	VCCP	G25	EX_ADDR[20]	H25	VSS
E26	EX_ADDR[19]	F26	EX_ADDR[21]	G26	EX_ADDR[22]	H26	EX_CS_N[1]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 21. Ball Map Assignment for Intel® IXP422 Network Processor (Sheet 3 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
J1	PCI_CLKIN	K1	PCI_CBE_N[2]	L1	PCI_DEVSEL_N	M1	PCI_CBE_N[1]
J2	VCCP	K2	VSS	L2	VCCP	M2	PCI_PAR
J3	VSS	K3	PCI_AD[17]	L3	PCI_STOP_N	M3	VSS
J4	PCI_AD[22]	K4	VCCP	L4	VCC	M4	PCI_IRDY_N
J5	VSS	K5	PCI_AD[19]	L5	PCI_FRAME_N	M5	VCCP
J6	PCI_AD[29]	K6	VCC				
				L11	VSS	M11	VSS
				L12	VSS	M12	VSS
				L13	VSS	M13	VSS
				L14	VSS	M14	VSS
				L15	VSS	M15	VSS
				L16	VSS	M16	VSS
J21	EX_ADDR[8]	K21	VCC				
J22	EX_ADDR[16]	K22	VSS	L22	VCCP	M22	EX_CS_N[5]
J23	VCC	K23	EX_CS_N[0]	L23	VCC	M23	EX_CLK
J24	EX_ADDR[23]	K24	EX_CS_N[3]	L24	EX_CS_N[6]	M24	EX_DATA[2]
J25	EX_CS_N[2]	K25	VCCP	L25	EX_DATA[0]	M25	VSS
J26	EX_CS_N[4]	K26	EX_CS_N[7]	L26	EX_DATA[1]	M26	EX_DATA[3]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 21. Ball Map Assignment for Intel® IXP422 Network Processor (Sheet 4 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
N1	PCI_AD[11]	P1	PCI_CBE_N[0]	R1	PCI_AD[10]	T1	PCI_AD[6]
N2	VCCP	P2	PCI_AD[14]	R2	VSS	T2	PCI_TRDY_N
N3	VCC	P3	PCI_AD[13]	R3	PCI_AD[9]	T3	VSS
N4	PCI_PERR_N	P4	VSS	R4	VCC	T4	PCI_AD[2]
N5	PCI_AD[15]	P5	PCI_AD[12]	R5	PCI_AD[4]	T5	VCCP
N11	VSS	P11	VSS	R11	VSS	T11	VSS
N12	VSS	P12	VSS	R12	VSS	T12	VSS
N13	VSS	P13	VSS	R13	VSS	T13	VSS
N14	VSS	P14	VSS	R14	VSS	T14	VSS
N15	VSS	P15	VSS	R15	VSS	T15	VSS
N16	VSS	P16	VSS	R16	VSS	T16	VSS
N22	VCC	P22	EX_DATA[6]	R22	VCCP	T22	EX_RDY_N[0]
N23	VSS	P23	EX_DATA[7]	R23	VCC	T23	VSS
N24	VCC	P24	EX_DATA[8]	R24	EX_DATA[12]	T24	EX_DATA[14]
N25	EX_DATA[4]	P25	VCCP	R25	EX_DATA[11]	T25	VSS
N26	EX_DATA[5]	P26	EX_DATA[9]	R26	EX_DATA[10]	T26	EX_DATA[13]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 21. Ball Map Assignment for Intel® IXP422 Network Processor (Sheet 5 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
U1	PCI_AD[8]	V1	PCI_AD[5]	W1	PCI_AD[1]	Y1	N/C
U2	VCCP	V2	VSS	W2	VCCP	Y2	N/C
U3	PCI_AD[0]	V3	PCI_AD[3]	W3	N/C	Y3	N/C
U4	PCI_AD[7]	V4	VCC	W4	VSS	Y4	VCC
U5	N/C	V5	N/C	W5	N/C	Y5	VCCP
U6	VCC	V6	VSS	W6	N/C	Y6	ETH_TXEN0
U21	VCC	V21	GPIO[6]	W21	GPIO[1]	Y21	RXDATA1
U22	GPIO[14]	V22	GPIO[9]	W22	VCCP	Y22	GPIO[0]
U23	EX_RDY_N[1]	V23	VCC	W23	GPIO[8]	Y23	VCC
U24	EX_RDY_N[2]	V24	GPIO[13]	W24	VSS	Y24	GPIO[5]
U25	GPIO[15]	V25	VCCP	W25	GPIO[11]	Y25	VCCP
U26	EX_DATA[15]	V26	EX_RDY_N[3]	W26	GPIO[12]	Y26	GPIO[10]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 21. Ball Map Assignment for Intel® IXP422 Network Processor (Sheet 6 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
AA1	N/C	AB1	N/C	AC1	VSS	AD1	ETH_TXCLK0
AA2	VCCP	AB2	N/C	AC2	ETH_TXDATA0[0]	AD2	ETH_RXDV0
AA3	VSS	AB3	ETH_TXDATA0[3]	AC3	VCCP	AD3	VSS
AA4	N/C	AB4	ETH_TXDATA0[1]	AC4	VCC	AD4	ETH_CRS0
AA5	ETH_TXDATA0[2]	AB5	VSS	AC5	ETH_RXDATA0[0]	AD5	ETH_MDC
AA6	VCC	AB6	ETH_RXCLK0	AC6	VSS	AD6	ETH_TXDATA1[0]
AA7	ETH_RXDATA0[1]	AB7	VCCP	AC7	VCC	AD7	ETH_RXDATA1[3]
AA8	VSS	AB8	ETH_TXDATA1[2]	AC8	ETH_RXDATA1[2]	AD8	ETH_RXCLK1
AA9	ETH_TXDATA1[1]	AB9	ETH_RXDATA1[1]	AC9	VCC	AD9	VSS
AA10	VCC	AB10	VCCP	AC10	VCC	AD10	VSSOSCP
		AB11	VCCP	AC11	VCCOSCP	AD11	VCCP
		AB12	VSS	AC12	VCC	AD12	PLL_LOCK
		AB13	N/C	AC13	RESET_IN_N	AD13	PWRON_RESET_N
		AB14	VCCP	AC14	VCC	AD14	N/C
		AB15	N/C	AC15	N/C	AD15	N/C
		AB16	VSS	AC16	N/C	AD16	VSS
AA17	VCC	AB17	N/C	AC17	N/C	AD17	N/C
AA18	N/C	AB18	VCCP	AC18	VCC	AD18	N/C
AA19	N/C	AB19	N/C	AC19	N/C	AD19	VCCP
AA20	VSS	AB20	N/C	AC20	N/C	AD20	N/C
AA21	VCC	AB21	SCANTESTMODE_N	AC21	VCC	AD21	N/C
AA22	TXDATA1	AB22	VCCP	AC22	JTG_TRST_N	AD22	VSS
AA23	VSS	AB23	CTS0_N	AC23	VCC	AD23	JTG_TDO
AA24	GPIO[3]	AB24	CTS1_N	AC24	RXDATA0	AD24	VSS
AA25	VSS	AB25	VCCP	AC25	RTS1_N	AD25	TXDATA0
AA26	GPIO[7]	AB26	GPIO[4]	AC26	GPIO[2]	AD26	RTS0_N

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 21. Ball Map Assignment for Intel® IXP422 Network Processor (Sheet 7 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
AE1	ETH_RXDATA0[3]	AF1	ETH_RXDATA0[2]				
AE2	VCCP	AF2	ETH_MDIO				
AE3	ETH_COL0	AF3	(Reserved)				
AE4	ETH_TXEN1	AF4	ETH_TXDATA1[3]				
AE5	VCCP	AF5	ETH_TXCLK1				
AE6	ETH_RXDV1	AF6	ETH_RXDATA1[0]				
AE7	VSS	AF7	ETH_CRS1				
AE8	ETH_COL1	AF8	VSSOSC				
AE9	VCCP	AF9	OSC_IN				
AE10	VCCPLL1	AF10	VSSOSCP				
AE11	VSS	AF11	OSC_OUT				
AE12	VCCPLL2	AF12	VCCOSC				
AE13	VCCP	AF13	BYPASS_CLK				
AE14	N/C	AF14	N/C				
AE15	VSS	AF15	N/C				
AE16	N/C	AF16	N/C				
AE17	VCCP	AF17	N/C				
AE18	N/C	AF18	N/C				
AE19	VSS	AF19	N/C				
AE20	N/C	AF20	N/C				
AE21	VCCP	AF21	N/C				
AE22	N/C	AF22	N/C				
AE23	VCCP	AF23	N/C				
AE24	JTG_TDI	AF24	N/C				
AE25	VCCP	AF25	JTG_TMS				
AE26	HIGHZ_N	AF26	JTG_TCK				

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 22. Ball Map Assignment for Intel® IXP421 Network Processor (Sheet 1 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
A1	PCI_AD[27]	B1	PCI_AD[28]	C1	PCI_AD[26]	D1	PCI_AD[25]
A2	PCI_GNT_N[1]	B2	VCCP	C2	PCI_AD[30]	D2	VSS
A3	PCI_GNT_N[3]	B3	PCI_GNT_N[2]	C3	VSS	D3	PCI_AD[31]
A4	SDM_DATA[19]	B4	VCCP	C4	PCI_INTA_N	D4	VCC
A5	SDM_DATA[27]	B5	SDM_DATA[28]	C5	VSS	D5	PCI_SERR_N
A6	SDM_DATA[26]	B6	VCCP	C6	SDM_DATA[18]	D6	VCC
A7	SDM_DATA[25]	B7	SDM_DATA[21]	C7	VSS	D7	SDM_DATA[29]
A8	SDM_DATA[23]	B8	VSS	C8	VCCP	D8	SDM_DATA[20]
A9	SDM_DATA[14]	B9	SDM_DATA[0]	C9	SDM_DATA[24]	D9	VCC
A10	SDM_DATA[13]	B10	VCCP	C10	VSS	D10	SDM_DATA[15]
A11	SDM_DATA[11]	B11	SDM_DATA[12]	C11	SDM_DATA[2]	D11	SDM_DATA[1]
A12	SDM_DATA[10]	B12	VSS	C12	SDM_DATA[4]	D12	VCC
A13	SDM_DATA[6]	B13	SDM_DATA[9]	C13	VSS	D13	SDM_DATA[5]
A14	SDM_DATA[8]	B14	VCCP	C14	SDM_DATA[7]	D14	VCC
A15	SDM_DQM[1]	B15	SDM_DQM[2]	C15	SDM_DQM[3]	D15	SDM_WE_N
A16	SDM_CS_N[0]	B16	VSS	C16	VCCP	D16	SDM_CS_N[1]
A17	SDM_CLKOUT	B17	SDM_CKE	C17	SDM_CAS_N	D17	SDM_BA[1]
A18	SDM_RAS_N	B18	VCCP	C18	SDM_ADDR[11]	D18	VCC
A19	SDM_ADDR[12]	B19	SDM_ADDR[10]	C19	VSS	D19	SDM_ADDR[0]
A20	SDM_ADDR[9]	B20	VSS	C20	SDM_ADDR[6]	D20	VSS
A21	SDM_ADDR[8]	B21	SDM_ADDR[1]	C21	SDM_ADDR[2]	D21	VCC
A22	SDM_ADDR[5]	B22	VCCP	C22	VSS	D22	EX_ALE
A23	EX_RD_N	B23	EX_IOWAIT_N	C23	EX_ADDR[0]	D23	VCC
A24	EX_ADDR[1]	B24	VSS	C24	EX_ADDR[4]	D24	EX_ADDR[6]
A25	EX_ADDR[3]	B25	VCCP	C25	EX_ADDR[7]	D25	RCOMP
A26	EX_ADDR[5]	B26	EX_ADDR[9]	C26	EX_ADDR[13]	D26	EX_ADDR[17]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 22. Ball Map Assignment for Intel® IXP421 Network Processor (Sheet 2 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
E1	PCI_AD[23]	F1	PCI_AD[20]	G1	PCI_AD[21]	H1	PCI_AD[16]
E2	VCCP	F2	PCI_IDSEL	G2	VCCP	H2	PCI_AD[18]
E3	PCI_REQ_N[2]	F3	VCC	G3	PCI_AD[24]	H3	VCC
E4	VSS	F4	PCI_REQ_N[0]	G4	VSS	H4	PCI_CBE_N[3]
E5	PCI_GNT_N[0]	F5	VCCP	G5	PCI_REQ_N[1]	H5	VCC
E6	SDM_DATA[16]	F6	VCC	G6	VSS	H6	PCI_REQ_N[3]
E7	VCCP	F7	SDM_DATA[31]				
E8	SDM_DATA[30]	F8	VSS				
E9	VSS	F9	SDM_DATA[17]				
E10	SDM_DATA[22]	F10	VCC				
E11	VCCP						
E12	SDM_DATA[3]						
E13	VSS						
E14	SDM_DQM[0]						
E15	VCCP						
E16	SDM_BA[0]						
E17	VSS	F17	VCC				
E18	SDM_ADDR[7]	F18	SDM_ADDR[4]				
E19	VCCP	F19	VSS				
E20	SDM_ADDR[3]	F20	USB_DPOS				
E21	USB_DNEG	F21	VCC	G21	EX_ADDR[2]	H21	VSS
E22	VCCP	F22	EX_WR_N	G22	VSS	H22	EX_ADDR[11]
E23	VSS	F23	VCC	G23	EX_ADDR[12]	H23	EX_ADDR[18]
E24	EX_ADDR[10]	F24	EX_ADDR[14]	G24	VSS	H24	VCCP
E25	EX_ADDR[15]	F25	VCCP	G25	EX_ADDR[20]	H25	VSS
E26	EX_ADDR[19]	F26	EX_ADDR[21]	G26	EX_ADDR[22]	H26	EX_CS_N[1]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 22. Ball Map Assignment for Intel® IXP421 Network Processor (Sheet 3 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
J1	PCI_CLKIN	K1	PCI_CBE_N[2]	L1	PCI_DEVSEL_N	M1	PCI_CBE_N[1]
J2	VCCP	K2	VSS	L2	VCCP	M2	PCI_PAR
J3	VSS	K3	PCI_AD[17]	L3	PCI_STOP_N	M3	VSS
J4	PCI_AD[22]	K4	VCCP	L4	VCC	M4	PCI_IRDY_N
J5	VSS	K5	PCI_AD[19]	L5	PCI_FRAME_N	M5	VCCP
J6	PCI_AD[29]	K6	VCC				
				L11	VSS	M11	VSS
				L12	VSS	M12	VSS
				L13	VSS	M13	VSS
				L14	VSS	M14	VSS
				L15	VSS	M15	VSS
				L16	VSS	M16	VSS
J21	EX_ADDR[8]	K21	VCC				
J22	EX_ADDR[16]	K22	VSS	L22	VCCP	M22	EX_CS_N[5]
J23	VCC	K23	EX_CS_N[0]	L23	VCC	M23	EX_CLK
J24	EX_ADDR[23]	K24	EX_CS_N[3]	L24	EX_CS_N[6]	M24	EX_DATA[2]
J25	EX_CS_N[2]	K25	VCCP	L25	EX_DATA[0]	M25	VSS
J26	EX_CS_N[4]	K26	EX_CS_N[7]	L26	EX_DATA[1]	M26	EX_DATA[3]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 22. Ball Map Assignment for Intel® IXP421 Network Processor (Sheet 4 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
N1	PCI_AD[11]	P1	PCI_CBE_N[0]	R1	PCI_AD[10]	T1	PCI_AD[6]
N2	VCCP	P2	PCI_AD[14]	R2	VSS	T2	PCI_TRDY_N
N3	VCC	P3	PCI_AD[13]	R3	PCI_AD[9]	T3	VSS
N4	PCI_PERR_N	P4	VSS	R4	VCC	T4	PCI_AD[2]
N5	PCI_AD[15]	P5	PCI_AD[12]	R5	PCI_AD[4]	T5	VCCP
N11	VSS	P11	VSS	R11	VSS	T11	VSS
N12	VSS	P12	VSS	R12	VSS	T12	VSS
N13	VSS	P13	VSS	R13	VSS	T13	VSS
N14	VSS	P14	VSS	R14	VSS	T14	VSS
N15	VSS	P15	VSS	R15	VSS	T15	VSS
N16	VSS	P16	VSS	R16	VSS	T16	VSS
N22	VCC	P22	EX_DATA[6]	R22	VCCP	T22	EX_RDY_N[0]
N23	VSS	P23	EX_DATA[7]	R23	VCC	T23	VSS
N24	VCC	P24	EX_DATA[8]	R24	EX_DATA[12]	T24	EX_DATA[14]
N25	EX_DATA[4]	P25	VCCP	R25	EX_DATA[11]	T25	VSS
N26	EX_DATA[5]	P26	EX_DATA[9]	R26	EX_DATA[10]	T26	EX_DATA[13]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 22. Ball Map Assignment for Intel® IXP421 Network Processor (Sheet 5 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
U1	PCI_AD[8]	V1	PCI_AD[5]	W1	PCI_AD[1]	Y1	HSS_TXCLK0
U2	VCCP	V2	VSS	W2	VCCP	Y2	HSS_RXCLK0
U3	PCI_AD[0]	V3	PCI_AD[3]	W3	HSS_RXFRAME0	Y3	HSS_TXFRAME1
U4	PCI_AD[7]	V4	VCC	W4	VSS	Y4	VCC
U5	HSS_TXDATA0	V5	HSS_TXFRAME0	W5	HSS_TXCLK1	Y5	VCCP
U6	VCC	V6	VSS	W6	HSS_RXFRAME1	Y6	ETH_TXEN0
U21	VCC	V21	GPIO[6]	W21	GPIO[1]	Y21	RXDATA1
U22	GPIO[14]	V22	GPIO[9]	W22	VCCP	Y22	GPIO[0]
U23	EX_RDY_N[1]	V23	VCC	W23	GPIO[8]	Y23	VCC
U24	EX_RDY_N[2]	V24	GPIO[13]	W24	VSS	Y24	GPIO[5]
U25	GPIO[15]	V25	VCCP	W25	GPIO[11]	Y25	VCCP
U26	EX_DATA[15]	V26	EX_RDY_N[3]	W26	GPIO[12]	Y26	GPIO[10]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 22. Ball Map Assignment for Intel® IXP421 Network Processor (Sheet 6 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
AA1	HSS_RXDATA0	AB1	HSS_TXDATA1	AC1	VSS	AD1	ETH_TXCLK0
AA2	VCCP	AB2	HSS_RXDATA1	AC2	ETH_TXDATA0[0]	AD2	ETH_RXDV0
AA3	VSS	AB3	ETH_TXDATA0[3]	AC3	VCCP	AD3	VSS
AA4	HSS_RXCLK1	AB4	ETH_TXDATA0[1]	AC4	VCC	AD4	ETH_CRS0
AA5	ETH_TXDATA0[2]	AB5	VSS	AC5	ETH_RXDATA0[0]	AD5	ETH_MDC
AA6	VCC	AB6	ETH_RXCLK0	AC6	VSS	AD6	N/C
AA7	ETH_RXDATA0[1]	AB7	VCCP	AC7	VCC	AD7	N/C
AA8	VSS	AB8	N/C	AC8	N/C	AD8	N/C
AA9	N/C	AB9	N/C	AC9	VCC	AD9	VSS
AA10	VCC	AB10	VCCP	AC10	VCC	AD10	VSSOSCP
		AB11	VCCP	AC11	VCCOSCP	AD11	VCCP
		AB12	VSS	AC12	VCC	AD12	PLL_LOCK
		AB13	UTP_OP_DATA[7]	AC13	RESET_IN_N	AD13	PWRON_RESET_N
		AB14	VCCP	AC14	VCC	AD14	UTP_OP_DATA[4]
		AB15	UTP_OP_SOC	AC15	UTP_OP_DATA[1]	AD15	UTP_OP_DATA[2]
		AB16	VSS	AC16	UTP_OP_FCI	AD16	VSS
AA17	VCC	AB17	UTP_IP_DATA[6]	AC17	UTP_OP_ADDR[1]	AD17	UTP_OP_ADDR[3]
AA18	UTP_IP_FCI	AB18	VCCP	AC18	VCC	AD18	UTP_IP_DATA[7]
AA19	UTP_IP_ADDR[0]	AB19	UTP_IP_CLK	AC19	UTP_IP_DATA[2]	AD19	VCCP
AA20	VSS	AB20	UTP_IP_ADDR[1]	AC20	UTP_IP_SOC	AD20	UTP_IP_DATA[1]
AA21	VCC	AB21	SCANTESTMODE_N	AC21	VCC	AD21	UTP_IP_ADDR[4]
AA22	TXDATA1	AB22	VCCP	AC22	JTG_TRST_N	AD22	VSS
AA23	VSS	AB23	CTS0_N	AC23	VCC	AD23	JTG_TDO
AA24	GPIO[3]	AB24	CTS1_N	AC24	RXDATA0	AD24	VSS
AA25	VSS	AB25	VCCP	AC25	RTS1_N	AD25	TXDATA0
AA26	GPIO[7]	AB26	GPIO[4]	AC26	GPIO[2]	AD26	RTS0_N

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, "Functional Signal Definitions" on page 30.

Table 22. Ball Map Assignment for Intel® IXP421 Network Processor (Sheet 7 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
AE1	ETH_RXDATA0[3]	AF1	ETH_RXDATA0[2]				
AE2	VCCP	AF2	ETH_MDIO				
AE3	ETH_COL0	AF3	(Reserved)				
AE4	N/C	AF4	N/C				
AE5	VCCP	AF5	N/C				
AE6	N/C	AF6	N/C				
AE7	VSS	AF7	N/C				
AE8	N/C	AF8	VSSOSC				
AE9	VCCP	AF9	OSC_IN				
AE10	VCCPLL1	AF10	VSSOSCP				
AE11	VSS	AF11	OSC_OUT				
AE12	VCCPLL2	AF12	VCCOSC				
AE13	VCCP	AF13	BYPASS_CLK				
AE14	UTP_OP_DATA[5]	AF14	UTP_OP_DATA[6]				
AE15	VSS	AF15	UTP_OP_DATA[3]				
AE16	UTP_OP_FCO	AF16	UTP_OP_DATA[0]				
AE17	VCCP	AF17	UTP_OP_CLK				
AE18	UTP_OP_ADDR[2]	AF18	UTP_OP_ADDR[4]				
AE19	VSS	AF19	UTP_OP_ADDR[0]				
AE20	UTP_IP_DATA[4]	AF20	UTP_IP_DATA[5]				
AE21	VCCP	AF21	UTP_IP_DATA[3]				
AE22	UTP_IP_FCO	AF22	UTP_IP_DATA[0]				
AE23	VCCP	AF23	UTP_IP_ADDR[3]				
AE24	JTG_TDI	AF24	UTP_IP_ADDR[2]				
AE25	VCCP	AF25	JTG_TMS				
AE26	HIGHZ_N	AF26	JTG_TCK				

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 23. Ball Map Assignment for Intel® IXP420 Network Processor (Sheet 1 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
A1	PCI_AD[27]	B1	PCI_AD[28]	C1	PCI_AD[26]	D1	PCI_AD[25]
A2	PCI_GNT_N[1]	B2	VCCP	C2	PCI_AD[30]	D2	VSS
A3	PCI_GNT_N[3]	B3	PCI_GNT_N[2]	C3	VSS	D3	PCI_AD[31]
A4	SDM_DATA[19]	B4	VCCP	C4	PCI_INTA_N	D4	VCC
A5	SDM_DATA[27]	B5	SDM_DATA[28]	C5	VSS	D5	PCI_SERR_N
A6	SDM_DATA[26]	B6	VCCP	C6	SDM_DATA[18]	D6	VCC
A7	SDM_DATA[25]	B7	SDM_DATA[21]	C7	VSS	D7	SDM_DATA[29]
A8	SDM_DATA[23]	B8	VSS	C8	VCCP	D8	SDM_DATA[20]
A9	SDM_DATA[14]	B9	SDM_DATA[0]	C9	SDM_DATA[24]	D9	VCC
A10	SDM_DATA[13]	B10	VCCP	C10	VSS	D10	SDM_DATA[15]
A11	SDM_DATA[11]	B11	SDM_DATA[12]	C11	SDM_DATA[2]	D11	SDM_DATA[1]
A12	SDM_DATA[10]	B12	VSS	C12	SDM_DATA[4]	D12	VCC
A13	SDM_DATA[6]	B13	SDM_DATA[9]	C13	VSS	D13	SDM_DATA[5]
A14	SDM_DATA[8]	B14	VCCP	C14	SDM_DATA[7]	D14	VCC
A15	SDM_DQM[1]	B15	SDM_DQM[2]	C15	SDM_DQM[3]	D15	SDM_WE_N
A16	SDM_CS_N[0]	B16	VSS	C16	VCCP	D16	SDM_CS_N[1]
A17	SDM_CLKOUT	B17	SDM_CKE	C17	SDM_CAS_N	D17	SDM_BA[1]
A18	SDM_RAS_N	B18	VCCP	C18	SDM_ADDR[11]	D18	VCC
A19	SDM_ADDR[12]	B19	SDM_ADDR[10]	C19	VSS	D19	SDM_ADDR[0]
A20	SDM_ADDR[9]	B20	VSS	C20	SDM_ADDR[6]	D20	VSS
A21	SDM_ADDR[8]	B21	SDM_ADDR[1]	C21	SDM_ADDR[2]	D21	VCC
A22	SDM_ADDR[5]	B22	VCCP	C22	VSS	D22	EX_ALE
A23	EX_RD_N	B23	EX_IOWAIT_N	C23	EX_ADDR[0]	D23	VCC
A24	EX_ADDR[1]	B24	VSS	C24	EX_ADDR[4]	D24	EX_ADDR[6]
A25	EX_ADDR[3]	B25	VCCP	C25	EX_ADDR[7]	D25	RCOMP
A26	EX_ADDR[5]	B26	EX_ADDR[9]	C26	EX_ADDR[13]	D26	EX_ADDR[17]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 23. Ball Map Assignment for Intel® IXP420 Network Processor (Sheet 2 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
E1	PCI_AD[23]	F1	PCI_AD[20]	G1	PCI_AD[21]	H1	PCI_AD[16]
E2	VCCP	F2	PCI_IDSEL	G2	VCCP	H2	PCI_AD[18]
E3	PCI_REQ_N[2]	F3	VCC	G3	PCI_AD[24]	H3	VCC
E4	VSS	F4	PCI_REQ_N[0]	G4	VSS	H4	PCI_CBE_N[3]
E5	PCI_GNT_N[0]	F5	VCCP	G5	PCI_REQ_N[1]	H5	VCC
E6	SDM_DATA[16]	F6	VCC	G6	VSS	H6	PCI_REQ_N[3]
E7	VCCP	F7	SDM_DATA[31]				
E8	SDM_DATA[30]	F8	VSS				
E9	VSS	F9	SDM_DATA[17]				
E10	SDM_DATA[22]	F10	VCC				
E11	VCCP						
E12	SDM_DATA[3]						
E13	VSS						
E14	SDM_DQM[0]						
E15	VCCP						
E16	SDM_BA[0]						
E17	VSS	F17	VCC				
E18	SDM_ADDR[7]	F18	SDM_ADDR[4]				
E19	VCCP	F19	VSS				
E20	SDM_ADDR[3]	F20	USB_DPOS				
E21	USB_DNEG	F21	VCC	G21	EX_ADDR[2]	H21	VSS
E22	VCCP	F22	EX_WR_N	G22	VSS	H22	EX_ADDR[11]
E23	VSS	F23	VCC	G23	EX_ADDR[12]	H23	EX_ADDR[18]
E24	EX_ADDR[10]	F24	EX_ADDR[14]	G24	VSS	H24	VCCP
E25	EX_ADDR[15]	F25	VCCP	G25	EX_ADDR[20]	H25	VSS
E26	EX_ADDR[19]	F26	EX_ADDR[21]	G26	EX_ADDR[22]	H26	EX_CS_N[1]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 23. Ball Map Assignment for Intel® IXP420 Network Processor (Sheet 3 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
J1	PCI_CLKIN	K1	PCI_CBE_N[2]	L1	PCI_DEVSEL_N	M1	PCI_CBE_N[1]
J2	VCCP	K2	VSS	L2	VCCP	M2	PCI_PAR
J3	VSS	K3	PCI_AD[17]	L3	PCI_STOP_N	M3	VSS
J4	PCI_AD[22]	K4	VCCP	L4	VCC	M4	PCI_IRDY_N
J5	VSS	K5	PCI_AD[19]	L5	PCI_FRAME_N	M5	VCCP
J6	PCI_AD[29]	K6	VCC				
				L11	VSS	M11	VSS
				L12	VSS	M12	VSS
				L13	VSS	M13	VSS
				L14	VSS	M14	VSS
				L15	VSS	M15	VSS
				L16	VSS	M16	VSS
J21	EX_ADDR[8]	K21	VCC				
J22	EX_ADDR[16]	K22	VSS	L22	VCCP	M22	EX_CS_N[5]
J23	VCC	K23	EX_CS_N[0]	L23	VCC	M23	EX_CLK
J24	EX_ADDR[23]	K24	EX_CS_N[3]	L24	EX_CS_N[6]	M24	EX_DATA[2]
J25	EX_CS_N[2]	K25	VCCP	L25	EX_DATA[0]	M25	VSS
J26	EX_CS_N[4]	K26	EX_CS_N[7]	L26	EX_DATA[1]	M26	EX_DATA[3]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 23. Ball Map Assignment for Intel® IXP420 Network Processor (Sheet 4 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
N1	PCI_AD[11]	P1	PCI_CBE_N[0]	R1	PCI_AD[10]	T1	PCI_AD[6]
N2	VCCP	P2	PCI_AD[14]	R2	VSS	T2	PCI_TRDY_N
N3	VCC	P3	PCI_AD[13]	R3	PCI_AD[9]	T3	VSS
N4	PCI_PERR_N	P4	VSS	R4	VCC	T4	PCI_AD[2]
N5	PCI_AD[15]	P5	PCI_AD[12]	R5	PCI_AD[4]	T5	VCCP
N11	VSS	P11	VSS	R11	VSS	T11	VSS
N12	VSS	P12	VSS	R12	VSS	T12	VSS
N13	VSS	P13	VSS	R13	VSS	T13	VSS
N14	VSS	P14	VSS	R14	VSS	T14	VSS
N15	VSS	P15	VSS	R15	VSS	T15	VSS
N16	VSS	P16	VSS	R16	VSS	T16	VSS
N22	VCC	P22	EX_DATA[6]	R22	VCCP	T22	EX_RDY_N[0]
N23	VSS	P23	EX_DATA[7]	R23	VCC	T23	VSS
N24	VCC	P24	EX_DATA[8]	R24	EX_DATA[12]	T24	EX_DATA[14]
N25	EX_DATA[4]	P25	VCCP	R25	EX_DATA[11]	T25	VSS
N26	EX_DATA[5]	P26	EX_DATA[9]	R26	EX_DATA[10]	T26	EX_DATA[13]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 23. Ball Map Assignment for Intel® IXP420 Network Processor (Sheet 5 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
U1	PCI_AD[8]	V1	PCI_AD[5]	W1	PCI_AD[1]	Y1	N/C
U2	VCCP	V2	VSS	W2	VCCP	Y2	N/C
U3	PCI_AD[0]	V3	PCI_AD[3]	W3	N/C	Y3	N/C
U4	PCI_AD[7]	V4	VCC	W4	VSS	Y4	VCC
U5	N/C	V5	N/C	W5	N/C	Y5	VCCP
U6	VCC	V6	VSS	W6	N/C	Y6	ETH_TXEN0
U21	VCC	V21	GPIO[6]	W21	GPIO[1]	Y21	RXDATA1
U22	GPIO[14]	V22	GPIO[9]	W22	VCCP	Y22	GPIO[0]
U23	EX_RDY_N[1]	V23	VCC	W23	GPIO[8]	Y23	VCC
U24	EX_RDY_N[2]	V24	GPIO[13]	W24	VSS	Y24	GPIO[5]
U25	GPIO[15]	V25	VCCP	W25	GPIO[11]	Y25	VCCP
U26	EX_DATA[15]	V26	EX_RDY_N[3]	W26	GPIO[12]	Y26	GPIO[10]

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 23. Ball Map Assignment for Intel® IXP420 Network Processor (Sheet 6 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
AA1	N/C	AB1	N/C	AC1	VSS	AD1	ETH_TXCLK0
AA2	VCCP	AB2	N/C	AC2	ETH_TXDATA0[0]	AD2	ETH_RXDV0
AA3	VSS	AB3	ETH_TXDATA0[3]	AC3	VCCP	AD3	VSS
AA4	N/C	AB4	ETH_TXDATA0[1]	AC4	VCC	AD4	ETH_CRS0
AA5	ETH_TXDATA0[2]	AB5	VSS	AC5	ETH_RXDATA0[0]	AD5	ETH_MDC
AA6	VCC	AB6	ETH_RXCLK0	AC6	VSS	AD6	ETH_TXDATA1[0]
AA7	ETH_RXDATA0[1]	AB7	VCCP	AC7	VCC	AD7	ETH_RXDATA1[3]
AA8	VSS	AB8	ETH_TXDATA1[2]	AC8	ETH_RXDATA1[2]	AD8	ETH_RXCLK1
AA9	ETH_TXDATA1[1]	AB9	ETH_RXDATA1[1]	AC9	VCC	AD9	VSS
AA10	VCC	AB10	VCCP	AC10	VCC	AD10	VSSOSCP
		AB11	VCCP	AC11	VCCOSCP	AD11	VCCP
		AB12	VSS	AC12	VCC	AD12	PLL_LOCK
		AB13	N/C	AC13	RESET_IN_N	AD13	PWRON_RESET_N
		AB14	VCCP	AC14	VCC	AD14	N/C
		AB15	N/C	AC15	N/C	AD15	N/C
		AB16	VSS	AC16	N/C	AD16	VSS
AA17	VCC	AB17	N/C	AC17	N/C	AD17	N/C
AA18	N/C	AB18	VCCP	AC18	VCC	AD18	N/C
AA19	N/C	AB19	N/C	AC19	N/C	AD19	VCCP
AA20	VSS	AB20	N/C	AC20	N/C	AD20	N/C
AA21	VCC	AB21	SCANTESTMODE_N	AC21	VCC	AD21	N/C
AA22	TXDATA1	AB22	VCCP	AC22	JTG_TRST_N	AD22	VSS
AA23	VSS	AB23	CTS0_N	AC23	VCC	AD23	JTG_TDO
AA24	GPIO[3]	AB24	CTS1_N	AC24	RXDATA0	AD24	VSS
AA25	VSS	AB25	VCCP	AC25	RTS1_N	AD25	TXDATA0
AA26	GPIO[7]	AB26	GPIO[4]	AC26	GPIO[2]	AD26	RTS0_N

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

Table 23. Ball Map Assignment for Intel® IXP420 Network Processor (Sheet 7 of 7)

Ball	Signal	Ball	Signal	Ball	Signal	Ball	Signal
AE1	ETH_RXDATA0[3]	AF1	ETH_RXDATA0[2]				
AE2	VCCP	AF2	ETH_MDIO				
AE3	ETH_COL0	AF3	(Reserved)				
AE4	ETH_TXEN1	AF4	ETH_TXDATA1[3]				
AE5	VCCP	AF5	ETH_TXCLK1				
AE6	ETH_RXDV1	AF6	ETH_RXDATA1[0]				
AE7	VSS	AF7	ETH_CRS1				
AE8	ETH_COL1	AF8	VSSOSC				
AE9	VCCP	AF9	OSC_IN				
AE10	VCCPLL1	AF10	VSSOSCP				
AE11	VSS	AF11	OSC_OUT				
AE12	VCCPLL2	AF12	VCCOSC				
AE13	VCCP	AF13	BYPASS_CLK				
AE14	N/C	AF14	N/C				
AE15	VSS	AF15	N/C				
AE16	N/C	AF16	N/C				
AE17	VCCP	AF17	N/C				
AE18	N/C	AF18	N/C				
AE19	VSS	AF19	N/C				
AE20	N/C	AF20	N/C				
AE21	VCCP	AF21	N/C				
AE22	N/C	AF22	N/C				
AE23	VCCP	AF23	N/C				
AE24	JTG_TDI	AF24	N/C				
AE25	VCCP	AF25	JTG_TMS				
AE26	HIGHZ_N	AF26	JTG_TCK				

NOTE: Interfaces not being utilized at a system level require external pull-up or pull-down resistors. For specific details and requirements, see Section 4.1, “Functional Signal Definitions” on page 30.

4.4 Package Thermal Specifications

The thermal characterization parameter "Ψ_{JT}" is proportional to the temperature difference between the top, center of the package and the junction temperature.

This can be a useful value for verifying device temperatures in an actual environment. By measuring the package of the device, the junction temperature can be estimated, if the thermal characterization parameter has been measured under similar conditions.

The use of Ψ_{JT} should not be confused with Θ_{jc}, which is the thermal resistance from the device junction to the external surface of the package or case nearest the die attachment — as the case is held at a constant temperature.

Case temperature = Junction Temperature - (Ψ_{JT} * Power Dissipation)

$T_{JC} = T_J - (\Psi_{JT} * \text{Power Dissipation})$

The case temperature can then be monitored to make sure that the maximum junction temperature is not violated. Examples are given in the following sections.

Note: For more information on YJT, refer to the EIA/JEDEC Standard 51-2, Section 4.

4.4.1 Commercial Temperature

"Commercial" temperature range is defined in terms of the ambient temperature range, which is specified as 0° C to 70° C. The maximum power (P) is 2.4 W and the maximum junction temperature (T_j) is 115 ° C.

Ψ_{JT} for commercial temperature is 0.89° C/W.

Using the preceding junction-temperature formula, the commercial temperature for a 266-MHz part — assuming a maximum power of 2 W — would be:

$T_{JC} = 115° C - (0.89 * 2.0)$

$T_{JC} = 113.22° C$

4.4.2 Extended Temperature

"Extended" temperature range is defined in terms of the ambient temperature range, which is specified as -40° C to 85° C. The maximum power (P) is 2.4 W and the maximum junction temperature (T_j) is 115° C.

Ψ_{JT} for extended temperature is 0.32° C/W.

Using the preceding junction-temperature formula, the industrial temperature for a 533-MHz part — assuming a maximum power of 2.4 W — would be:

$T_{JC} = 115° C - (0.32 * 2.4)$

$T_{JC} = 114.23° C$

5.0 Electrical Specifications

5.1 Absolute Maximum Ratings

Parameter	Maximum Rating
Ambient Air Temperature (Extended)	-40º C to 85º C
Ambient Air Temperature (Commercial)	0º C to 70º C
Supply Voltage Core	-0.3 V to 2.1 V
Supply Voltage I/O	-0.3 V to 3.6 V
Supply Voltage Oscillator (V_{CCOSC})	-0.3 V to 2.1 V
Supply Voltage Oscillator (V_{CCOSCP})	-0.3 V to 3.6 V
Supply Voltage PLL (V_{CCPLL1})	-0.3 V to 2.1 V
Supply Voltage PLL (V_{CCPLL2})	-0.3 V to 2.1 V
Voltage On Any I/O Ball	-0.3 V to 3. 6V
Storage Temperature	-55º C to 125º C

Warning: Stressing the device beyond the “absolute maximum ratings” may cause permanent damage. These are stress ratings only. Operation beyond the “operating conditions” is not recommended and extended exposure beyond the “operating conditions” may affect device reliability.

5.2 V_{CCPLL1}, V_{CCPLL2}, V_{CCOSCP}, V_{CCOSC} Pin Requirements

To reduce voltage-supply noise on the analog sections of the Intel® IXP4XX Product Line of Network Processors, the phase-lock loop circuits (V_{CCPLL1}, V_{CCPLL2}) and oscillator circuit (V_{CCOSCP}, V_{CCOSC}) require isolated voltage supplies.

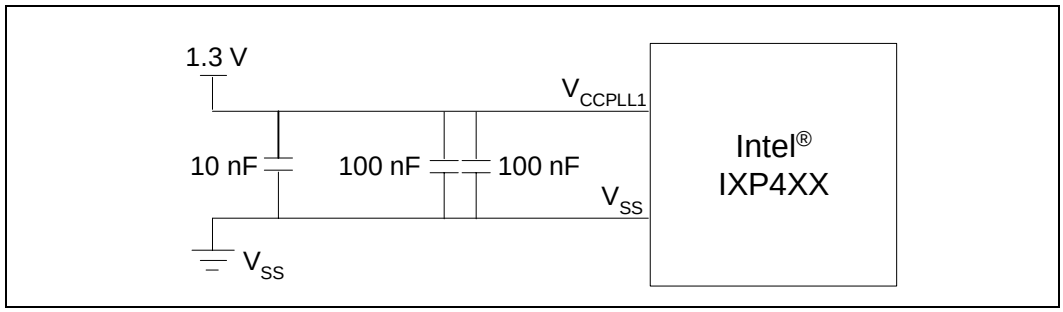
The filter circuits for each supply are shown in the following sections.

5.2.1 V_{CCPLL1} Requirement

A parallel combination of a 10-nF capacitor — for bypass — and a 200-nF capacitor — for a first-order filter with a cut-off frequency below 30 MHz — must be connected to the V_{CCPLL1} pin of the Intel® IXP4XX Product Line of Network Processors.

The ground of both capacitors should be connected to the nearest V_{SS} supply pin. Both capacitors should be located less than 0.5 inch away from the V_{CCPLL1} pin and the associated V_{SS} pin. In order to achieve the 200-nF capacitance, a parallel combination of two 100-nF capacitors may be used as long as the capacitors are placed directly beside each other.

Figure 8. V_{CCPLL1} Power Filtering Diagram

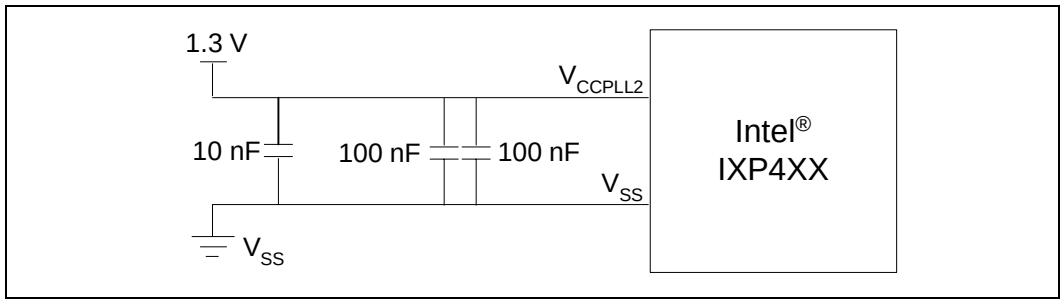


5.2.2 V_{CCPLL2} Requirement

A parallel combination of a 10-nF capacitor — for bypass — and a 200-nF capacitor — for a first-order filter with a cut-off frequency below 30 MHz — must be connected to the V_{CCPLL2} pin of the Intel® IXP4XX Product Line of Network Processors.

The ground of both capacitors should be connected to the nearest V_{SS} supply pin. Both capacitors should be located less than 0.5 inch away from the V_{CCPLL2} pin and the associated V_{SS} pin. In order to achieve the 200-nF capacitance, a parallel combination of two 100-nF capacitors may be used as long as the capacitors are placed directly beside each other.

Figure 9. V_{CCPLL2} Power Filtering Diagram



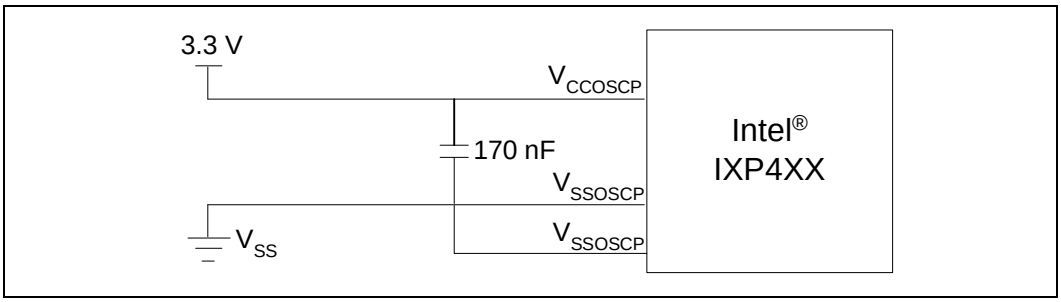
5.2.3 V_{CCOSCP} Requirement

A single 170-nF capacitor must be connected between the V_{CCP_OSC} pin and V_{SSP_OSC} pin of the Intel® IXP4XX Product Line of Network Processors. This capacitor value provides both bypass and filtering.

When 170 nF is an inconvenient size, capacitor values between 150 nF to 200 nF could be used with little adverse effects, assuming that the effective series resistance of the 200-nF capacitor is under 50 mΩ.

In order to achieve a 200-nF capacitance, a parallel combination of two 100-nF capacitors may be used as long as the capacitors are placed directly beside each other. V_{SSP_OSC} consists of two pins, AD10 and AF10. Ensure that both pins are connected as shown in Figure 10.

Figure 10. V_{CCOSCP} Power Filtering Diagram



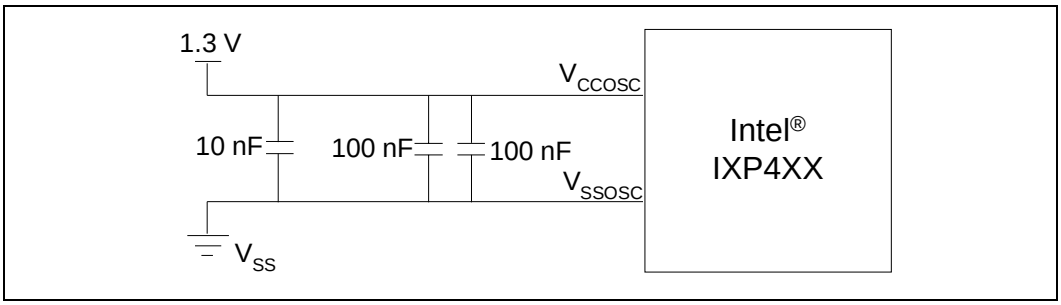
5.2.4 V_{CCOSC} Requirement

A parallel combination of a 10-nF capacitor — for bypass — and a 200-nF capacitor — for a first-order filter with a cut-off frequency below 33 MHz — must be connected to both of the V_{CCOSC} pins of the Intel® IXP4XX Product Line of Network Processors.

The grounds of both capacitors should be connected to the V_{SSOSC} supply pin. Both capacitors should be located less than 0.5 inch away from the V_{CCOSC} pin and the associated V_{SSOSC} pin.

In order to achieve a 200-nF capacitance, a parallel combination of two 100-nF capacitors may be used as long as the capacitors are placed directly beside each other.

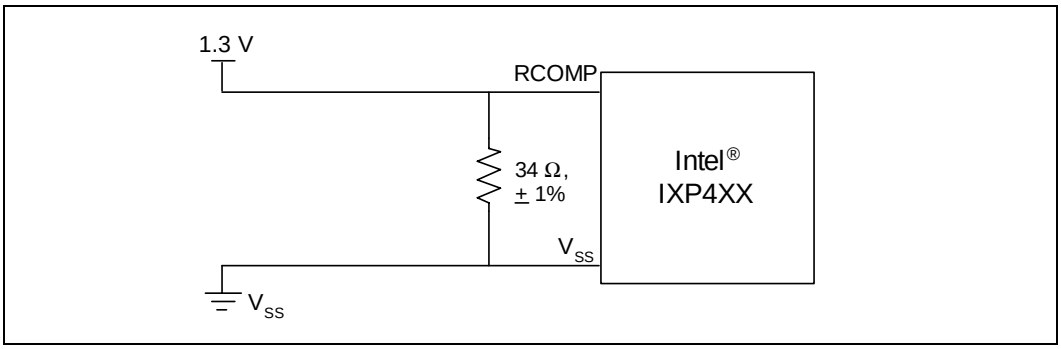
Figure 11. V_{CCOSC} Power Filtering Diagram



5.3 RCOMP Pin Requirements

Figure 12 shows the requirements for the RCOMP pin.

Figure 12. RCOMP Pin External Resistor Requirements



5.4 DC Specifications

5.4.1 Operating Conditions

Table 24. Operating Conditions

Symbol	Parameter	Min.	Typ.	Max.	Units	Notes
V_{CCP}	Voltage supplied to the I/O of the Intel® IXP4XX Product Line of Network Processors.	3.135	3.3	3.465	V	
V_{CC}	Voltage supplied to the internal logic of the Intel® IXP4XX Product Line of Network Processors.	1.235	1.3	1.365	V	
V_{CCOSC}	Voltage supplied to the internal oscillator logic of the Intel® IXP4XX Product Line of Network Processors.	1.235	1.3	1.365	V	
V_{CCOSCP}	Voltage supplied to the oscillator I/O of the Intel® IXP4XX Product Line of Network Processors.	3.135	3.3	3.465	V	
V_{CCPLL1}	Voltage supplied to the analog phase-lock loop of the Intel® IXP4XX Product Line of Network Processors.	1.235	1.3	1.365	V	
V_{CCPLL2}	Voltage supplied to the analog phase-lock loop of the Intel® IXP4XX Product Line of Network Processors.	1.235	1.3	1.365	V	

5.4.2 PCI DC Parameters

Table 25. PCI DC Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units	Notes
V_{IH}	Input-high voltage		0.5 V_{CCP}			V	3
V_{IL}	Input-low voltage				0.3 V_{CCP}	V	3
V_{OH}	Output-high voltage	I_{OUT} = -500 µA	0.9 V_{CCP}			V	3
V_{OL}	Output-low voltage	I_{OUT} = 1500 µA			0.1 V_{CCP}	V	3
I_{IL}	Input-leakage current	0 < V_{IN} < V_{CCP}	-10		10	µA	1, 3
C_{IN}	Input-pin capacitance			5		pF	2, 3
C_{OUT}	I/O or output pin capacitance			5		pF	2,3
C_{IDSEL}	IDSEL-pin capacitance			5		pF	2,3
L_{PIN}	Pin inductance			20		nH	2,3

NOTES:
1. Input leakage currents include hi-Z output leakage for all bidirectional buffers with tri-state outputs.
2. These values are typical values seen by the manufacturing process and are not tested.
3. For additional information, see the *PCI Local Bus Specification*, Revision 1.1.

5.4.3 USB v1.1 DC Parameters

Table 26. USB v1.1 DC Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units	Notes
V_{IH}	Input-high voltage		2.0			V	
V_{IL}	Input-low voltage				0.8	V	
V_{OH}	Output-high voltage	I_{OUT} = -6.1 * V_{OH}mA	2.8			V	
V_{OL}	Output-low voltage	IOUT = 6.1 * V_{OH}mA			0.3	V	
I_{IL}	Input-leakage current	0 < V_{IN} < V_{CCP}	-10		10	µA	
C_{IN}	Input-pin capacitance			5		pF	1

NOTES:
1. These values are typical values seen by the manufacturing process and are not tested.

5.4.4 UTOPIA-2 DC Parameters

Table 27. UTOPIA-2 DC Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units	Notes
V_{IH}	Input-high voltage		2.0			V	
V_{IL}	Input-low voltage				0.8	V	
V_{OH}	Output-high voltage	I_{OUT} = -8 mA	2.4			V	
V_{OL}	Output-low voltage	I_{OUT} = 8 mA			0.5	V	
I_{OH}	Output current at high voltage	V_{OH} > 2.4 V	-8			mA	
I_{OL}	Output current at low voltage	V_{OL} < 0.5 V	8			mA	
I_{IL}	Input-leakage current	0 < V_{IN} < V_{CCP}	-10		10	µA	1
C_{IN}	Input-pin capacitance			5		pF	2
C_{OUT}	I/O or output pin capacitance			5		pF	2

NOTES:
1. Input leakage currents include hi-Z output leakage for all bidirectional buffers with tri-state outputs.
2. These values are typical values seen by the manufacturing process and are not tested.

5.4.5 MII/RMII DC Parameters

Table 28. MII/RMII DC Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units	Notes
V_{IH}	Input-high voltage		2.0			V	
V_{IL}	Input-low voltage				0.8	V	
V_{OH}	Output-high voltage	I_{OUT} = -4 mA	2.4			V	
V_{OL}	Output-low voltage	I_{OUT} = 4mA			0.4	V	
I_{OH}	Output current at high voltage	V_{OH} > 2.4 V	-8			mA	
I_{OL}	Output current at low voltage	V_{OL} < 0.4 V	8			mA	
I_{IL}	Input-leakage current	0 < V_{IN} < V_{CCP}	-10		10	µA	
C_{IN}	Input-pin capacitance			5		pF	1

NOTES:
1. These values are typical values seen by the manufacturing process and are not tested.

5.4.6 MDI DC Parameters

Table 29. MDI DC Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units	Notes
V_{IH}	Input-high voltage		2.0			V	
V_{IL}	Input-low voltage				0.8	V	
V_{OH}	Output-high voltage	I_{OUT} = -4 mA	2.4			V	
V_{OL}	Output-low voltage	I_{OUT} = 4 mA			0.4	V	
I_{IL}	Input-leakage current	0 < V_{IN} < V_{CCP}	-10		10	µA	
C_{IN}	Input-pin capacitance			5		pF	1
C_{INMDIO}	Input-pin capacitance			5		pF	1

NOTES:
1. These values are typical values seen by the manufacturing process and are not tested.

5.4.7 SDRAM Bus DC Parameters

Table 30. SDRAM Bus DC Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units	Notes
V_{IH}	Input-high voltage		2.0			V	1
V_{IL}	Input-low voltage				0.8	V	2
V_{OH}	Output-high voltage	I_{OUT} = -4 mA	2.4			V	
V_{OL}	Output-low voltage	I_{OUT} = 4 mA			0.4	V	
I_{IL}	Input-leakage current	$0 < V_{IN} < V_{CCP}$	-5		5	µA	
I_{OL}	Output-leakage current	$0 < V_{IN} < V_{CCP}$	-5		5	µA	
C_{INCLK}	Input-pin capacitance			4		pF	3
C_{IO}	I/O-pin capacitance			5		pF	3

NOTES:
1. V_{IH} overshoot: $V_{IH\ (MAX)} = V_{CCP} + 2$ V for a pulse width $\leq$ 3 ns, and the pulse width cannot be greater than one third of the cycle rate.
2. V_{IL} undershoot: $V_{IL\ (MIN)} = -2$ V for a pulse width $\leq$ 3 ns cannot be exceeded.
3. These values are typical values seen by the manufacturing process and are not tested.

5.4.8 Expansion Bus DC Parameters

Table 31. Expansion Bus DC Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units	Notes
V_{IH}	Input-high voltage		2.0			V	
V_{IL}	Input-low voltage				0.8	V	
V_{OH}	Output-high voltage	I_{OUT} = -4 mA	2.4			V	1
V_{OL}	Output-low voltage	I_{OUT} = 4mA			0.4	V	1
I_{IL}	Input-leakage current	$0 < V_{IN} < V_{CCP}$	-10		10	µA	
C_{IN}	Input-pin capacitance			5		pF	2

NOTES:
1. Test conditions were a 70 pF load to ground.
2. These values are typical values seen by the manufacturing process and are not tested.

5.4.9 High-Speed, Serial Interface 0 DC Parameters

Table 32. High-Speed, Serial Interface 0 DC Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units	Notes
V_{IH}	Input-high voltage		2.0			V	
V_{IL}	Input-low voltage				0.8	V	
V_{OH}	Output-high voltage	I_{OUT} = -8 mA	2.4			V	
V_{OL}	Output-low voltage	I_{OUT} = 8 mA			0.4	V	
I_{IL}	Input-leakage current	$0 < V_{IN} < V_{CCP}$	-10		10	µA	
C_{IN}	Input-pin capacitance			5		pF	1

NOTES:
1. These values are typical values seen by the manufacturing process and are not tested.

5.4.10 High-Speed, Serial Interface 1 DC Parameters

Table 33. High-Speed, Serial Interface 1 DC Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units	Notes
V_{IH}	Input-high voltage		2.0			V	
V_{IL}	Input-low voltage				0.8	V	
V_{OH}	Output-high voltage	I_{OUT} = -8 mA	2.4			V	
V_{OL}	Output-low voltage	I_{OUT} = 8 mA			0.4	V	
I_{IL}	Input-leakage current	$0 < V_{IN} < V_{CCP}$	-10		10	µA	
C_{IN}	Input-pin capacitance			5		pF	1

NOTES:
1. These values are typical values seen by the manufacturing process and are not tested.

5.4.11 High-Speed UART DC Parameters

Table 34. High-Speed UART DC Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units	Notes
V_{IH}	Input-high voltage		2.0			V	
V_{IL}	Input-low voltage				0.8	V	
V_{OH}	Output-high voltage	I_{OUT} = -4 mA	2.4			V	
V_{OL}	Output-low voltage	I_{OUT} = 4 mA			0.4	V	
I_{IL}	Input-leakage current	$0 < V_{IN} < V_{CCP}$	-10		10	µA	
C_{IN}	Input-pin capacitance			5		pF	1

NOTES:
1. These values are typical values seen by the manufacturing process and are not tested.

5.4.12 Console UART DC Parameters

Table 35. Console UART DC Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units	Notes
V_{IH}	Input-high voltage		2.0			V	
V_{IL}	Input-low voltage				0.8	V	
V_{OH}	Output-high voltage	I_{OUT} = -4 mA	2.4			V	
V_{OL}	Output-low voltage	I_{OUT} = 4 mA			0.4	V	
I_{IL}	Input-leakage current	$0 < V_{IN} < V_{CCP}$	-10		10	µA	
C_{IN}	Input-pin capacitance			5		pF	1

NOTES:
1. These values are typical values seen by the manufacturing process and are not tested.

5.4.13 GPIO DC Parameters

Table 36. GPIO DC Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units	Notes
V_{IH}	Input-high voltage		2.0			V	
V_{IL}	Input-low voltage				0.8	V	
V_{OH}	Output-high voltage for GPIO 0 to GPIO 13	I_{OUT} = -16 mA	2.4			V	
V_{OL}	Output-low voltage for GPIO 0 to GPIO 13	I_{OUT} = 16 mA			0.4	V	
V_{OH}	Output-high voltage for GPIO 14 and GPIO 15	I_{OUT} = -4 mA	2.4			V	
V_{OL}	Output-low voltage for GPIO 14 and GPIO 15	I_{OUT} = 4 mA			0.4	V	
I_{IL}	Input-leakage current	$0 < V_{IN} < V_{CCP}$	-10		10	µA	
C_{IN}	Input-pin capacitance			5		pF	

5.4.14 JTAG DC Parameters

Table 37. JTAG DC Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units	Notes
V_{IH}	Input-high voltage		2.0			V	
V_{IL}	Input-low voltage				0.8	V	
V_{OH}	Output-high voltage	I_{OUT} = -4 mA	2.4			V	
V_{OL}	Output-low voltage	I_{OUT} = 4 mA			0.4	V	
I_{IL}	Input-leakage current	$0 < V_{IN} < V_{CCP}$	-10		10	µA	
C_{IN}	Input-pin capacitance			5		pF	1

NOTES:
1. These values are typical values seen by the manufacturing process and are not tested.

5.4.15 Reset DC Parameters

Table 38. PWRON_Reset _N Parameters

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units	Notes
V_{IH}	Input-high voltage		1.0			V	
V_{IL}	Input-low voltage				0.3	V	

5.5 AC Specifications

5.5.1 Clock Signal Timings

5.5.1.1 Intel® IXP4XX Product Line of Network Processors Clock Timings

Table 39. Intel® IXP4XX Product Line of Network Processors Clock Timings

Symbol	Parameter	Min.	Nom.	Max.	Units	Notes
V_{IH}	Input-high voltage	2.0			V	
V_{IL}	Input-low voltage			0.8	V	
$T_{FREQUENCY}$	Clock frequency for Intel® IXP4XX Product Line of Network Processors crystal or oscillator.		33.33		MHz	1, 4
$\triangle_{FREQUENCY}$	Clock tolerance over -40º C to 85º C.	-50		50	ppm	
C_{IN}	Pin capacitance of Intel® IXP4XX Product Line of Network Processors' inputs.		5		pF	
C_{SHUNT}	C_{SHUNT} is a crystal parameter sometimes referred to as the holder capacitance.	2	3	4	pF	
C_1	Load capacitance				pF	2
C_2	Load capacitance				pF	2
T_{DC}	Duty cycle	35	50	65	%	3

NOTES:
1. This value could be an oscillator input or a series resonant frequency from a crystal. If used as an oscillator input, tie to the crystal input pin and leave the crystal output pin disconnected.
2. Use the component values recommended by the crystal manufacturer.
3. This parameter applies when driving the clock input with an oscillator.
4. The reference-clock input slope should not exceed more than 2.5 V/nS to ensure proper PLL operation.

Figure 13. Typical Connection to a Crystal

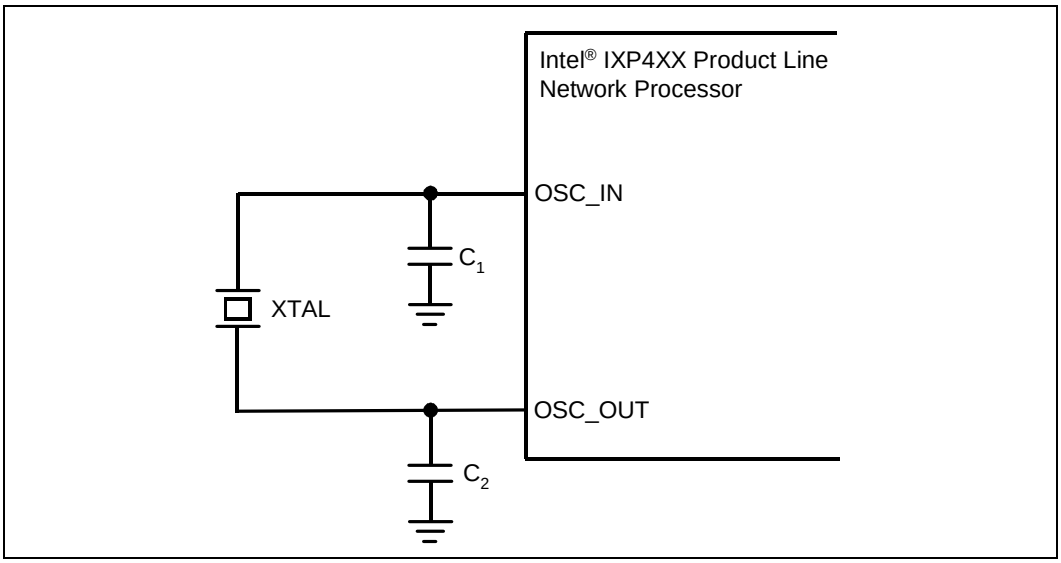
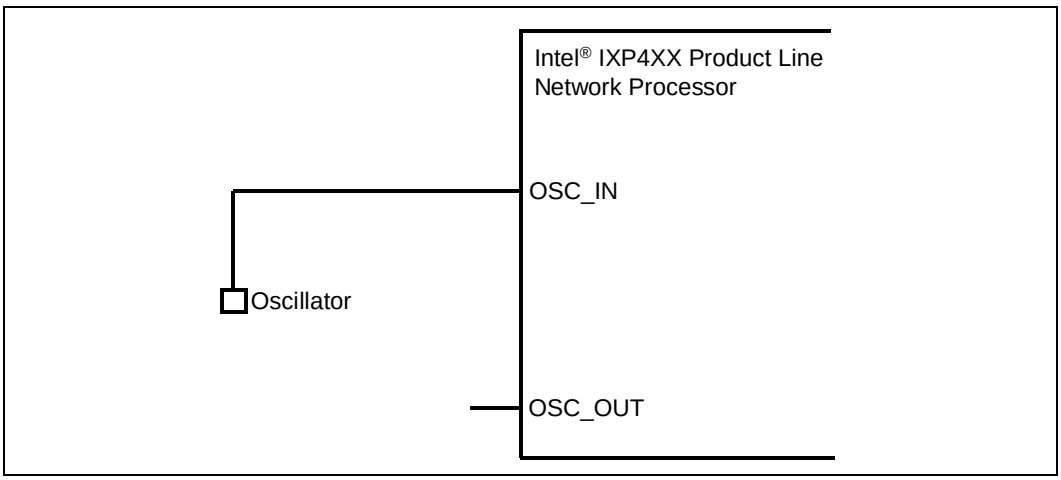


Figure 14. Typical Connection to an Oscillator



5.5.1.2 PCI Clock Timings

Table 40. PCI Clock Timings

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

5.5.1.3 MII/RMII Clock Timings

Table 41. MII/RMII Clock Timings

Symbol	Parameter	Min.	Nom.	Max.	Units	Notes
$T_{period100Mbit}$	Clock period for Tx and Rx clock when operating in RMII mode of operation		50	50	MHz	
$T_{period100Mbit}$	Clock period for Tx and Rx Ethernet clocks		25	25	MHz	
$T_{period10Mbit}$	Clock period for Tx and Rx Ethernet clocks		2.5	2.5	MHz	
T_{duty}	Duty cycle for Tx and Rx Ethernet clocks	35	50	65	%	
$T_{rise/fall}$	Rise and fall time requirements for Tx and Rx Ethernet clocks			2	ns	

5.5.1.4 UTOPIA-2 Clock Timings

Table 42. UTOPIA-2 Clock Timings

Symbol	Parameter	Min.	Nom.	Max.	Units	Notes
T_{period}	Clock period for Tx and Rx UTOPIA-2 clocks			33	MHz	
T_{duty}	Duty cycle for Tx and Rx UTOPIA-2 clocks	40	50	60	%	
$T_{rise/fall}$	Rise and fall time requirements for Tx and Rx UTOPIA-2 clocks			2	ns	

5.5.1.5 Expansion Bus Clock Timings

Table 43. Expansion Bus Clock Timings

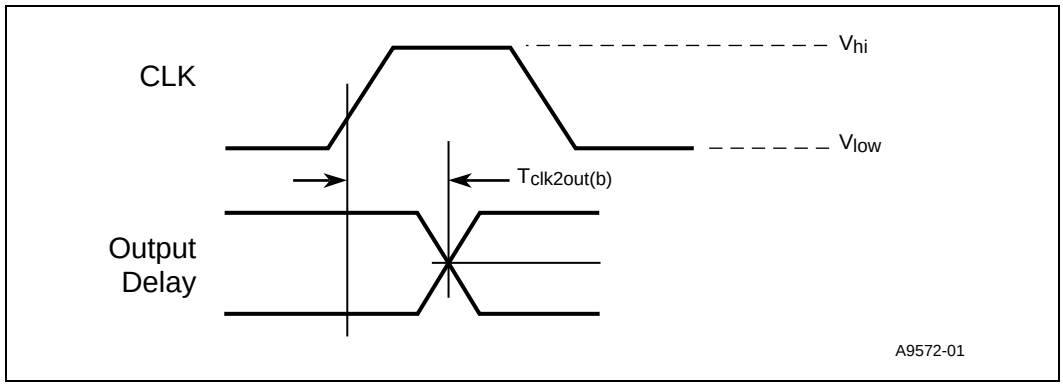
Symbol	Parameter	Min.	Nom.	Max.	Units	Notes
T_{period}	Clock period for expansion-bus clock			66	MHz	
T_{duty}	Duty cycle for expansion-bus clock	40	50	60	%	
$T_{rise/fall}$	Rise and fall time requirements for expansion-bus clock			2	ns	

5.5.2 Bus Signal Timings

The AC timing waveforms are shown in the following sections.

5.5.2.1 PCI

Figure 15. PCI Output Timing



NOTE: $V_{HI} = 0.6\ V_{CC}$ and $V_{LOW} = 0.2\ V_{CC}$

Figure 16. PCI Input Timing

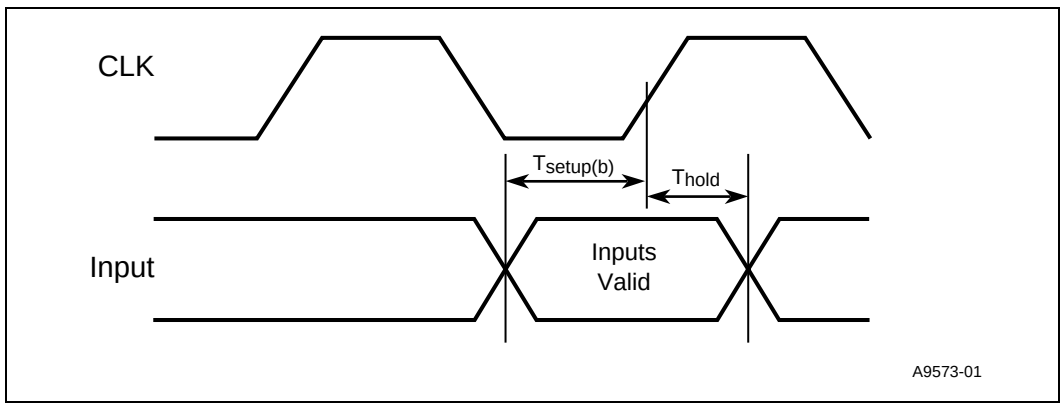


Table 44. PCI Bus Signal Timings

Symbol	Parameter	33 MHz		66 MHz		Units	Notes
		Min.	Max.	Min.	Max.		
$T_{clk2outb}$	Clock to output for all bused signals. This is the PCI_AD[31:0], PCI_CBE_N [3:0], PCI_PAR, PCI_FRAME_N, PCI_IRDY_N, PCI_TRDY_N, PCI_STOP_N, PCI_DEVSEL_N, PCI_PERR_N, PCI_SERR_N	2	11	1	6	ns	1, 2, 5, 7, 8
$T_{clk2out}$	Clock to output for all point-to-point signals. This is the PCI_GNT_N and PCI_REQ_N(0) only.	2	12	1	6	ns	1, 2, 5, 7, 8
T_{setupb}	Input setup time for all bused signals. This is the PCI_AD[31:0], PCI_CBE_N [3:0], PCI_PAR, PCI_FRAME_N, PCI_IRDY_N, PCI_TRDY_N, PCI_STOP_N, PCI_DEVSEL_N, PCI_PERR_N, PCI_SERR_N	7		3		ns	4, 6, 7, 8
T_{setup}	Input setup time for all point-to-point signals. This is the PCI_REQ_N and PCI_GNT_N(0) only.	10, 12		5		ns	4, 7, 8
T_{hold}	Input hold time from clock.	0		0		ns	4, 7, 8
$T_{rst-off}$	Reset active-to-output float delay		40		40	ns	5, 6, 7, 8

NOTES:
1. See the timing measurement conditions.
2. Parts compliant to the 3.3 V signaling environment.
3. REQ# and GNT# are point-to-point signals and have different output valid delay and input setup times than do bused signals. GNT# has a setup of 10 ns for 33 MHz and 5 ns for 66 MHz; REQ# has a setup of 12 ns for 33 MHz and 5 ns for 66 MHz.
4. RST# is asserted and de-asserted asynchronously with respect to CLK.
5. All PCI outputs must be asynchronously driven to a tri-state value when RST# is active.
6. Setup time applies only when the device is not driving the pin. Devices cannot drive and receive signals at the same time.
7. Timing was tested with a 70-pF capacitor to ground.
8. For additional information, see the *PCI Local Bus Specification*, Revision 1.1.

5.5.2.2 USB V1.1 Interface

For timing parameters, see the USB v 1.1 specification. The Intel® IXP4XX Product Line of Network Processors' USB v 1.1 interface is a device or function controller only. The Intel® IXP4XX Product Line of Network Processors' USB v 1.1 interface cannot be line-powered.

5.5.2.3 UTOPIA-2

Figure 17. UTOPIA-2 Input Timings

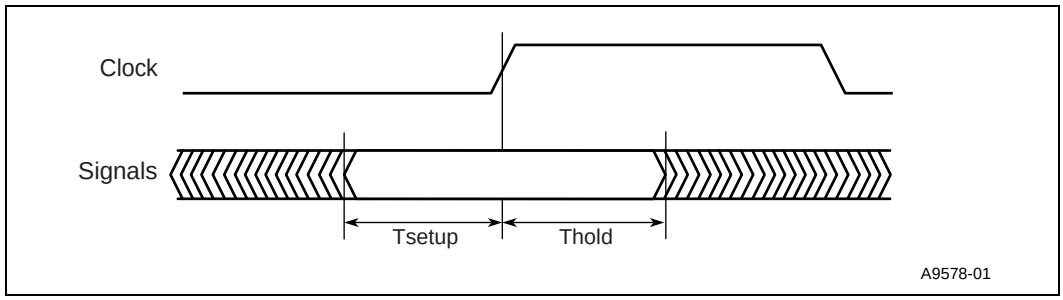


Table 45. UTOPIA-2 Input Timings Values

Symbol	Parameter	Min.	Max.	Units	Notes
T_{setup}	Input setup prior to rising edge of clock. Inputs included in this timing are UTP_IP_DATA[7:0], UTP_IP_SOC, AND UTP_IP_FCI, and UTP_OP_FCI.	8		ns	
T_{hold}	Input hold time after the rising edge of the clock. Inputs included in this timing are UTP_IP_DATA[7:0], UTP_IP_SOC, and UTP_IP_FCI, and UTP_OP_FCI.	1		ns	

Figure 18. UTOPIA-2 Output Timings

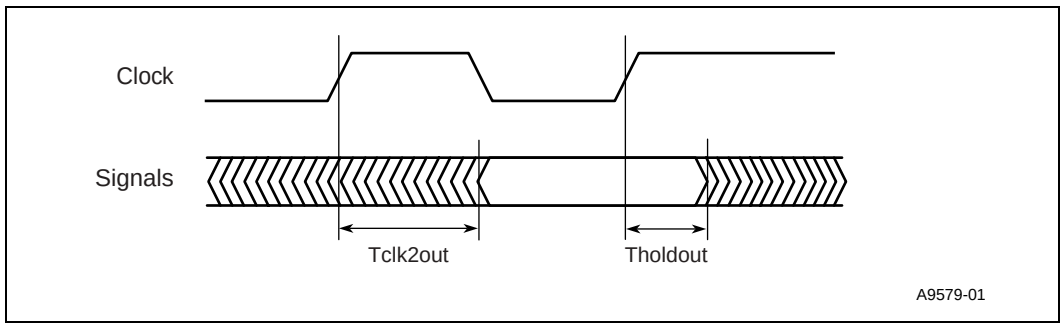


Table 46. UTOPIA-2 Output Timings Values

Symbol	Parameter	Min.	Max.	Units	Notes
$T_{clk2out}$	Rising edge of clock to signal output. Outputs included in this timing are UTP_IP_DATA[3:0], UTP_OP_SOC, UTP_OP_FCO, UTP_IP_FCO, UTP_OP_DATA[7:0], and UTP_OP_ADDR[3:0].		17	ns	1
$T_{holdout}$	Signal output hold time after the rising edge of the clock. Outputs included in this timing are UTP_IP_DATA[3:0], UTP_OP_SOC, UTP_OP_FCO, UTP_IP_FCO, UTP_OP_DATA[7:0], and UTP_OP_ADDR[3:0].	1		ns	1

NOTES:
1. Timing was tested with a 70-pF capacitor to ground.

5.5.2.4 MII/RMII

Figure 19. MII/RMII Output Timings

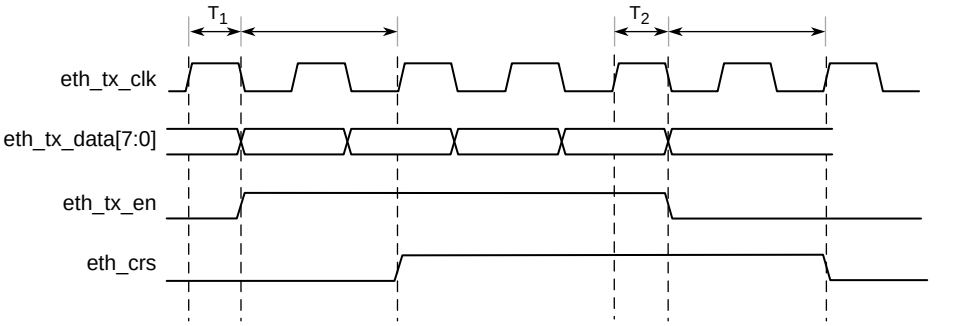


Table 47. MII/RMII Output Timings Values

Symbol	Parameter	Min.	Max.	Units	Notes
T_1	Clock to output delay for ETH_TXDATA and ETH_TXEN.	0	17	ns	1
T_2	ETH_TXDATA and ETH_TXEN hold time after ETH_TXCLK.	2		ns	

NOTES:
1. These values satisfy the RMII timing requirements, therefore satisfying the MII specification requirement of 0 ns to 25 ns clock to output delay.

Figure 20. MII/RMII Input Timings

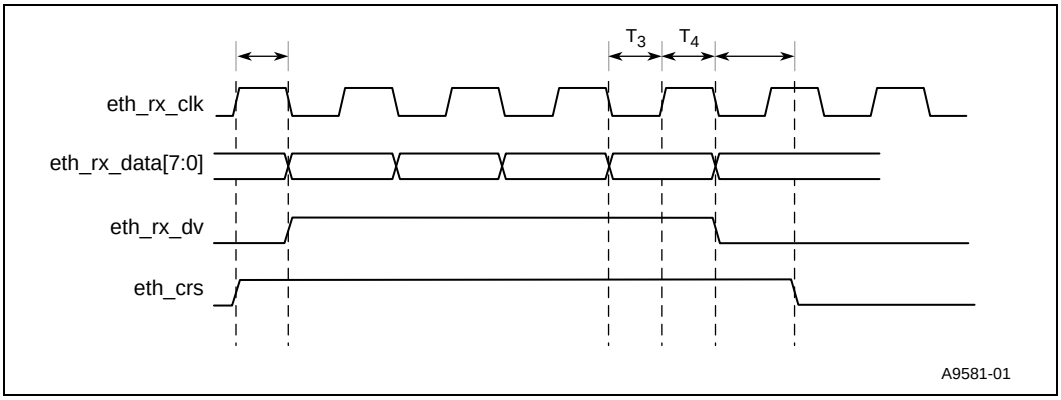


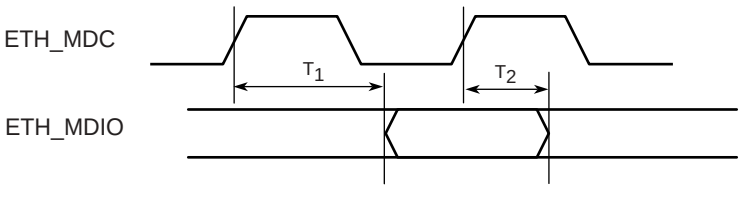
Table 48. MII/RMII Input Timings Values

Symbol	Parameter	Min.	Max.	Units	Notes
T_3	ETH_RXDATA and ETH_RXDV setup time prior to rising edge of ETH_RXCLK	5.5		ns	1, 2
T_4	ETH_RXDATA and ETH_RXDV hold time after the rising edge of ETH_RXCLK	0		ns	1, 2

NOTES:
1. These values meet the RMII timing requirements, satisfying the 10-ns setup and hold time requirements necessary for the MII specification.
2. Timing tests were performed with a 70-pF capacitor to ground.

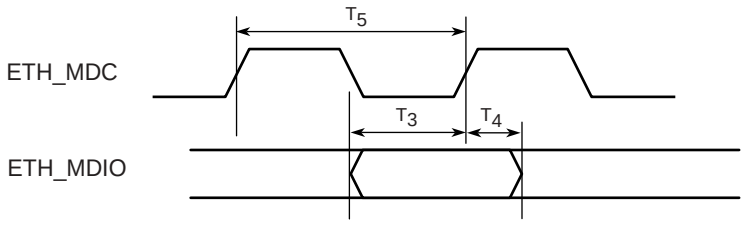
5.5.2.5 MDIO

Figure 21. MDIO Output Timings



NOTE: Processor is Sourcing MDIO.

Figure 22. MDIO Input Timings



NOTE: PHY is sourcing MDIO.

Table 49. MDIO Timings Values

Symbol	Parameter	Min.	Max.	Units	Notes
T1	ETH_MDIO, clock to output timing with respect to rising edge of ETH_MDC clock		ETH_MDC/2 + 10 ns	ns	
T2	ETH_MDIO output hold timing after the rising edge of ETH_MDC clock	10		ns	
T3	ETH_MDIO input setup prior to rising edge of ETH_MDC clock	2		ns	
T4	ETH_MDIO hold time after the rising edge of ETH_MDC clock	0		ns	
T5	ETH_MDC clock period	125	500	ns	

5.5.2.6 SDRAM Bus

Figure 23. SDRAM Input Timings

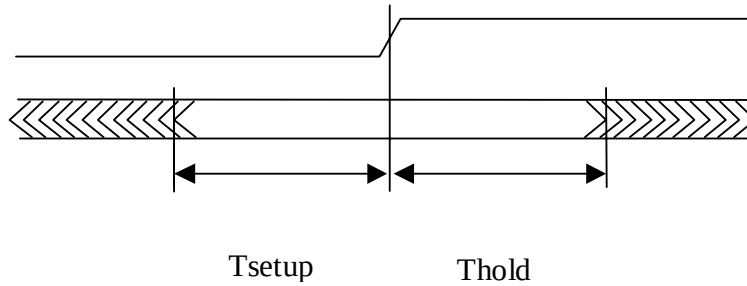


Table 50. SDRAM Input Timings Values

Symbol	Parameter	Min.	Max.	Units	Notes
T_{setup}	Input setup prior to rising edge of clock. Inputs included in this timing are SDM_DQ[31:0] (during a read operation).	1.4		ns	
T_{hold}	Input hold time after the rising edge of the clock. Inputs included in this timing are SDM_DQ[31:0] (during a read operation).	1.5		ns	

Figure 24. SDRAM Output Timings

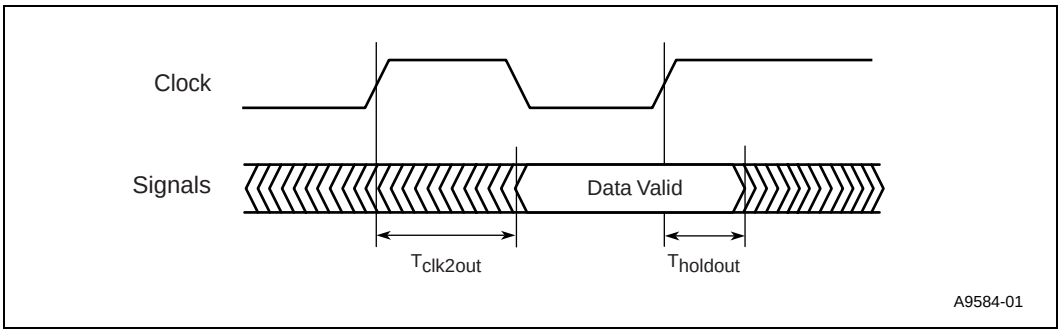


Table 51. SDRAM Output Timings Values

Symbol	Parameter	Min.	Max.	Units	Notes
$T_{clk2out}$	Rising edge of clock-to-signal output. Outputs included in this timing are SDM_ADDR[12:0], SDM_BA[1:0], SDM_DQM[3:0], SDM_CKE, SDM_WE_N, SDM_CS_N[1:0], SDM_CAS_N, SDM_RAS_N, SDM_DQ[31:0] (during a write operation).		5.5	ns	1
$T_{holdout}$	Signal output hold time after the rising edge of the clock. Outputs included in this timing are SDM_DQ[31:0] (during a write operation).	1.5		ns	1

NOTES:
1. Timing test were performed with a 70-pF load to ground.

5.5.2.7 Expansion Bus

Figure 25. Intel Multiplexed Mode

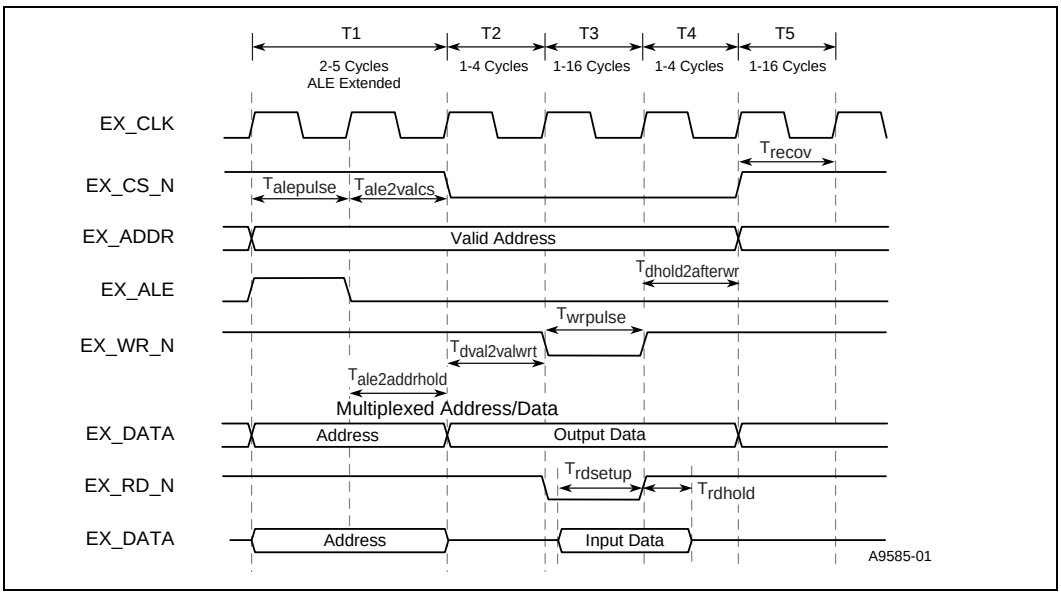


Table 52. Intel Multiplexed Mode Values

Symbol	Parameter	Min.	Max.	Units	Notes
Talepulse	Pulse width of ALE (ADDR is valid at the rising edge of ALE)	T	4T	ns	1, 7
Tale2addrhold	Valid address hold time after from falling edge of ALE	T	T	ns	1, 2, 7
Tdval2valwrt	Write data valid prior to WR_N falling edge	T	4T	ns	3, 7
Twrpulse	Pulse width of the WR_N	T	16T	ns	4, 7
Tdholdafterwr	Valid data after the rising edge of WR_N	T	4T	ns	5, 7
Tale2valcs	Valid chip select after the falling edge of ALE	T	4T	ns	7
Trdsetup	Data valid required before the rising edge of RD_N	3		ns	
Trdhold	Data hold required after the rising edge of RD_N	0		ns	
Trecov	Time needed between successive accesses on expansion interface.	T	16T	ns	6

NOTES:
1. The EX_ALE signal is extended form T to 4T nnsec based on the programming of the T1 timing parameter. The parameter Tale2addrhold is fixed at T.
2. Setting the address phase parameter (T1) will adjust the duration that the address appears to the external device.
3. Setting the data setup phase parameter (T2) will adjust the duration that the data appears prior to a data strobe (read or write) to an external device.
4. Setting the data strobe phase parameter (T3) will adjust the duration that the data strobe appears (read or write) to an external device. Data will be available during this time as well.
5. Setting the data hold strobe phase parameter (T4) will adjust the duration that the chip selects, address, and data (during a write) will be held.
6. Setting the recovery phase parameter (T5) will adjust the duration between successive accesses on the expansion interface.
7. T is the period of the clock measured in ns.
8. Clock to output delay for all signals will be a maximum of 15 ns for devices requiring operation in synchronous mode.
9. Timing tests were performed with a 70-pF capacitor to ground.

Figure 26. Intel Simplex Mode

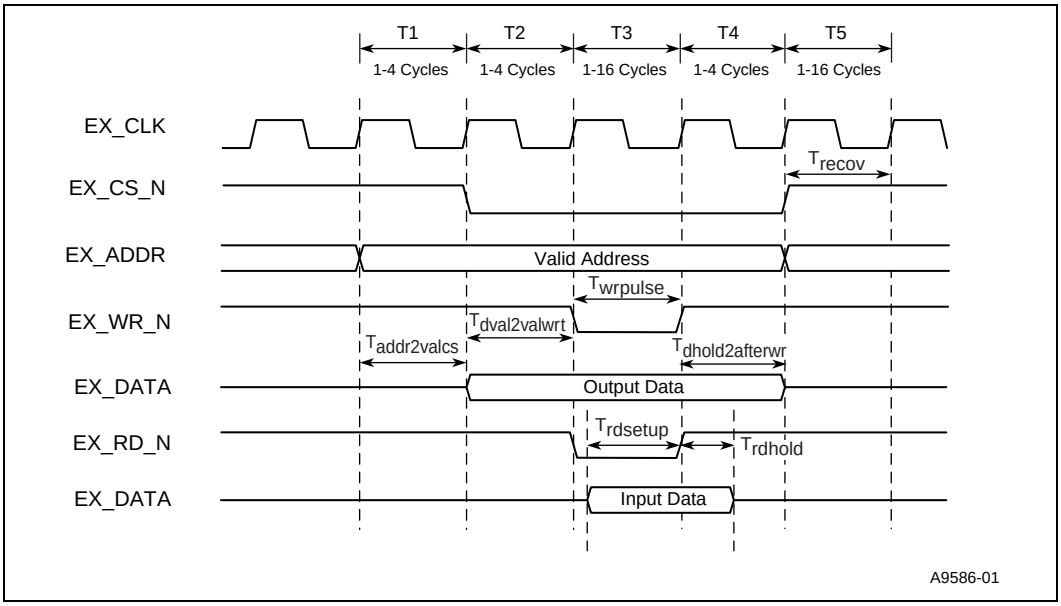


Table 53. Intel Simplex Mode Values

Symbol	Parameter	Min.	Max.	Units	Notes
$T_{addr2valcs}$	Valid address to valid chip select	T	4T	ns	1, 2, 7
$T_{dval2valwrt}$	Write data valid prior to EXPB_IO_WRITE_N falling edge	T	4T	ns	3, 7
$T_{wrpulse}$	Pulse width of the EXP_IO_WRITE_N	T	16T	ns	4, 7
$T_{dholdafterwr}$	Valid data after the rising edge of EXPB_IO_WRITE_N	T	4T	ns	5, 7
$T_{rdsetup}$	Data valid required before the rising edge of EXP_IO_READ_N	3		ns	
T_{rdhold}	Data hold required after the rising edge of EXP_IO_READ_N	0		ns	
T_{recov}	Time required between successive accesses on the expansion interface.	T	16T	ns	6

NOTES:
1. EX_ALE is not valid in simplex mode of operation.
2. Setting the address phase parameter (T1) will adjust the duration that the address appears to the external device.
3. Setting the data setup phase parameter (T2) will adjust the duration that the data appears prior to a data strobe (read or write) to an external device.
4. Setting the data strobe phase parameter (T3) will adjust the duration that the data strobe appears (read or write) to an external device. Data will be available during this time as well.
5. Setting the data hold strobe phase parameter (T4) will adjust the duration that the chip selects, address, and data (during a write) will be held.
6. Setting the recovery phase parameter (T5) will adjust the duration between successive accesses on the expansion interface.
7. T is the period of the clock measured in ns.
8. Clock to output delay for all signals will be a maximum of 15 ns for devices requiring operation in synchronous mode.
9. Timing tests were performed with a 70-pF capacitor to ground.

Figure 27. Motorola* Multiplexed Mode

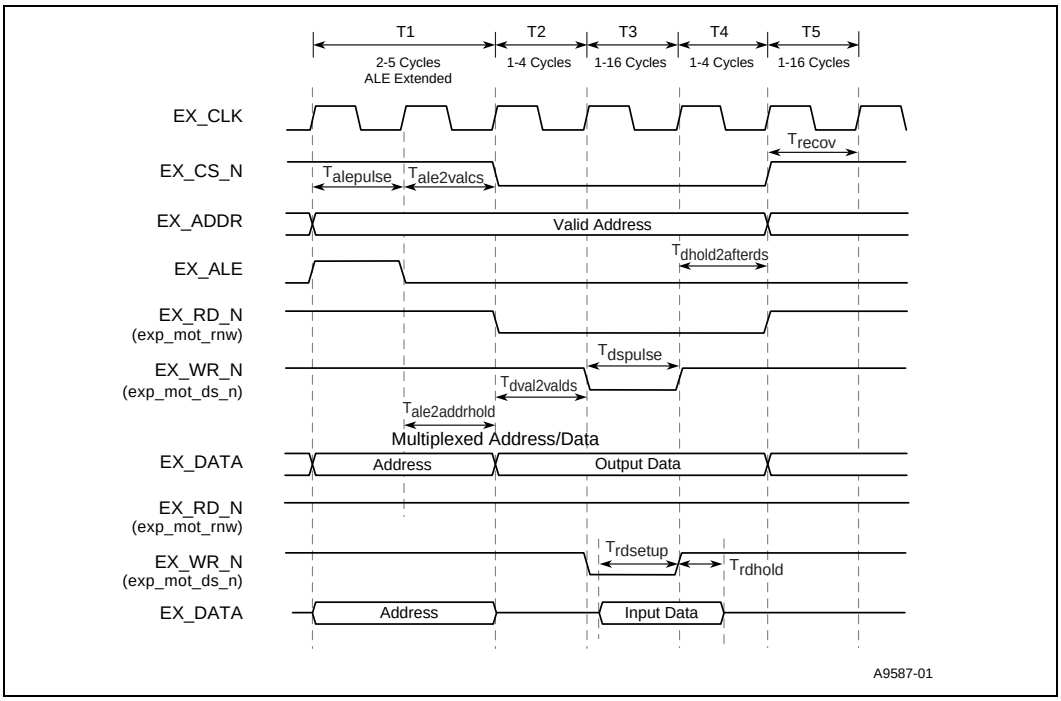


Table 54. Motorola* Multiplexed Mode Values

Symbol	Parameter	Min.	Max.	Units	Notes
$T_{alepulse}$	Pulse width of ALE (ADDR is valid at the rising edge of ALE)	T	4T	ns	1, 7
$T_{ale2addrhold}$	Valid address hold time after from falling edge of ALE	T	T	ns	1, 2, 7
$T_{dval2valds}$	Write data valid prior to EXP_MOT_DS_N falling edge	T	4T	ns	3, 7
$T_{dspulse}$	Pulse width of the EXP_MOT_DS_N	T	16T	ns	4, 7
$T_{dholdafterds}$	Valid data after the rising edge of EXP_MOT_DS_N	T	4T	ns	5, 7
$T_{ale2valcs}$	Valid chip select after the falling edge of ALE	T	4T	ns	7
$T_{rdsetup}$	Data valid required before the rising edge of EXP_MOT_DS_N	3		ns	
T_{rdhold}	Data hold required after the rising edge of EXP_MOT_DS_N	0		ns	
T_{recov}	Time needed between successive accesses on expansion interface.	T	16T	ns	6

NOTES:
1. The EX_ALE signal is extended form T to 4T nnsec, based on the programming of the T1 timing parameter. The parameter Tale2addrhold is fixed at T.
2. Setting the address phase parameter (T1) will adjust the duration that the address appears to the external device.
3. Setting the data setup phase parameter (T2) will adjust the duration that the data appears prior to a data strobe (read or write) to an external device.
4. Setting the data strobe phase parameter (T3) will adjust the duration that the data strobe appears (read or write) to an external device. Data will be available during this time as well.
5. Setting the data hold strobe phase parameter (T4) will adjust the duration that the chip selects, address, and data (during a write) will be held.
6. Setting the recovery phase parameter (T5) will adjust the duration between successive accesses on the expansion interface.
7. T is the period of the clock measured in ns.
8. Clock to output delay for all signals will be a maximum of 15 ns for devices requiring operation in synchronous mode.
9. Timing tests were performed with a 70-pF capacitor to ground.

Figure 28. Motorola* Simplex Mode

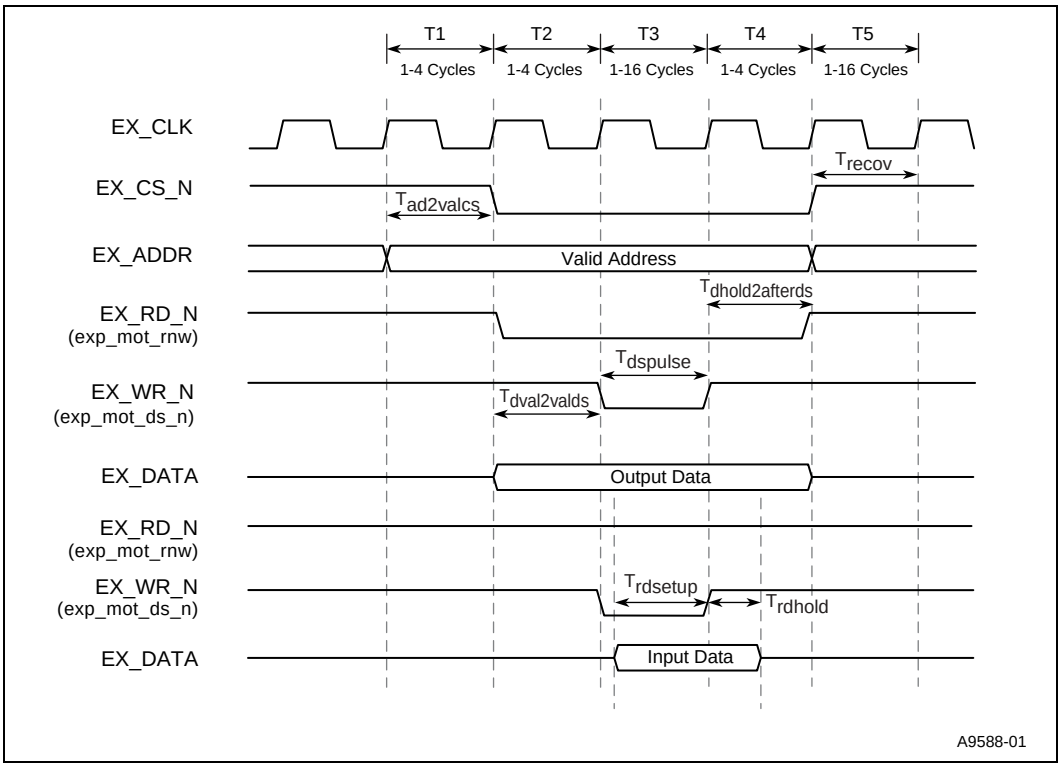


Table 55. Motorola* Simplex Mode Values

Symbol	Parameter	Min.	Max.	Units	Notes
$T_{ad2valcs}$	Valid address to valid chip select	T	4T	ns	1, 2, 7
$T_{dval2valds}$	Write data valid prior to EXP_MOT_DS_N falling edge	T	4T	ns	3, 7
$T_{dspulse}$	Pulse width of the EXP_MOT_DS_N	T	16T	ns	4, 7
$T_{dholdafterds}$	Valid data after the rising edge of EXP_MOT_DS_N	T	4T	ns	5, 7
$T_{rdsetup}$	Data valid required before the rising edge of EXP_MOT_DS_N	3		ns	
T_{rdhold}	Data hold required after the rising edge of EXP_MOT_DS_N	0		ns	
T_{recov}	Time required between successive accesses on the expansion interface.	T	16T	ns	6

NOTES:
1. EX_ALE is not valid in simplex mode of operation.
2. Setting the address phase parameter (T1) will adjust the duration that the address appears to the external device.
3. Setting the data setup phase parameter (T2) will adjust the duration that the data appears prior to a data strobe (read or write) to an external device.
4. Setting the data strobe phase parameter (T3) will adjust the duration that the data strobe appears (read or write) to an external device. Data will be available during this time as well.
5. Setting the data hold strobe phase parameter (T4) will adjust the duration that the chip selects, address, and data (during a write) will be held.
6. Setting the recovery phase parameter (T5) will adjust the duration between successive accesses on the expansion interface.
7. T is the period of the clock measured in ns.
8. Clock to output delay for all signals will be a maximum of 15 ns for devices requiring operation in synchronous mode.
9. Timing tests were performed with a 70-pF capacitor to ground.

Figure 29. HPI – 8 Mode Write Accesses

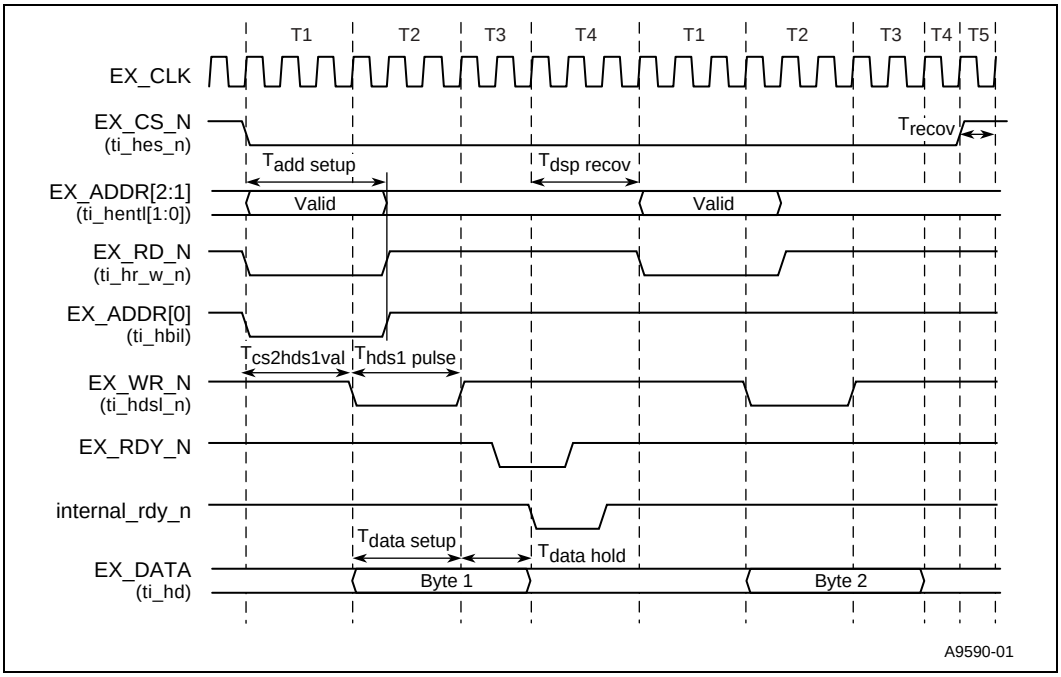


Table 56. HPI – 8 Mode Write Accesses Values

Symbol	Parameter	Min.	Max.	Units	Notes
T_{add_setup}	Valid time that address is asserted on the line. The address is asserted at the same time as chip select.	3T	5T	ns	1, 2, 8
$T_{cs2hds1val}$	Delay from chip select being active and the HDS1 data strobe being active.	3T	4T	ns	8
T_{hds1_pulse}	Pulse width of the HDS1 data strobe	2T	4T	ns	3, 7
T_{data_setup}	Data valid prior to the rising edge of the HDS1 data strobe.	3T	4T	ns	4, 8
T_{data_hold}	Data valid after the rising edge of the HDS1 data strobe.	2T	16T	ns	5, 8
T_{dsp_recov}	Recovery time between successive writes to the DSP.	T	4T	ns	6, 8
T_{recov}	Time required between successive accesses on the expansion interface.	T	16T	ns	7, 8

NOTES:
1. The address phase parameter (T1) must be set to a minimum value of 2. This value allows three T clocks for the address phase. This setting is required to ensure that in the event of an HRDY, the Intel® IXP4XX Product Line processor has had sufficient time to recognize the HRDY and hold the address phase for at least one clock pulse after the HRDY is de-activated.
2. Setting the address phase parameter (T1) will adjust the duration that the address appears to the external device.
3. The data-setup phase parameter (T2) must be set to a minimum value of 2. This value allows three T clocks for the address phase. This setting is required to ensure that in the event of an HRDY, the Intel® IXP4XX Product Line processor has had sufficient time to recognize the HRDY and hold the data setup phase for at least one clock pulse after the HRDY is de-activated.
4. Write data will always be valid at least one T clock cycle prior to the rising edge of the data strobe (HDS1) regardless of the any phase parameters. However, due to minimum timing requirements for the data setup phase parameter (T2) to operate in HPI mode, the minimum data valid prior to the data strobe will be at least three T clocks.
5. The data strobe phase parameter (T3) must be set to a minimum value of 1. This value allows two T clocks for the address phase. This setting is required to ensure that in the event of an HRDY, the Intel® IXP4XX Product Line processor has had sufficient time to recognize the HRDY and hold the data setup phase for at least one clock pulse after the HRDY is de-activated.
6. Setting the data strobe hold phase parameter (T4) will adjust the duration between successive accesses to the DSP. An inter-frame recovery time is required between each byte that is transferred.
7. Setting the recovery phase parameter (T5) will adjust the duration between successive accesses on the expansion interface.
8. T is the period of the clock measured in ns.
9. HRDY can be asserted by the DSP at any point in the access. The interface will delay the current transaction until HRDY is de-active.
10. Clock to output delay for all signals will be a maximum of 15 ns for devices requiring operation in synchronous mode.
11. Timing tests were performed with a 70-pF capacitor to ground.

Figure 30. HPI – 8 Mode Read Accesses

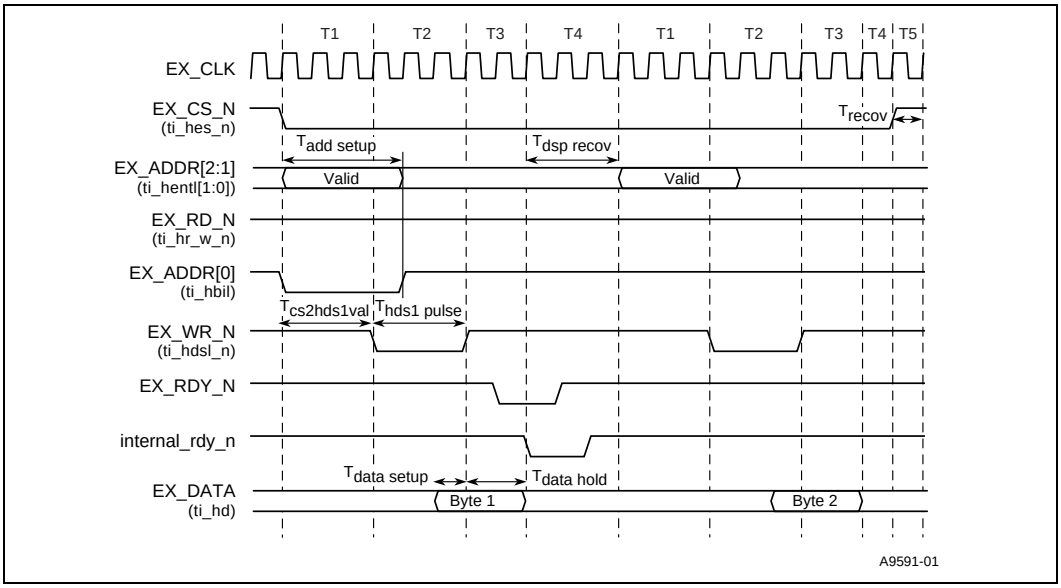
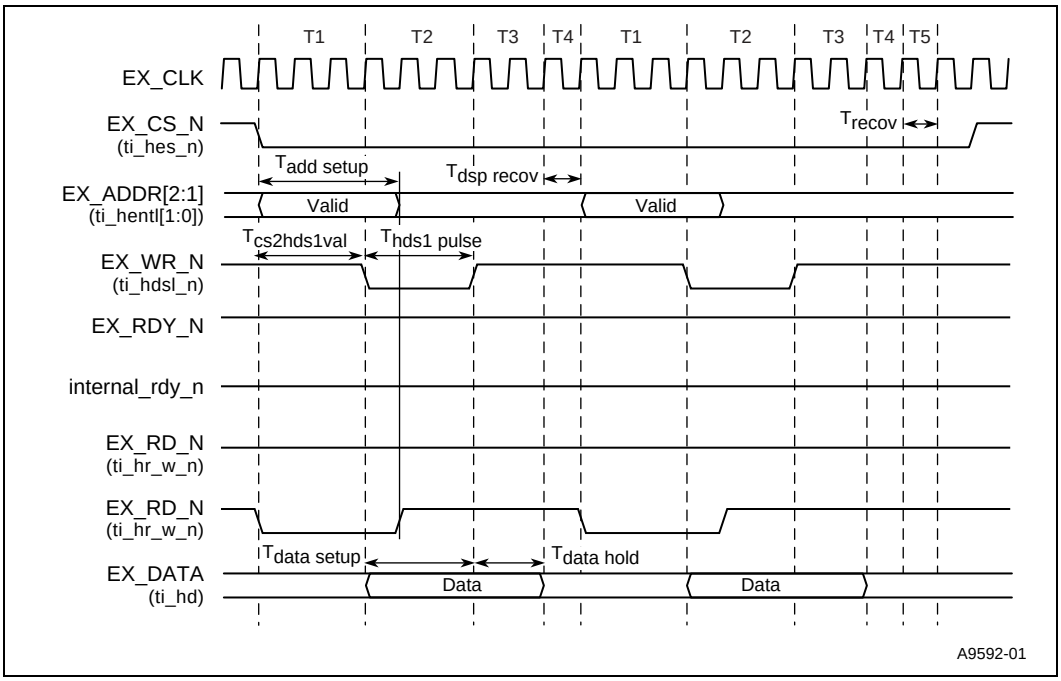


Table 57. HPI – 8 Mode Read Accesses Values

Symbol	Parameter	Min.	Max.	Units	Notes
T_{add_setup}	Valid time that address is asserted on the line. The address is asserted at the same time as chip select.	3T	5T	ns	1, 2, 8
$T_{cs2hds1val}$	Delay from chip select being active and the hds1 data strobe being active.	3T	4T	ns	8
T_{hds1_pulse}	Pulse width of the hds1 data strobe	2T	4T	ns	3, 7
T_{data_setup}	Data valid prior to the rising edge of the hds1 data strobe.	3		ns	4, 8
T_{data_hold}	Data valid after the rising edge of the hds1 data strobe.	0		ns	5, 8
T_{dsp_recov}	Recovery time between successive writes to the DSP.	T	4T	ns	6, 8
T_{recov}	Time required between successive accesses on the expansion interface.	T	16T	ns	7, 8

NOTES:
1. The address phase parameter (T1) must be set to a minimum value of 2. This value allows three T clocks for the address phase. This setting is required to ensure that in the event of an HRDY, the Intel® IXP4XX Product Line processor has had sufficient time to recognize the HRDY and hold the address phase for at least one clock pulse after the HRDY is de-active.
2. Setting the address phase parameter (T1) will adjust the duration that the address appears to the external device.
3. The data setup phase parameter (T2) must be set to a minimum value of 2. This value allows three T clocks for the address phase. This setting is required to ensure that in the event of an HRDY, the Intel® IXP4XX Product Line processor has had sufficient time to recognize the HRDY and hold the data setup phase for at least one clock pulse after the HRDY is de-active.
4. Read data must be valid at least 3 ns prior to the rising edge of the data strobe (HDS1).
5. Read data has no hold time requirement.
6. Setting the data strobe hold phase parameter (T4) will adjust the duration between successive accesses to the DSP. An inter-frame recover y time is required between each byte that is transferred.
7. Setting the recovery phase parameter (T5) will adjust the duration between successive accesses on the expansion interface.
8. T is the period of the clock measured in ns.
9. HRDY can be asserted by the DSP at any point in the access. The interface will delay the current transaction until HRDY is de-active.
10. Clock to output delay for all signals will be a maximum of 15 ns for devices requiring operation in synchronous mode.
11. Timing tests were performed with a 70-pF capacitor to ground.

Figure 31. TI HPI-16 Multiplexed Mode



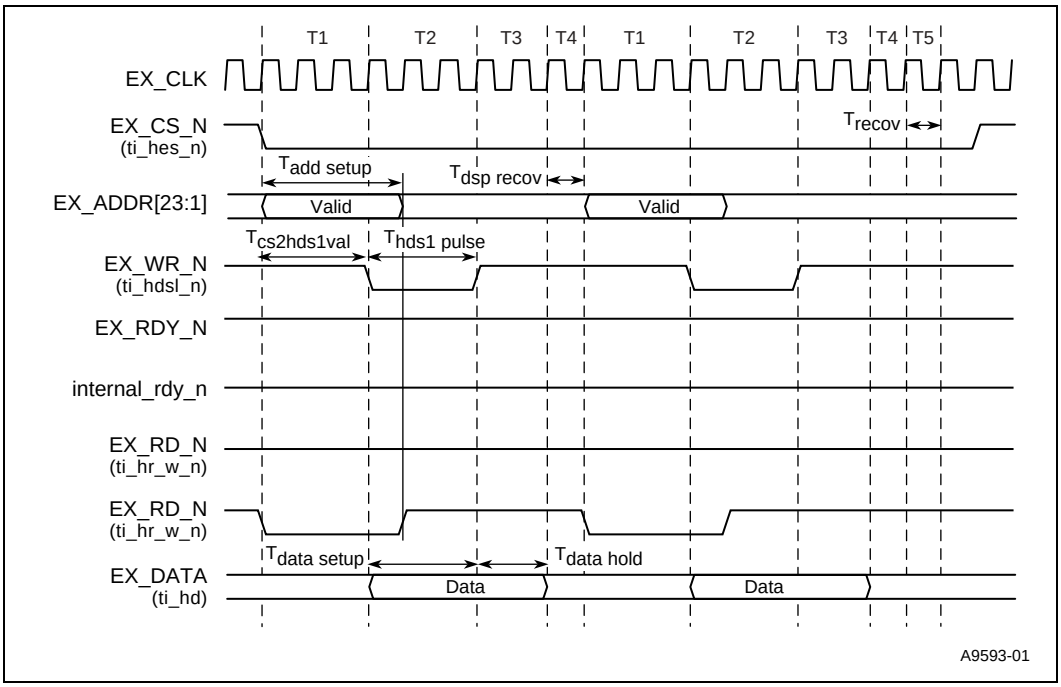
NOTE: EX_RD_N is used to produce the Texas Instruments* HR_W_N signal. A logic level high on this signal denotes a read access. A logic level low denotes a write access. In the figure, the first EX_RD_N signal is for a read access. The second EX_RD_N is for a write access. The EX_DATA signal applies to both read and write accesses.

Table 58. TI HPI-16 Multiplexed Mode Values

Symbol	Parameter	Min.	Max.	Units	Notes
T_{add_setup}	Valid time before the address is captured by the DSP. The address is asserted at the same time as chip select.	3T	5T	ns	1, 2, 8
$T_{cs2hds1val}$	Delay from chip select being active and the hds1 data strobe being active.	3T	4T	ns	8
T_{hds1_pulse}	Pulse width of the hds1 data strobe	2T	4T	ns	3, 7
T_{data_setup}	Read data valid prior to the rising edge of the hds1 data strobe.	3		ns	4, 8
T_{data_hold}	Read data valid after the rising edge of the hds1 data strobe	0		ns	5, 8
T_{data_setup}	Write data valid prior to the rising edge of the hds1 data strobe	3T	4T	ns	4, 8
T_{data_hold}	Write data valid after the rising edge of the hds1 data strobe	2T	16T	ns	5, 8
T_{dsp_recov}	Recovery time between successive writes to the DSP	T	4T	ns	6, 8
T_{recov}	Time required between successive accesses on the expansion interface	T	16T	ns	7, 8

NOTES:
1. The address phase parameter (T1) must be set to a minimum value of 2. This value allows three T clocks for the address phase. This setting is required to ensure that in the event of an HRDY, the Intel® IXP4XX Product Line processor has had sufficient time to recognize the HRDY and hold the address phase for at least one clock pulse after the hrdy is de-active.
2. Setting the address phase parameter (T1) will adjust the duration that the address appears to the external device.
3. The data setup phase parameter (T2) must be set to a minimum value of 2. This value allows three T clocks for the address phase. This setting is required to ensure that in the event of an HRDY, the Intel® IXP4XX Product Line processor has had sufficient time to recognize the HRDY and hold the data setup phase for at least one clock pulse after the HRDY is de-active.
4. Read data must be valid at least 3 ns prior to the rising edge of the data strobe (HDS1). Write data will be always valid at least one T clock cycle prior to the rising edge of the data strobe (HDS1) regardless of the any phase parameters. However, due to minimum timing requirements for the data setup phase parameter (T2) to operate in HPI mode the minimum data valid prior to the data strobe will be at least three T clocks.
5. Read data has no hold time requirement. For write data, the data strobe phase parameter (T3) must be set to a minimum value of 1. This value allows two T clocks for the address phase. This setting is required to ensure that in the event of an HRDY, the Intel® IXP4XX Product Line processor has had sufficient time to recognize the HRDY and hold the data setup phase for at least one clock pulse after the HRDY is de-active.
6. Setting the data strobe hold phase parameter (T4) will adjust the duration between successive accesses to the DSP. An inter-frame recover y time is required between each byte that is transferred.
7. Setting recovery-phase parameter (T5) will adjust duration between successive accesses on expansion interface.
8. T is the period of the clock measured in ns.
9. HRDY can be asserted by the DSP at any point in the access. The interface will delay the current transaction until HRDY is de-active.
10. Clock to output delay for all signals to be a maximum 15 s for devices requiring operation in synchronous mode.
11. Timing tests were performed with a 70-pF capacitor to ground.

Figure 32. TI HPI-16 Simplex Mode



NOTE: EX_RD_N is used to produce the Texas Instruments* HR_W_N signal. A logic level high on this signal denotes a read access. A logic level low denotes a write access. In the figure above, the first EX_RD_N signal is for a read access. The second EX_RD_N is f or a write access. The EX_DATA signal applies to both read and write accesses.

Table 59. TI HPI-16 Simplex Mode Values

Symbol	Parameter	Min.	Max.	Units	Notes
T_{add_setup}	Valid time before the address is captured by the DSP. The address is asserted at the same time as chip select.	3T	5T	ns	1, 2, 8
$T_{cs2hds1val}$	Delay from chip select being active and the hds1 data strobe being active	3T	4T	ns	8
T_{hds1_pulse}	Pulse width of the hds1 data strobe	2T	4T	ns	3, 7
T_{data_setup}	Read data valid prior to the rising edge of the hds1 data strobe.	3		ns	4, 8
T_{data_hold}	Read data valid after the rising edge of the hds1 data strobe	0		ns	5, 8
T_{data_setup}	Write data valid prior to the rising edge of the hds1 data strobe	3T	4T	ns	4, 8
T_{data_hold}	Write data valid after the rising edge of the hds1 data strobe	2T	16T	ns	5, 8
T_{dsp_recov}	Recovery time between successive writes to the DSP	T	4T	ns	6, 8
T_{recov}	Time required between successive accesses on the expansion interface	T	16T	ns	7, 8

NOTES:
1. The address phase parameter (T1) must be set to a minimum value of 2. This value allows three T clocks for the address phase. This setting is required to ensure that in the event of an HRDY, the Intel® IXP4XX Product Line processor has had sufficient time to recognize the HRDY and hold the address phase for at least one clock pulse after the HRDY is de-active.
2. Setting the address phase parameter (T1) will adjust the duration that the address appears to the external device.
3. The data setup phase parameter (T2) must be set to a minimum value of 2. This value allows three T clocks for the address phase. This setting is required to ensure that in the event of an HRDY, the Intel® IXP4XX Product Line processor has had sufficient time to recognize the HRDY and hold the data setup phase for at least one clock pulse after the HRDY is de-active.
4. Read data must be valid at least 3 ns prior to the rising edge of the data strobe (HDS1). Write data will be always valid at least one T clock cycle prior to the rising edge of the data strobe (HDS1) regardless of the any phase parameters. However, due to minimum timing requirements for the data setup phase parameter (T2) to operate in HPI mode, the minimum data valid prior to the data strobe will be at least three T clocks.
5. Read data has no hold time requirement. For write data, the data strobe phase parameter (T3) must be set to a minimum value of 1. This value allows two T clocks for the address phase. This setting is required to ensure that in the event of an HRDY, the Intel® IXP4XX Product Line processor has had sufficient time to recognize the HRDY and hold the data setup phase for at least one clock pulse after the HRDY is de-active.
6. Setting the data strobe hold phase parameter (T4) will adjust the duration between successive accesses to the DSP. An inter-frame recovery time is required between each byte that is transferred.
7. Setting the recovery phase parameter (T5) will adjust the duration between successive accesses on the expansion interface.
8. T is the period of the clock measured in ns.
9. HRDY can be asserted by the DSP at any point in the access. The interface will delay the current transaction until HRDY is de-active.
10. Clock to output delay for all signals will be a maximum of 15 ns for devices requiring operation in synchronous mode.
11. Timing tests were performed with a 70-pF capacitor to ground.

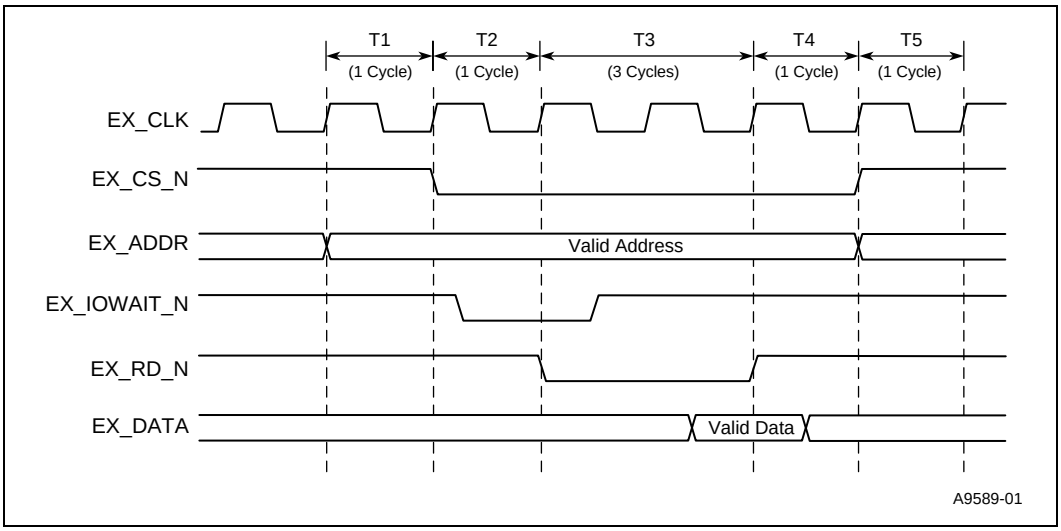
5.5.2.7.1 EX_IOWAIT_N

The EX_IOWAIT_N signal is available to be shared by devices attached to Chip Selects 0 through 7 and is used as required by slow devices.

If the external device asserts EX_IOWAIT_N during the strobe phase of a read transfer, the controller will hold in that phase until the EX_IOWAIT_N goes false. At that time, the controller will immediately transition to the hold phase regardless of the setting of the programming parameter (T3) for the strobe phase.

The EX_IOWAIT_N signal only affects the interface during the strobe phase of a read transfer. If Chip Selects 4 through 7 are configured in HPI mode of operation, each chip select will have a corresponding HRDY signal called EX_RDY. The polarity of the ready signal is programmable. Chip Select 4 corresponds to EX_RDY signal 0 and Chip Select 7 corresponds to EX_RDY signal 3.

Figure 33. Expansion Bus I/O Wait



NOTE: Notice that the access is an Intel-style simplex read access. The data strobe phase is set to a value to last three clock cycles. The data is returned from the peripheral device prior to the three clocks and the peripheral device de-asserts EX_IOWAIT_N. The data strobe phase terminates after two clocks even though the strobe phase was configured to pulse for three clocks.

5.5.2.8 High-Speed, Serial Interfaces

Figure 34. High-Speed, Serial Timings

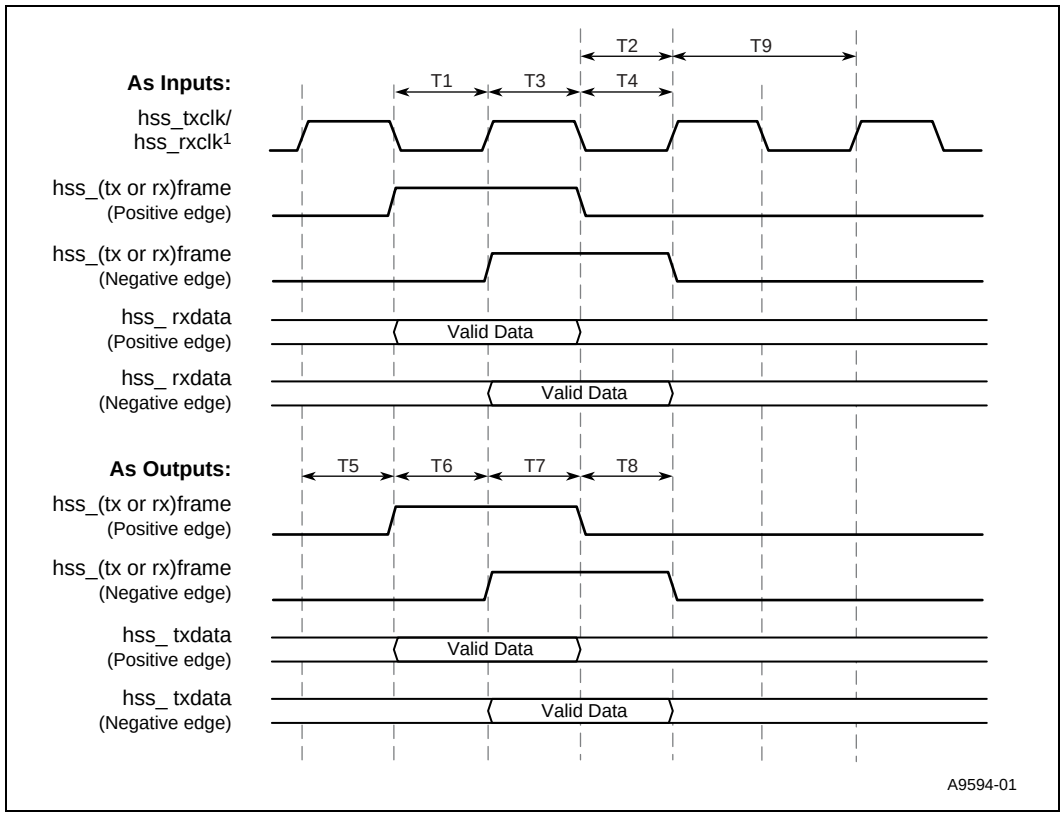


Table 60. High-Speed, Serial Timings Values

Symbol	Parameter	Min.	Max.	Units	Notes
T1	Setup time of HSS_TXFRAME, HSS_RXFRAME, and HSS_RXDATA prior to the rising edge of clock	5		ns	1, 2, 3
T2	Hold time of HSS_TXFRAME, HSS_RXFRAME, and HSS_RXDATA after the rising edge of clock	0		ns	1, 2, 3
T3	Setup time of HSS_TXFRAME, HSS_RXFRAME, and HSS_RXDATA prior to the falling edge of clock	5		ns	1, 2, 3
T4	Hold time of HSS_TXFRAME, HSS_RXFRAME, and HSS_RXDATA after the falling edge of clock	0		ns	1, 2, 3
T5	Rising edge of clock to output delay for HSS_TXFRAME, HSS_RXFRAME, and HSS_TXDATA		15	ns	1, 4
T6	Falling edge of clock to output delay for HSS_TXFRAME, HSS_RXFRAME, and HSS_TXDATA		15	ns	1, 3, 4
T7	Output Hold Delay after rising edge of final clock for HSS_TXFRAME, HSS_RXFRAME, and HSS_TXDATA	0		ns	1, 3, 4
T8	Output Hold Delay after falling edge of final clock for HSS_TXFRAME, HSS_RXFRAME, and HSS_TXDATA	0		ns	1,3, 4
T9	HSS_TXCLK period and HSS_RXCLK period	1/8.192 MHz	1/512 KHz	ns	5

NOTES:
1. HSS_TXCLK and HSS_RXCLK may be coming from external independent sources or being driven by the Intel® IXP4XX Product Line processor. The signals are shown to be synchronous for illustrative purposes and are not required to be synchronous.
2. Applicable when the HSS_RXFRAME and HSS_TXFRAME signals are being driven by an external source as inputs into the Intel® IXP4XX Product Line processor. Always applicable to HSS_RXDATA.
3. The HSS_RXFRAME and HSS_TXFRAME can be configured to accept data on the rising or falling edge of the given reference clock. HSS_RXFRAME and HSS_RXDATA signals are synchronous to HSS_RXCLK and HSS_TXFRAME and HSS_TXDATA signals are synchronous to the HSS_TXCLK.
4. Applicable when the HSS_RXFRAME and HSS_TXFRAME signals are being driven by the Intel® IXP4XX Product Line processor to an external source. Always applicable to HSS_TXDATA.
5. The HSS_TXCLK can be configured to be driven by an external source or be driven by the Intel® IXP4XX Product Line processor. The slowest clock speed that can be accepted or driven is 512 KHz. The maximum clock speed that can be accepted or driven is 8.192 MHz. The clock duty cycle accepted will be 50/50 + 20%.
6. Timing tests were performed with a 70-pF capacitor to ground and a 10-KΩ pull-up resistor.

5.5.2.9 JTAG

Figure 35. Boundary-Scan General Timings

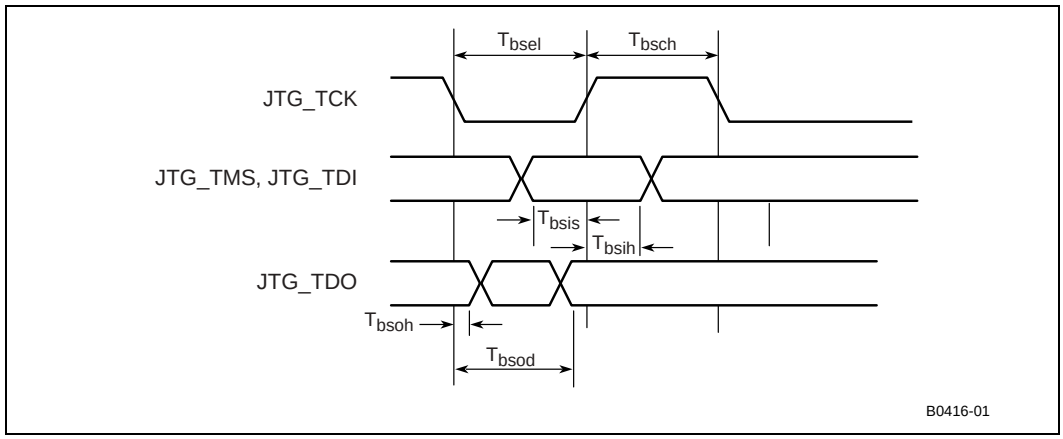


Figure 36. Boundary-Scan Reset Timings

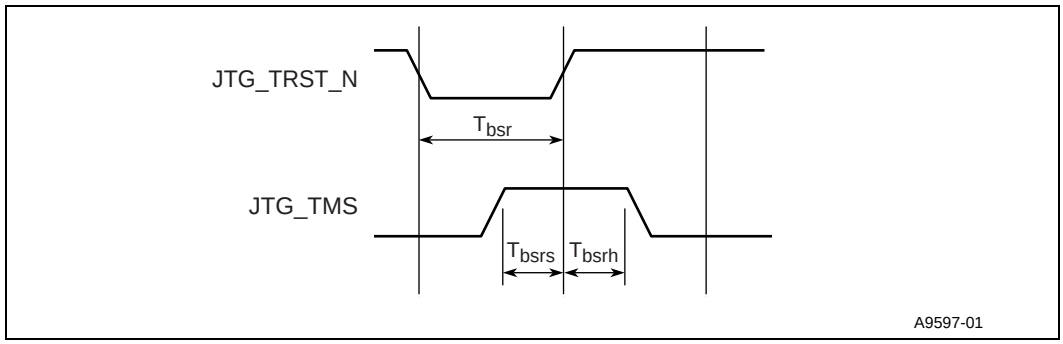


Table 61. Boundary-Scan Interface Timings Values

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units	Notes
T_{bscl}	JTAG_TCK low time		50			ns	2
T_{bsch}	JTAG_TCK high time		50			ns	2
T_{bsis}	JTAG_TDI, JTAG_TMS setup time to rising edge of JTAG_TCK		10			ns	
T_{bsih}	JTAG_TDI, JTAG_TMS hold time from rising edge of JTAG_TCK		10			ns	
T_{bsoh}	JTAG_TDO hold time after falling edge of JTAG_TCK		1.5			ns	1
T_{bsod}	JTAG_TDO clock to output from falling edge of JTAG_TCK				40	ns	1
T_{bsr}	JTAG_TRST_N reset period		30			ns	
T_{bsrs}	JTAG_TMS setup time to rising edge of JTAG_TRST_N		10			ns	
T_{bsrh}	JTAG_TMS hold time from rising edge of JTAG_TRST_N		10			ns	

NOTES:
1. Tests completed with a TBD pF load to ground on JTAG_TDO.
2. JTAG_TCK may be stopped indefinitely in either the low or high phase.

5.5.3 Reset Timings

Figure 37. Reset Timings

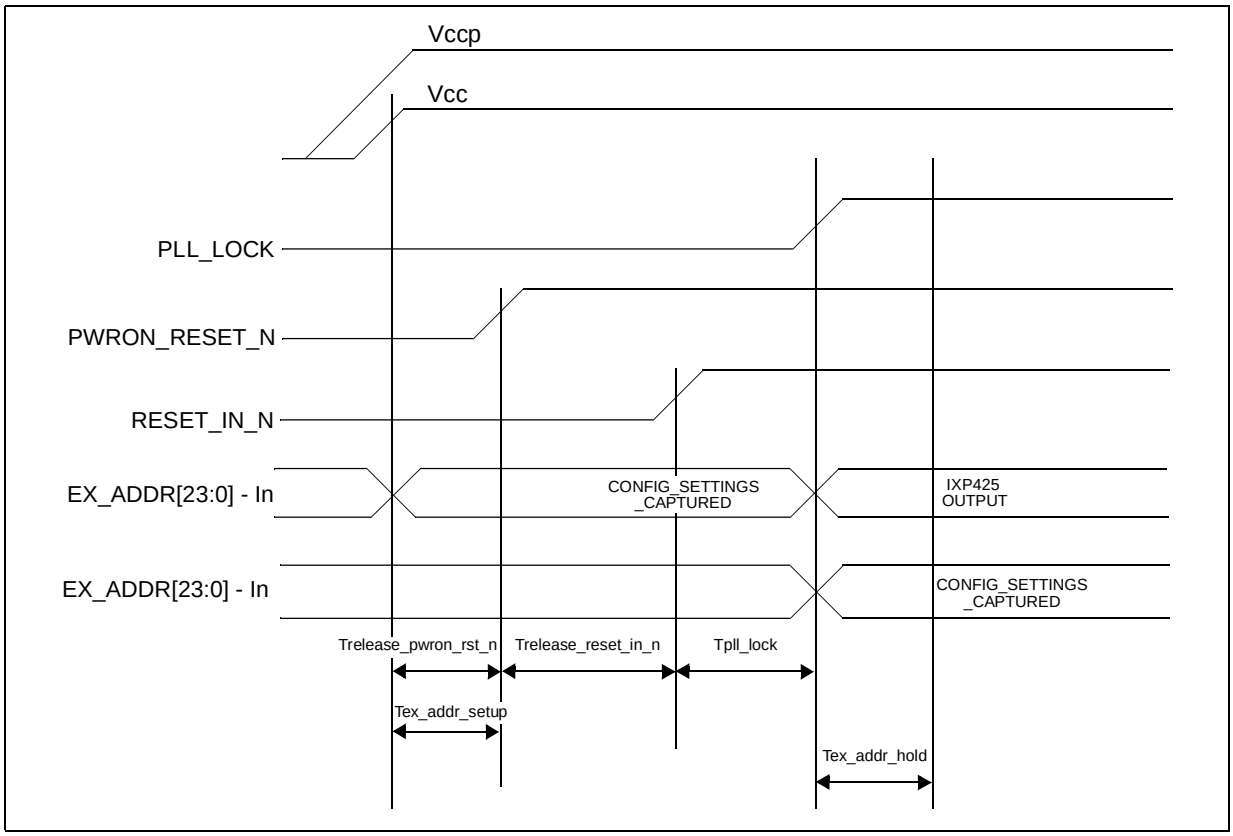


Table 62. Reset Timings Table Parameters

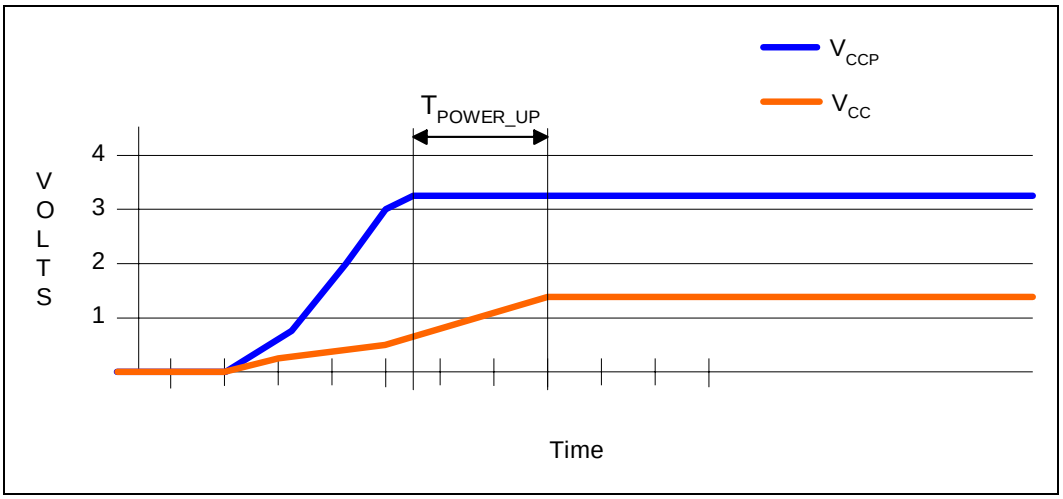
Symbol	Parameter	Min.	Typ.	Max.	Units	Note
$T_{RELEASE\_PWON\_RST\_N}$	Minimum time required to hold the PWON_RST_N at logic 0 state after stable power has been applied to the Intel® IXP4XX Product Line of Network Processors. When using a crystal to drive the Intel® IXP4XX Product Line of Network Processors system clock. (OSC_IN and OSC_OUT)	500			ms	1
$T_{RELEASE\_RESET\_IN\_N}$	Minimum time required to hold the RESET_IN_N at logic 0 state after PWON_RST_N has been released to a logic 1 state. The RESET_IN_N signal must be held low when the PWON_RST_N signal is held low.	10			ns	
T_{PLL_LOCK}	Maximum time for PLL_LOCK signal to drive to logic 1 after RESET_IN_N is driven to logic 1 state. The boot sequence does not occur until this period is complete.			10	µs	
$T_{EX\_ADDR\_SETUP}$	Minimum time for the EX_ADDR signals to drive the inputs prior to RESET_IN_N being driven to logic 1 state. This is used for sampling configuration information.	50			ns	2
$T_{EX\_ADDR\_HOLD}$	Minimum/maximum time for the EX_ADDR signals to drive the inputs prior to PLL_LOCK being driven to logic 1 state. This is used for sampling configuration information.	0		20	ns	2
T_{WARM_RESET}	Minimum time required to drive RESET_IN_N signal to logic 0 in order to cause a reset after the Intel® IXP4XX Product Line of Network Processors has been in normal operation. The power must remain stable and the PWON_RST_N signal must remain stable.	500			ns	

NOTES:
1. $T_{RELEASE\_PWRON\_RST\_N}$ is the time required for the internal oscillator to reach stability. When an external oscillator is being used in place of a crystal, the 500-ms delay is not required.
2. The expansion bus address is captured as a derivative of the RESET_IN_N signal going high. When a programmable-logic device is used to drive the EX_ADDR signals instead of pull-downs, the signals must be active until PLL_LOCK is active.

5.6 Power Sequence

The 3.3-V I/O voltage (V_{CCP}) must be powered up 1µs before the core voltage (V_{CC}). The Intel® IXP4XX Product Line of Network Processors' core voltage (V_{CC}) must never become stable prior to the 3.3-V I/O voltage (V_{CCP}). The V_{CCOSC}, V_{CCPLL1}, and V_{CCPLL2} voltages follow the V_{CC} power-up pattern. The V_{CCOSCP} follows the V_{CCP} power-up pattern. The value for T_{POWER_UP} must be at least 1 µs. The T_{POWER_UP} timing parameter is measured from V_{CCP} at 1.5V and V_{CC} at 0.5V

Figure 38. Power-up Sequence Timing



5.7 Power Dissipation

The following power assessments assume full speed operation by all peripherals and internal components. If applications do not require use of certain peripherals or full speed operation from the peripherals are not required, the power required by the part may be significantly less than the numbers stated below.

Table 63. Power Dissipation Values

Part Type	Power Rail	Maximum Power Dissipation (Watts)
Intel® IXP4XX Product Line of Network Processors 266 MHz	V_{CC}	1.0
	V_{CCP}	1.0
Intel® IXP4XX Product Line of Network Processors 400 MHz	V_{CC}	1.1
	V_{CCP}	1.1
Intel® IXP4XX Product Line of Network Processors 533 MHz	V_{CC}	1.2
	V_{CCP}	1.2

5.8 Ordering Information

For ordering information, please contact your local Intel sales representative.