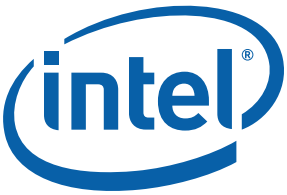


An Introduction to the Intel® QuickPath Interconnect

January 2009

Document Number: 320412-001US

Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information.

Contents

Revision History

Document Number	Revision Number	Description	Date
320412	-001US	Initial Release	January 2009

§

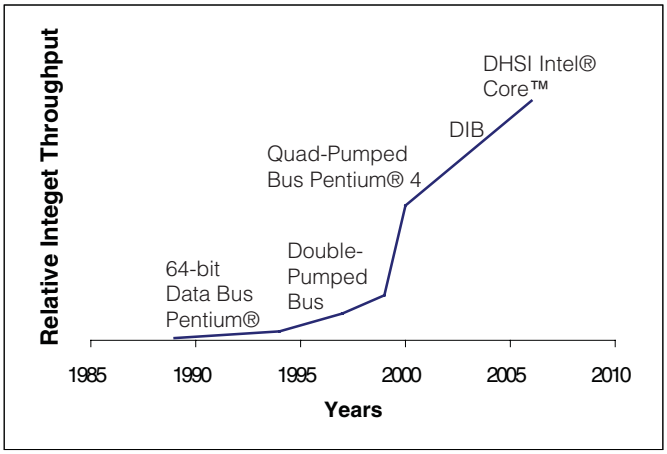
Executive Overview

Intel® microprocessors advance their performance ascension through ongoing microarchitecture evolutions and multi-core proliferation. The processor interconnect has similarly evolved (see Figure 1), thereby keeping pace with microprocessor needs through faster buses, quad-pumped buses, dual independent buses (DIB), dedicated high-speed interconnects (DHSI), and now the Intel® QuickPath Interconnect.

The Intel® QuickPath Interconnect is a high-speed, packetized, point-to-point interconnect used in Intel's next generation of microprocessors first produced in the second half of 2008. The narrow high-speed links stitch together processors in a distributed shared memory[1]-style platform architecture. Compared with today's wide front-side buses, it offers much higher bandwidth with low latency. The Intel® QuickPath Interconnect has an efficient architecture allowing more interconnect performance to be achieved in real systems. It has a snoop protocol optimized for low latency and high scalability, as well as packet and lane structures enabling quick completions of transactions. Reliability, availability, and serviceability features (RAS) are built into the architecture to meet the needs of even the most mission-critical servers. With this compelling mix of performance and features, it's evident that the Intel® QuickPath Interconnect provides the foundation for future generations of Intel microprocessors and that various vendors are designing innovative products around this interconnect technology.

1. Scalable shared memory or distributed shared-memory (DSM) architectures are terms from Hennessy & Patterson, Computing Architecture: A Quantitative Approach. It can also be referred to as NUMA, non-uniform memory access.

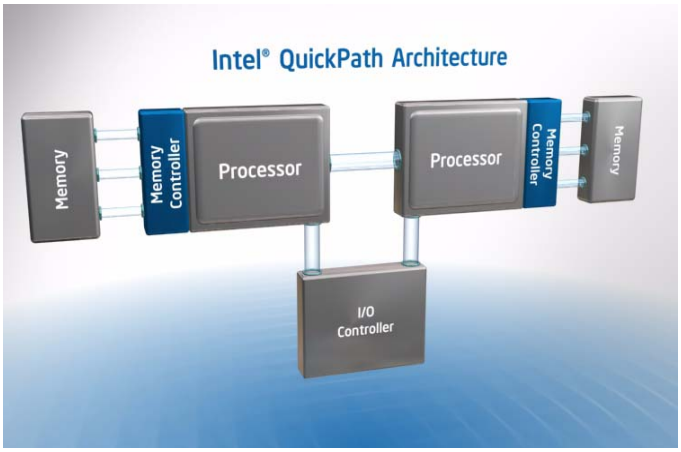
Figure 1. Performance and Bandwidth Evolution



Introduction

For the purpose of this paper, we will start our discussion with the introduction of the Intel® Pentium® Pro processor in 1992. The Pentium® Pro microprocessor was the first Intel architecture microprocessor to support symmetric multiprocessing in various multiprocessor configurations. Since then, Intel has produced a steady flow of products that are designed to meet the rigorous needs of servers and workstations. In 2008, just over fifteen years after the Pentium® Pro processor was introduced, Intel took another big step in enterprise computing with the production of processors for the next generation of Intel 45-nm Hi-k Intel® Core™ microarchitecture.

Figure 2. Intel® QuickPath Architecture



The processors based on next-generation, 45-nm Hi-k Intel® Core™ microarchitecture also utilize a new system of framework for Intel micro-processors called the Intel® QuickPath Architecture (see Figure 2). This architecture generally includes memory controllers integrated into the microprocessors, which are connected together with a high-speed, point-to-point interconnect. The new Intel® QuickPath Interconnect provides high bandwidth and low latency, which deliver the interconnect performance needed to unleash the new microarchitecture and deliver the Reliability, Availability, and Serviceability (RAS) features expected in enterprise applications. This new interconnect is one piece of a balanced platform approach to achieving superior performance. It is a key ingredient in keeping pace with the next generation of microprocessors.

Paper Scope and Organization

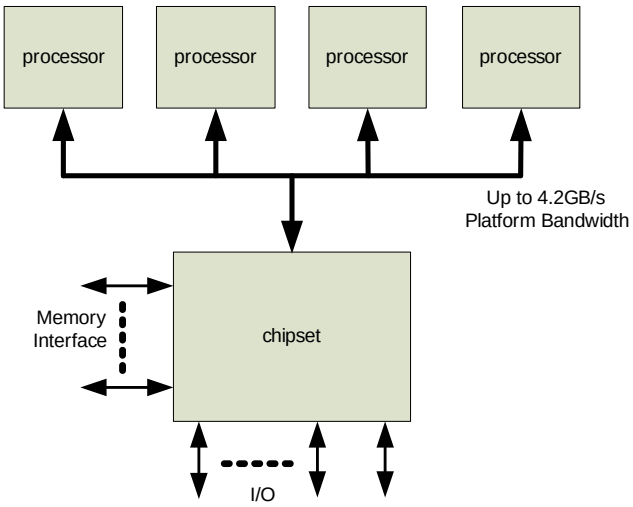
The remainder of this paper will describe features of the Intel® QuickPath Interconnect. First, a short overview of the evolution of the processor interface, including the Intel® QuickPath Interconnect, is provided then each of the Intel® QuickPath Interconnect architectural layers is defined, an overview of the coherency protocol described, board layout features surveyed, and some initial performance expectations provided. The primary audience for this paper is the technology enthusiast as well as other individuals who want to understand the future direction of this interconnect technology and the benefits it offers. Although all efforts are made to ensure accuracy, the information provided in this paper is brief and should not be used to make design decisions. Please contact your Intel representative for technical design collateral, as appropriate.

Evolution of Processor Interface

In older shared bus systems, as shown in Figure 3, all traffic is sent across a single shared bi-directional bus, also known as the front-side bus (FSB). These wide buses (64-bit for Intel® Xeon® processors and 128-bit for Intel® Itanium® processors) bring in multiple data bytes at a time. The challenge to this approach was the electrical constraints encountered with increasing the frequency of the wide source synchronous buses. To get around this, Intel evolved the bus through a series of technology improvements.

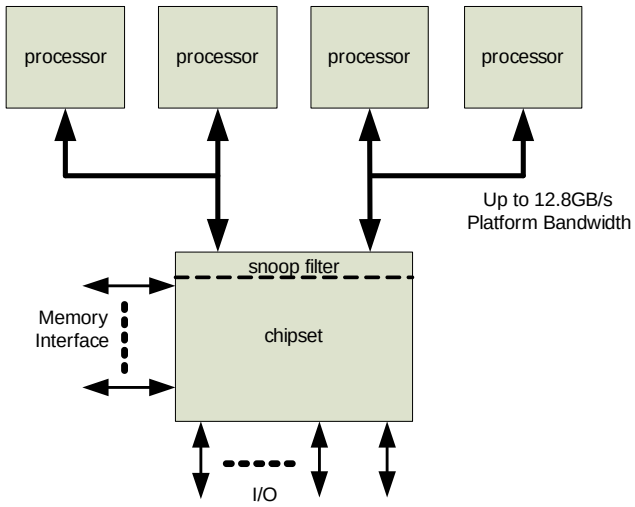
Initially in the late 1990s, data was clocked in at 2X the bus clock, also called double-pumped. Today's Intel® Xeon® processor FSBs are quad-pumped, bringing the data in at 4X the bus clock. The top theoretical data rate on FSBs today is 1.6 GT/s.

Figure 3. Shared Front-side Bus, up until 2004



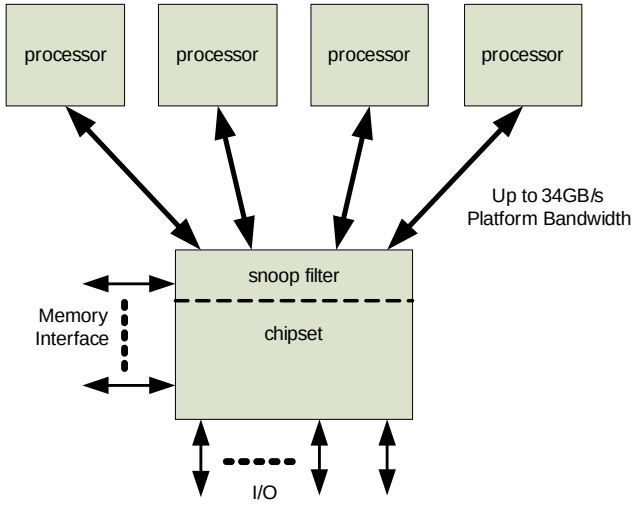
To further increase the bandwidth of the front-side bus based platforms, the single-shared bus approach evolved into dual independent buses (DIB), as depicted in Figure 4. DIB designs essentially doubled the available bandwidth. However, all snoop traffic had to be broadcast on both buses, and if left unchecked, would reduce effective bandwidth. To minimize this problem, snoop filters were employed in the chipset to cache snoop information, thereby significantly reducing bandwidth loading.

Figure 4. Dual Independent Buses, circa 2005



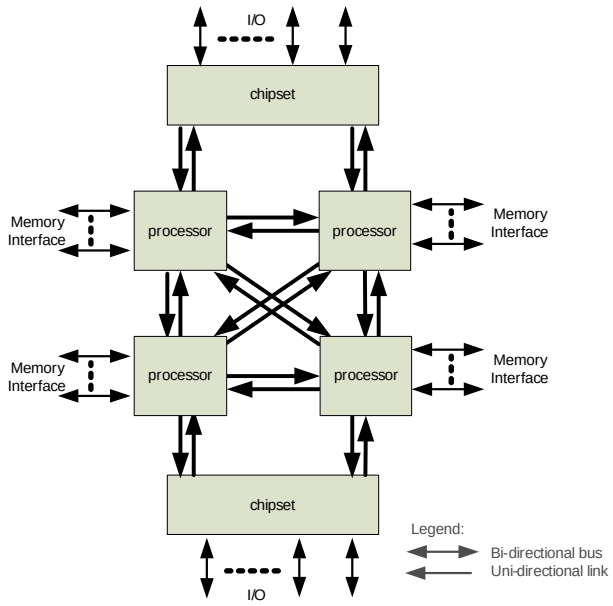
The DIB approach was extended to its logical conclusion with the introduction of dedicated high-speed interconnects (DHSI), as shown in Figure 5. DHSI-based platforms use four FSBs, one for each processor in the platform. Again, snoop filters were employed to achieve bandwidth scaling.

Figure 5. Dedicated High-speed Interconnects, 2007



With the production of processors based on next-generation, 45-nm Hi-k Intel® Core™ microarchitecture, the Intel® Xeon® processor fabric will transition from a DHSI, with the memory controller in the chipset, to a distributed shared memory architecture using Intel® QuickPath Interconnects. This configuration is shown in Figure 6. With its narrow uni-directional links based on differential signaling, the Intel® QuickPath Interconnect is able to achieve substantially higher signaling rates, thereby delivering the processor interconnect bandwidth necessary to meet the demands of future processor generations.

Figure 6. Intel® QuickPath Interconnect

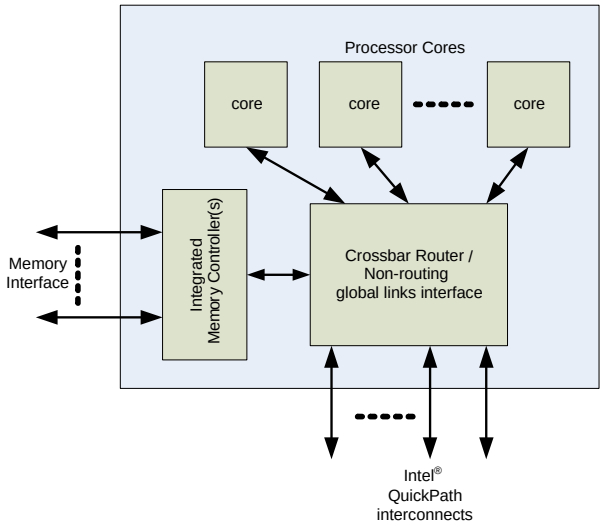


Interconnect Overview

The Intel® QuickPath Interconnect is a high-speed point-to-point interconnect. Though sometimes classified as a serial bus, it is more accurately considered a point-to-point link as data is sent in parallel across multiple lanes and packets are broken into multiple parallel transfers. It is a contemporary design that uses some techniques similar to other point-to-point interconnects, such as PCI Express* and Fully-Buffered DIMMs. There are, of course, some notable differences between these approaches, which reflect the fact that these interconnects were designed for different applications. Some of these similarities and differences will be explored later in this paper.

Figure 7 shows a schematic of a processor with external Intel® QuickPath Interconnects. The processor may have one or more cores. When multiple cores are present, they may share caches or have separate caches. The processor also typically has one or more integrated memory controllers. Based on the level of scalability supported in the processor, it may include an integrated crossbar router and more than one Intel® QuickPath Interconnect port (a port contains a pair of uni-directional links).

Figure 7. Block Diagram of Processor with Intel® QuickPath Interconnects

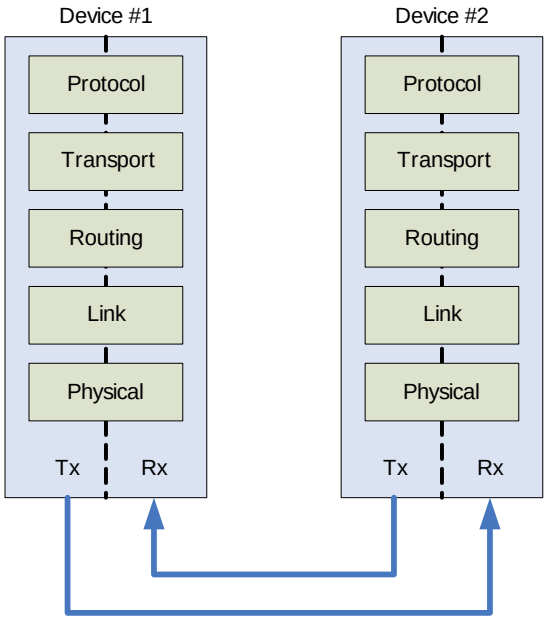


The physical connectivity of each interconnect link is made up of twenty differential signal pairs plus a differential forwarded clock. Each port supports a link pair consisting of two uni-directional links to complete the connection between two components. This supports traffic in both directions simultaneously. To facilitate flexibility and longevity, the interconnect is defined as having five layers (see Figure 8): Physical, Link, Routing, Transport, and Protocol.

- The Physical layer consists of the actual wires carrying the signals, as well as circuitry and logic to support ancillary features required in the transmission and receipt of the 1s and 0s. The unit of transfer at the Physical layer is 20-bits, which is called a Phit (for Physical unit).
- The next layer up the stack is the Link layer, which is responsible for reliable transmission and flow control. The Link layer's unit of transfer is an 80-bit Flit (for Flow control unit).
- The Routing layer provides the framework for directing packets through the fabric.
- The Transport layer is an architecturally defined layer (not implemented in the initial products) providing advanced routing capability for reliable end-to-end transmission.
- The Protocol layer is the high-level set of rules for exchanging packets of data between devices. A packet is comprised of an integral number of Flits.

The Intel® QuickPath Interconnect includes a cache coherency protocol to keep the distributed memory and caching structures coherent during system operation. It supports both low-latency source snooping and a scalable home snoop behavior. The coherency protocol provides for direct cache-to-cache transfers for optimal latency.

Figure 8. Architectural Layers of the Intel® QuickPath Interconnect



Within the Physical layer are several features (such as lane/polarity reversal, data recovery and deskew circuits, and waveform equalization) that ease the design of the high-speed link. Initial bit rates supported are 4.8 GT/s and 6.4 GT/s. With the ability to transfer 16 bits of data payload per forwarded clock edge, this translates to 19.2 GB/s and 25.6 GB/s of theoretical peak data bandwidth per link pair.

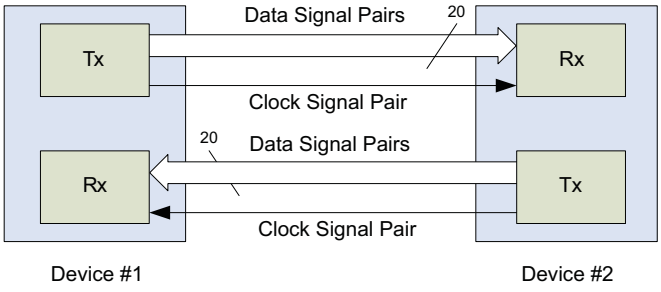
At these high bit rates, RAS requirements are met through advanced features which include: CRC error detection, link-level retry for error recovery, hot-plug support, clock fail-over, and link self-healing. With this combination of features, performance, and modularity, the Intel® QuickPath Interconnect provides a high-bandwidth, low-latency interconnect solution capable of unleashing the performance potential of the next-generation of Intel microarchitecture.

Interconnect Details

The remainder of this paper describes the Intel® QuickPath Interconnect in more detail. Each of the layers will be defined, an overview of the coherency protocol described, board layout features surveyed and some initial thoughts on performance provided.

Physical Layer

Figure 9. Physical Layer Diagram



The Physical layer consists of the actual wires carrying the signals, as well as the circuitry and logic required to provide all features related to the transmission and receipt of the information transferred across the link. A link pair consists of two uni-directional links that operate simultaneously. Each full link is comprised of twenty 1-bit lanes that use differential signaling and are DC coupled. The specification defines operation in full, half, and quarter widths. The operational width is identified during initialization, which can be initiated during operation as part of a RAS event. A Phit contains all the information transferred by the Physical layer on a single clock edge. At full-width that would be 20 bits, at half-width 10 bits and at quarter-width 5 bits. A Flit (see the "Link Layer" section) is always 80 bits regardless of the link width, so the number of Phits needed to transmit a Flit will increase by a factor of two or four for half and quarter-width links, respectively.

Each link also has one and only one forwarded clock. The clock lane is required for each direction. In all, there are a total of eighty-four (84) individual signals to make up a single Intel® QuickPath Interconnect port. This is significantly fewer signals than the wider, 64-bit front-side bus, which has approximately 150 pins. All transactions are encapsulated across these links, including configuration and all interrupts. There are no side-band signals. Table 1 compares two different interconnect technologies.

Table 1. Processor Interconnect Comparison

	Intel Front-Side Bus[3]	Intel® QuickPath Interconnect[3]
Topology	Bus	Link
Signaling Technology[1]	GTL+	Diff.
Rx Data Sampling[2]	SrcSync	FwdClk
Bus Width (bits)	64	20
Max Data Transfer Width	64	16
Requires Side-band Signals	Yes	No
Total Number of Pins	150	84
Clocks Per Bus	1	1
Bi-directional Bus	Yes	No
Coupling	DC	DC
Requires 8/10-bit encoding	No	No

[1] Diff. stands for differential signaling.
[2] SrcSync stands for source synchronous data sampling. FwdClk(s) means forwarded clock(s).
[3] Source: Intel internal presentation, December, 2007.

The link is operated at a double-data rate, meaning the data bit rate is twice the forwarded clock frequency. The forwarded clock is a separate signal, not an encoded clock as used in PCI Express* Gen 1 and Gen 2. The initial product implementations are targeting bit rates of 6.4 GT/s and 4.8 GT/s. Table 2 compares the bit rates of different interconnect technologies.

Table 2. Contemporary Interconnect Bit Rates

Technology	Bit Rate (GT/s)
Intel front-side bus	1.6[1]
Intel® QuickPath Interconnect	6.4/4.8
Fully-Buffered DIMM	4.0[2]
PCI Express* Gen1[3]	2.5
PCI Express* Gen2[3]	5.0
PCI Express* Gen3[3]	8.0

[1]Transfer rate available on Intel® Xeon® Processor-based Workstation Platform.
[2]Transfer rate available on Intel® Xeon® Processor-based Workstation Platform.
[3]Source: PCI Express* 3.0 Frequently Asked Questions, PCI-SIG, August 2007.

The receivers include data recovery circuitry which can allow up to several bit-time intervals of skew between data lanes and the forwarded clock. De-skew is performed on each lane independently. The amount of skew that can be tolerated is product-specific and design collateral should be consulted for specifics. On the transmitter side, waveform equalization is used to obtain proper electrical characteristics. Signal integrity simulation is required to define the tap settings for waveform equalization. Routing lengths are also product and frequency specific. One implementation example would be that the routing length could vary from 14” to 24” with zero to two connectors and 6.4 GT/s to 4.8 GT/s. Again, consult the design collateral and your Intel technical representative for specifics. To ease routing constraints, both lane and polarity reversal are supported. Polarity reversal allows an individual lane’s differential pair to be swapped. In addition to reversing a lane, an entire link can be reversed to ease routing. So lanes 0 to 19 can be connected to 19 to 0 and the initialization logic would adjust for the proper data transfer. This aids board routing by avoiding cross-over of signals. Both types of reversal are discovered automatically by the receiver during initialization. The logic configures itself appropriately to accommodate the reversal without any need for external means, such as straps or configuration registers.

Logic circuits in the Physical layer are responsible for the link reset, initialization and training. Like other high-speed links, the Physical layer is designed to expect a low rate of bit errors due to random noise and jitter in the system. These errors are routinely detected and corrected through functions in the Link layer. The Physical layer also performs periodic retraining to avoid the Bit Error Rate (BER) from exceeding the specified threshold. The Physical layer supports loop-back test modes using the Intel® Interconnect Built-In Self Test (Intel® IBIST). Intel® IBIST tools provide a mechanism for testing the entire interconnect path at full operational speed without the need for external test equipment. These tools greatly assist in the validation efforts required to ensure proper platform design integrity.

Link Layer

The Link layer has three main responsibilities: (1) it guarantees reliable data transfer between two Intel® QuickPath Interconnect protocol or routing entities; (2) it is responsible for the flow control between two protocol agents; and (3) in abstracting the Physical layer, it provides services to the higher layers.

The smallest unit of measure at the Link layer is a Flit (flow control unit). Each Flit is 80 bits long, which on a full-width link would translate to four Phits. The Link layer presents a set of higher-level services to the stack. These services include multiple message classes and multiple virtual networks, and together are used to prevent protocol deadlocks.

Message Classes

The Link layer supports up to fourteen (14) Protocol layer message classes of which six are currently defined. The remaining eight message classes are reserved for future use. The message classes provide independent transmission channels (virtual channels) to the Protocol layer, thereby allowing sharing of the physical channel.

The message classes are shown in Table 3. The messages with the SNP, NDR, DRS, NCS and NCB message encodings are unordered. An unordered channel has no required relationship between the order in which messages are sent on that channel and the order in which they are received. The HOM message class does have some ordering requirements with which components are required to comply.

Table 3. Message Classes

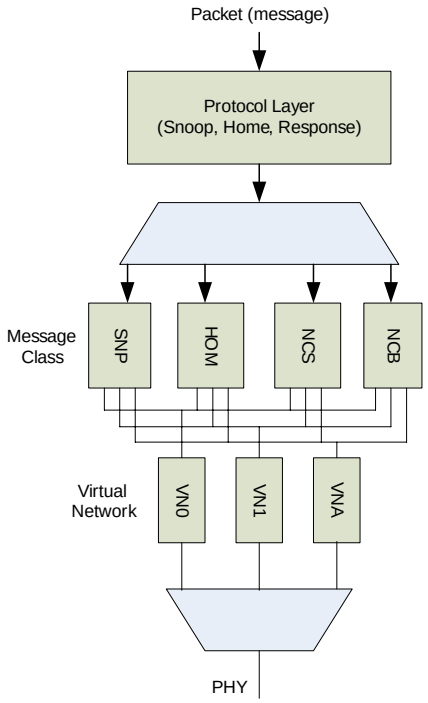
Name	Abbr	Ordering	Data
Snoop	SNP	None	No
Home	HOM	Required	No
Non-data Response	NDR	None	No
Data Response	DRS	None	Yes
Non-coherent Standard	NCS	None	No
Non-coherent Bypass	NCB	None	Yes

Virtual Networks

Virtual networks provide the Link layer with an additional method for replicating each message class into independent virtual channels. Virtual networks facilitate the support of a variety of features, including reliable routing, support for complex network topologies, and a reduction in required buffering through adaptively buffered virtual networks.

The Link layer supports up to three virtual networks. Each message class is subdivided among the three virtual networks. There are up to two independently buffered virtual networks (VN0 and VN1) and one shared adaptive buffered virtual network (VNA). See Figure 10. The total number of virtual channels supported is the product of the virtual networks supported and the message classes supported. For the Intel® QuickPath Interconnect Link layer this is a maximum of eighteen virtual channels (three SNP, three HOM, three NDR, three DRS, three NCS, and three NCB).

Figure 10. Virtual Networks

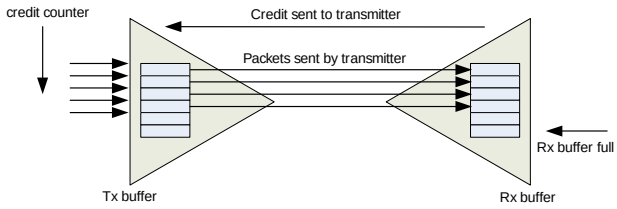


Note: Only 4 of 6 message classes shown.

Credit/Debit Scheme

The Link layer uses a credit/debit scheme for flow control, as depicted in Figure 11. During initialization, a sender is given a set number of credits to send packets, or Flits, to a receiver. Whenever a packet or Flit is sent to the receiver, the sender decrements its credit counters by one credit, which can represent either a packet or a Flit depending on the type of virtual network being used. Whenever a buffer is freed at the receiver, a credit is returned to the sender for that buffer. When the sender's credits for a given channel have been exhausted, it stops sending on that channel. Each packet contains an embedded flow control stream. This flow control stream returns credits from a receiving Link layer entity to a sending Link layer entity. Credits are returned after the receiving Link layer has consumed the received information, freed the appropriate buffers, and is ready to receive more information into those buffers.

Figure 11. Credit/Debit Flow Control



Reliable Transmission

CRC error checking and recovery procedures are provided by the Link layer to isolate the effects of routine bit errors (which occur on the physical interconnect) from having any higher layer impact. The Link layer generates the CRC at the transmitter and checks it at the receiver. The Link layer requires the use of 8 bits of CRC within each 80-bit Flit. An optional rolling 16-bit CRC method is available for the most demanding RAS applications. The CRC protects the entire information flow across the link; all signals are covered. The receiving Link layer performs the CRC calculation on the incoming Flit and if there is an error, initiates a link level retry. If an error is detected, the Link layer automatically requests that the sender backup and retransmit the Flits that were not properly received. The retry process is initiated by a special cycle Control Flit, which is sent back to the transmitter to start the backup and retry process. The Link layer initiates and controls this process; no software intervention is needed. The Link layer reports the retry event to the software stack for error tracking and predictive maintenance algorithms. Table 4 details some of the error detection and recovery capabilities provided by the Intel® QuickPath Interconnect. These capabilities are a key element of meeting the RAS requirements for new platforms.

Table 4. RAS: Error Detection and Recovery Features

Feature	Intel® QuickPath Interconnect
CRC checking	Required
All signals covered	Yes
CRC type	8 bits / 80 bits
CRC bits/64-B packet	72
Impact of error	Recovered
Cycle penalty[1] (bit times)	None
Additional features	Rolling CRC

[1] Lower is better. The cycle penalty increases latency and reduces bandwidth utilization.

In addition to error detection and recovery, the Link layer controls the use of some additional RAS features in the Physical layer, such as self-healing links and clock fail-over. Self-healing allows the interconnect to recover from multiple hard errors with no loss of data or software intervention. Unrecoverable soft errors will initiate a dynamic link width reduction cycle. The sender and receiver negotiate to connect through a reduced link width. The link automatically reduces its width to either half or quarter-width, based on which quadrants (five lanes) of the link are good. As the data transmission is retried in the process, no data loss results. Software is notified of the events, but need not directly intervene or control the operation. In the case of the clock lane succumbing to errors, the clock is mapped to a pre-defined data lane and the link continues to operate at half-width mode. If the data lane that is serving as the fail-over is not functional, there is a second fail-over clock path. Both the self-healing and clock fail-over functionalities are direction independent. This means one direction of the link could be in a RAS mode, and the other direction could still be operating at full capacity. In these RAS modes, there is no loss of interconnect functionality or error protection. The only impact is reduced bandwidth of the link that is operating in RAS mode.

Routing Layer

The Routing layer is used to determine the course that a packet will traverse across the available system interconnects. Routing tables are defined by firmware and describe the possible paths that a packet can follow. In small configurations, such as a two-socket platform, the routing options are limited and the routing tables quite simple. For larger systems, the routing table options are more complex, giving the flexibility of routing and rerouting traffic depending on how (1) devices are populated in the platform, (2) system resources are partitioned, and (3) RAS events result in mapping around a failing resource.

Transport Layer

The optional Transport layer provides end-to-end transmission reliability. This architecturally defined layer is not part of Intel's initial product implementations and is being considered for future products.

Protocol Layer

In this layer, the packet is defined as the unit of transfer. The packet contents definition is standardized with some flexibility allowed to meet differing market segment requirements. The packets are categorized into six different classes, as mentioned earlier in this paper: home, snoop, data response, non-data response, non-coherent standard, and non-coherent bypass. The requests and responses affect either the coherent system memory space or are used for non-coherent transactions (such as configuration, memory-mapped I/O, interrupts, and messages between agents).

The system's cache coherency across all distributed caches and integrated memory controllers is maintained by all the distributed agents that participate in the coherent memory space transactions, subject to the rules defined by this layer. The Intel® QuickPath Interconnect coherency protocol allows both home snoop and source snoop behaviors. Home snoop behavior is optimized for greater scalability, whereas source snoop is optimized for lower latency. The latter is used primarily in smaller scale systems where the smaller number of agents creates a relatively low amount of snoop traffic. Larger systems with more snoop agents could develop a significant amount of snoop traffic and hence would benefit from a home snoop mode of operation. As part of the coherence scheme, the Intel® QuickPath Interconnect implements the popular MESI[2] protocol and, optionally, introduces a new F-state.

MESIF

The Intel® QuickPath Interconnect implements a modified format of the MESI coherence protocol. The standard MESI protocol maintains every cache line in one of four states: modified, exclusive, shared, or invalid. A new read-only forward state has also been introduced to enable cache-to-cache clean line forwarding. Characteristics of these states are summarized in Table 5. Only one agent can have a line in this F-state at any given time; the other agents can have S-state copies. Even when a cache line has been forwarded in this state, the home agent still needs to respond with a completion to allow retirement of the resources tracking the transaction. However, cache-to-cache transfers offer a low-latency path for returning data other than that from the home agent's memory.

Table 5. Cache States

State	Clean / Dirty	May Write?	May Forward?	May Transition To?
M – Modified	Dirty	Yes	Yes	-
E – Exclusive	Clean	Yes	Yes	MSIF
S – Shared	Clean	No	No	I
I – Invalid	-	No	No	-
F – Forward	Clean	No	Yes	SI

2. The MESI protocol (pronounced "messy"), [named] after the four states of its cache lines: Modified, Exclusive, Shared, and Invalid. - In Search of Clusters, by Gregory Pfister.

Protocol Agents

The Intel® QuickPath Interconnect coherency protocol consists of two distinct types of agents: caching agents and home agents. A micro-processor will typically have both types of agents and possibly multiple agents of each type.

A caching agent represents an entity which may initiate transactions into coherent memory, and which may retain copies in its own cache structure. The caching agent is defined by the messages it may sink and source according to the behaviors defined in the cache coherence protocol. A caching agent can also provide copies of the coherent memory contents to other caching agents.

A home agent represents an entity which services coherent transactions, including handshaking as necessary with caching agents. A home agent supervises a portion of the coherent memory. Home agent logic is not specifically the memory controller circuits for main memory, but rather the additional Intel® QuickPath Interconnect logic which maintains the coherency for a given address space. It is responsible for managing the conflicts that might arise among the different caching agents. It provides the appropriate data and ownership responses as required by a given transaction's flow.

There are two basic types of snoop behaviors supported by the Intel® QuickPath Interconnect specification. Which snooping style is implemented is a processor architecture specific optimization decision. To over-simplify, source snoop offers the lowest latency for small multi-processor configurations. Home snooping offers optimization for the best performance in systems with a high number of agents.

The next two sections illustrate these snooping behaviors.

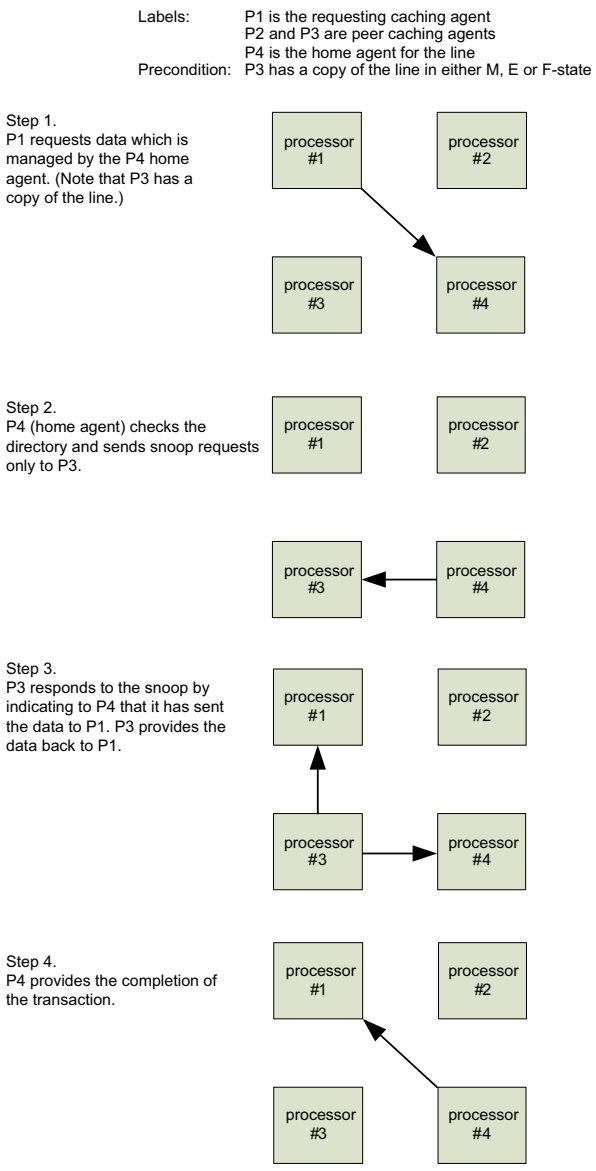
Home Snoop

The home snoop coherency behavior defines the home agent as responsible for the snooping of other caching agents. The basic flow for a message involves up to four steps (see Figure 12). This flow is sometimes referred to as a three-hop snoop because the data is delivered in step 3. To illustrate, using a simplified read request to an address managed by a remote home agent, the steps are:

1. The caching agent issues a request to the home agent that manages the memory in question.
2. The home agent uses its directory structure to target a snoop to the caching agent that may have a copy of the memory in question.
3. The caching agent responds back to the home agent with the status of the address. In this example, processor #3 has a copy of the line in the proper state, so the data is delivered directly to the requesting cache agent.
4. The home agent resolves any conflicts, and if necessary, returns the data to the original requesting cache agent (after first checking to see if data was delivered by another caching agent, which in this case it was), and completes the transaction.

The Intel® QuickPath Interconnect home snoop behavior implementation typically includes a directory structure to target the snoop to the specific caching agents that may have a copy of the data. This has the effect of reducing the number of snoops and snoop responses that the home agent has to deal with on the interconnect fabric. This is very useful in systems that have a large number of agents, although it comes at the expense of latency and complexity. Therefore, home snoop is targeted at systems optimized for a large number of agents.

Figure 12. Home Snoop Example



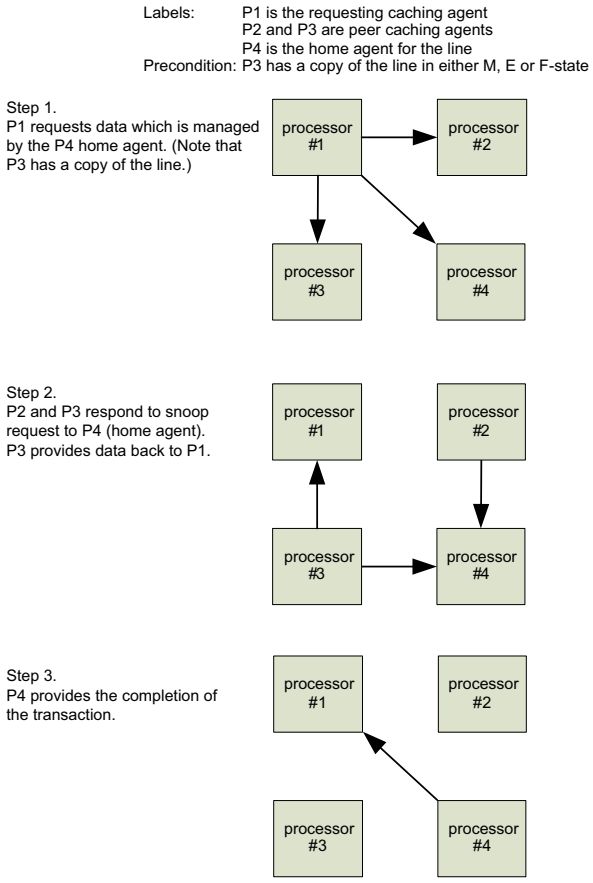
Source Snoop

The source snoop coherency behavior streamlines the completion of a transaction by allowing the source of the request to issue both the request and any required snoop messages. The basic flow for a message involves only three steps, sometimes referred to as a two-hop snoop since data can be delivered in step 2. Refer to Figure 13. Using the same read request discussed in the previous section, the steps are:

1. The caching agent issues a request to the home agent that manages the memory in question and issues snoops to all the other caching agents to see if they have copies of the memory in question.
2. The caching agents respond to the home agent with the status of the address. In this example, processor #3 has a copy of the line in the proper state, so the data is delivered directly to the requesting cache agent.
3. The home agent resolves any conflicts and completes the transaction.

The source snoop behavior saves a “hop,” thereby offering a lower latency. This comes at the expense of requiring agents to maintain a low latency path to receive and respond to snoop requests; it also imparts additional bandwidth stress on the interconnect fabric, relative to the home snoop method. Therefore, the source snoop behavior is most effective in platforms with only a few agents.

Figure 13. Source Snoop Example



Performance

Much is made about the relationship between the processor interconnect and overall processor performance. There is not always a direct correlation between interconnect bandwidth, latency and processor performance. Instead, the interconnect must perform at a level that will not limit processor performance. Microprocessor performance has been continually improving by architectural enhancements, multiple cores, and increases in processor frequency. It is important that the performance of the interconnect remain sufficient to support the latent capabilities of a processor's architecture. However, providing more interconnect bandwidth or better latency by itself does not equate to an increase in performance. There are a few benchmarks which create a synthetic stress on the interconnect and can be used to see a more direct correlation between processor performance and interconnect performance, but they are not good proxies for real end-user performance. The Intel® QuickPath Interconnect offers a high-bandwidth and low-latency interconnect solution. The raw bandwidth of the link is 2X the bandwidth of the previous Intel® Xeon® processor front-side bus. To the first order, it is clear to see the increase in bandwidth that the Intel® QuickPath Interconnect offers.

Raw Bandwidth

Raw and sustainable bandwidths are very different and have varying definitions. The raw bandwidth, or maximum theoretical bandwidth, is the rate at which data can be transferred across the connection without any accounting for the packet structure overhead or other effects. The simple calculation is the number of bytes that can be transferred per second. The Intel® QuickPath Interconnect is a double-pumped data bus, meaning data is captured at the rate of one data transfer per edge of the forwarded clock. So every clock period captures two chunks of data. The maximum amount of data sent across a full-width Intel® QuickPath Interconnect is 16 bits, or 2 bytes. Note that 16 bits are used for this calculation, not 20 bits. Although the link has up to 20 1-bit lanes, no more than 16 bits of real 'data' payload are ever transmitted at a time, so the more accurate calculation would be to use 16 bits of data at a time. The maximum frequency of the initial Intel® QuickPath Interconnect implementation is 3.2 GHz. This yields a double-pumped data rate of 6.4 GT/s with 2 bytes per transition, or 12.8 GB/s. An Intel® QuickPath Interconnect link pair operates two uni-directional links simultaneously, which gives a final theoretical raw bandwidth of 25.6 GB/s. Using similar calculations, Table 6 shows the processor's bus bandwidth per port.

Table 6. Processor Bus Bandwidth Comparison, Per Port

Bandwidth (Higher is better)	Intel Front Side Bus	Intel® QuickPath Interconnect
Year	2007	2008
Rate (GT/s)	1.6	6.4
Width (bytes)	8	2
Bandwidth (GB/s)	12.8	25.6
Coherency	Yes	Yes

[1] Source: Intel internal presentation, December 2007.

Packet Overhead

The maximum theoretical bandwidth is not sustainable in a real system. Bandwidth is implementation specific. A step closer to sustainable bandwidth is accounting for the packet overhead. Although still not a true measurable result, it is closer to real-life bandwidth. The typical Intel® QuickPath Interconnect packet has a header Flit which requires four Phits to send across the link. The typical data transaction in microprocessors is a 64-byte cache line. The data payload, therefore,

requires 32 Phits to transfer. The CRC codes are sent inline with the data which is one of the reasons only 16 bits of data are transmitted at a time even though 20 lanes are available. So a data packet would require four Phits of header plus 32 Phits of payload, or 36 total Phits. At 6.4 GT/s, that means a 64-byte cache line would transfer in 5.6 ns. Table 7 compares the overhead associated with different interconnects for sending a small (64-byte) packet. The table shows that the Intel® QuickPath Interconnect has lower overhead than PCI Express* when handling smaller sized data packets, which means more bandwidth and lower latency.

Table 7. Small Packet Header Overhead Comparison

Overhead (Lower is better)	Intel® QuickPath Interconnect[1]	PCI Express*[2]
Version		Gen2
Max payload size (bytes)	64	4096
Packet payload size (bytes)	64	64
Number of packets	1	1
Payload (phits)	32	32
Header + CRC (phits)	4	7
8b/10b encoding impact[1] (phits)	n/a	8
Total overhead (phits)	4	15
Total packet size (phits)	36	47
Total overhead (%)	11%	32%

[1] Source: Intel internal presentation, December 2007.
[2] Source: PCI Express System Architecture, MindShare* 2004.

I/O Packet Size

The previous bandwidth comparison was based on a maximum data payload size of 64 bytes, which is the size of a cache line. Cache line sizes are the basic unit for processor oriented buses, such as the Intel® QuickPath Interconnect. For more I/O orientated buses like PCI Express*, larger packets are possible. This is important for I/O applications that transfer larger packets and can, therefore, amortize the impact of the header over the large packet. When comparing the interconnects for I/O applications, such as adapter cards, it is important to consider the large packet sizes. Table 8 shows that when larger sized packets are used, PCI Express* provides relatively low overhead.

Table 8. Impact of Packet Size Overhead Amortization

Overhead (Lower is better)	Intel® QuickPath Interconnect[1]	PCI Express*[2]
Version		Gen2
Packet payload size (bytes)	64	256
Total payload size (bytes)	4096	4096
Number of packets	64	16
Payload (phits)	2048	2048
Header + CRC (phits)	256	112
8b/10b encoding impact[1] (phits)	n/a	432
Total overhead (phits)	256	544
Total packet size (phits)	2304	2592
Total overhead (%)	11%	21%

[1] Source: Intel internal presentation, December 2007.
[2] Source: PCI Express System Architecture, MindShare* 2004.

In addition to I/O bandwidth, another important differentiator is latency. Intel and IBM made a series of suggestions to the PCI SIG for improving latency that the PCI SIG evaluated and modified through its specification development process to be incorporated in a future version of PCI Express*. These latency improvements are expected to be available in third-generation PCI Express* products.

Reliability, Availability, and Serviceability

A key difference between desktop computers and servers is reliability, availability, and serviceability. Although all users want a reliable computer, the impact of a problem is typically much greater on a server running a business' core application than on a personal workstation. Therefore, processors used in servers typically have more RAS features than their client brethren. For example, more expensive ECC memory is standard in servers, whereas clients use the less expensive non-ECC DRAM. Likewise, on the processor interconnect, RAS features are important to a server processor interconnect, especially the larger expandable and mission-critical servers found in IT data centers. The Intel® QuickPath Interconnect was designed with these types of applications in mind and its architecture includes several important RAS features. Product developers will optimize the overall system design by including the set of RAS features as needed to meet the requirements of a particular platform. As such, the inclusion of some of these features is product specific.

One feature common across all Intel® QuickPath Interconnect implementations is the use of CRC. The interconnect transmits 8 bits of CRC with every Flit, providing error detection without a latency performance penalty. Other protocols use additional cycles to send the CRC, therefore impacting performance. Many of these RAS features were identified previously in the Link layer section of this document. These RAS features include retry mode and per-Flit CRC, as well as product-specific features such as rolling CRC, hot detect, link self healing, and clock fail-over.

Processor Bus Applications

In recent years, Intel has enabled innovative silicon designs that are closely coupled to the processor. For years, the processor bus (front-side bus) has been licensed to companies to develop chipsets that allow servers with eight, sixteen, thirty-two, or more processors. More recently, several companies have developed products that allow for a re-programmable FPGA module to plug into an existing Intel® Xeon® processor socket. These innovative accelerator applications allow for significant speed-up of niche algorithms in areas like financial analytics, image processing, and oil and gas recovery. These innovative applications will continue on Intel's new processor platforms that use the Intel® QuickPath Interconnect instead of the front-side bus. Unlike high-performance I/O and accelerator designs found on PCI Express*, these innovative products require the benefits of close coupling to the processor, such as ultra-low latency and cache coherency. The bulk of high-performance I/O and accelerator designs are found on the PCI Express* bus, which has an open license available through the PCI SIG. This bus, however, has the limitation of being non-coherent. For applications requiring coherency, the Intel® QuickPath Interconnect technology could be licensed from Intel. As new innovative accelerator applications are developed, they can take advantage of PCI Express* for the broadest set of applications and Intel® QuickPath Interconnect for the remaining cache-coherent niche.

Summary

With its high-bandwidth, low-latency characteristics, the Intel® QuickPath Interconnect advances the processor bus evolution, unlocking the potential of next-generation microprocessors. The 25.6 GB/s of low-latency bandwidth provides the basic fabric required for distributed shared memory architectures.The inline CRC codes provide more error coverage with less overhead than serial CRC approaches. Features such as lane reversal, polarity reversal, clock fail-over, and self-healing links ease design of highly reliable and available products.The two-hop, source snoop behavior with cache line forwarding offers the shortest request completion in mainstream systems, while the home snoop behavior allows for optimizing highly scalable servers. With all these features and performance it's no surprise that various vendors are designing innovative products around this interconnect technology and that it provides the foundation for the next generation of Intel® microprocessors and many more to come.

§