Ivy Bridge (IVB) is Intel’s first processor (CPU) design that services product markets from high-end desktops to mission-critical computing.

With one converged design, IVB enables a portfolio of products and meets power, performance, and cost targets. Using Intel’s 22-nm process technology, IVB achieves scalability in die size, core count, cache size, socket count, and memory size while improving power efficiency and decreasing idle power.

Ivy Bridge (IVB) represents Intel’s first server CPU that services product markets from high-end desktops to mission-critical computing. With one converged design, IVB enables the refresh of a rich portfolio of products on a single CPU architecture generation over a two-quarter time window, as opposed to four to five quarters, which was typical in prior-generation servers. IVB uses Intel’s 22-nm process technology, integrates the IVB core, and adds several architecture extensions and innovations to deliver scalable and energy-efficient performance in all target markets.

As we emphasize convergence, it is important to understand the target server space, which comprises the Intel Xeon processor E3-1200 v2, E5-4600/2600/1600 v2 (referred to as the E5 line), and E7-8800/4800/2800 v2 (referred to as the E7 line) product families. The E3 line is targeted for single-socket platforms for small business and dense computing needs. The E5 line offers products for the volume entry-level, efficient computing, high-end density servers, as well as workstations, whereas the E7 line delivers servers to the high-end, expandable, and mission-critical segments. The E5 line supports up to four-socket platforms using Intel QuickPath Interconnect (Intel QPI)—that is, glueless system configurations—and the E7 line supports up to eight-socket glueless configurations. Both lines offer product options for higher socket count scalability using external node controllers. Historically, Intel has supported the described E5 and E7 server space with two server designs that used the same converged core but differed significantly in the remainder of the designs.

IVB plugs as a CPU refresh into the Romley platform, which was developed for the original Intel Xeon processor E5 family (formerly code-named Sandy Bridge [SNB]) products. It also launches the new Brickland platform for the expandable and mission-critical server segments as a follow-up to the Stoutland platform, which uses Intel Xeon processor E7 family (formerly code-named Westmere1 [WSM]). Across these two platforms, IVB offers approximately 80 different

Irma Esmer Papazian
Sailesh Kottapalli
Jeff Baxter
Jeff Chamberlain
Geetha Vedaraman
Brian Morris
Intel
product offerings called stock keeping units (SKUs).

Converged architecture

IVB builds a superset architecture comprised of all E5 and E7 features and requirements while addressing modularity goals to enable coverage from the top of the SKU stack to the bottom.

Figure 1 shows the key building blocks of a server CPU. The IVB core is the execution engine of the CPU and has two levels of internal caches. The last level cache (LLC) is the large third-level cache structure shared by all cores on the same die. The integrated memory controller (MC) manages outstanding memory requests to locally attached memory. The coherence engine, which is also called the home agent (HA), manages system coherence for its attached memory. The integrated I/O (IIO) controller is the agent that manages PCI Express (PCIe), Direct Media Interface, and Crystal Beach Direct Memory Access. The off-die interconnect is the Intel QPI agent that enables connectivity to other sockets as well as to the external node controller. Finally, the on-die interconnect is the medium that ties all the building blocks together. Each of these blocks is impacted by the converged design. In this article, we will describe the architectural enhancements and feature additions applied to each block that went through major changes over SNB to comply with the convergence and performance goals of the IVB CPU.

On-die interconnect and LLC

IVB uses a modular ring-based on-die interconnect architecture similar to that used in the WSM and SNB server processors. This modular design treats the HAs, the IIO agent, the external QPI links, the cores, and the LLC banks as independent agents on a shared internal ring bus. Structurally, the HAs are placed along the bottom of the die, the external QPI and IIO interfaces are placed along the top of the die, and the cores and LLC slices are arranged in columns in between.

This organization provides a reasonable aspect ratio and latency for a die with two columns of cores. With two columns, the ring can route through both columns then across the system agents at top and bottom to form a single bidirectional ring that passes through all agents on the die. Figure 2a shows the ring for this case. This is the approach used on SNB, and it would have been sufficient for most of the IVB E5 product line. But, for the design to also achieve performance gains in the E7 product line, core counts greater than 10 were required. This made
two columns insufficient, because increasing the core count in each column beyond five cores creates a lopsided aspect ratio for the die and degrades the bandwidth deliverable from the system agents to any individual core. Hence, the needs of the E7 product line make it necessary to extend the die horizontally by adding a third column of core and cache slices. In addition, for this solution to cover core count ranges from four to 15 without wasting area and power at the lower core counts, it was a requirement that a single design support both the two- and three-column layouts while using the same single set of ring wires through each column as the previous generation.

To accommodate all the requirements, IVB extends to three parallel core and LLC columns. The ability to control which column any given message routes through is supported by multiplexing the ring wires at the point where the three columns of wires intersect together. Viewed globally, this ends up logically breaking up the ring into three distinct virtual rings that interconnect, as shown in Figure 2b. The left diagram shows a configuration in which the two outer columns can reach the middle column through the counterclockwise rotating paths while the two outer columns can reach each other through the clockwise rotating paths. The right diagram shows a configuration in which the paths used to move messages between the outer and inner columns are reversed. The multiplexers (muxes) alternate in phase between these two configurations at a constant frequency and enable any agent on the ring to reach any other agent on the ring from either direction simply by aligning the transmission of the message so it reaches the
mux at the appropriate time. A single set of ring schedulers supports both the two- and three-column versions of the die by letting the scheduler treat the temporal phase of the mux as a “don’t care” when the die has only two columns. This virtualized ring architecture enables a single design to efficiently support scalability without sacrificing performance, power, or area in either market segment.

Memory controller

The IVB memory controller services two types of memory. The E5 segment uses direct-attach DDR3 to provide low-latency access to commodity DIMMs in a configuration like that in SNB. E5 supports up to four channels of DDR3, with up to three DIMMs per channel, for a total of 12 DIMMs per CPU socket.

A critical value proposition of the E7 segment is support for greater memory capacity. To enable this requirement, E7 uses a memory buffer to facilitate access to more DIMMs. Each memory buffer has a single IVB interface that connects to two DDR3 channels, which doubles the number of DDR3 channels supported per socket. IVB E5 v2 can support up to 1.5 Tbytes per two-socket system, and the IVB E7 v2 can support up to 12 Tbytes per eight-socket system.

The common memory controller schedules memory accesses regardless of whether the socket connects directly to the DIMMs or to a memory buffer. Platform compatibility with the SNB server sets the number of native DDR channels at four, with support for DDR3-1867. For buffered memory, the same four channels connect to memory expansion buffers using the Intel Scalable Memory Interconnect 2.0 protocol (SMI2). SMI2 shares a physical interface with DDR3 and enables high-bandwidth memory expansion at enhanced transfer rates. SMI2 uses a subset of the pins needed for DDR3 by compressing the command and control pins, saving memory pins for E7 relative to the E5 platform.

The SMI2 protocol can optimize for performance by interleaving across the DIMMs behind the memory buffer independently (rather than lock-stepped) at speeds up to DDR3-1333, with SMI2 at 2,667 million transfers per second (MTps). Because the data bus widths of DDR3 and SMI2 are matched, SMI2 can sink the bandwidth of both DDR3 channels; in this mode, the read data responses are time-division multiplexed across DDR channels to minimize latency. In lock-step mode, the DDR3 and SMI2 busses operate at up to 1,600 MTps, with SMI2 amalgamating the data from two DDR channels.

Memory scheduling operates similarly, regardless of direct attach versus SMI2 mode. The E7 scheduler understands that two DDR channels share a bidirectional SMI2 link for the purpose of optimizing bandwidth for read and write modes; otherwise, the scheduling algorithms are identical. The scheduler has deep buffering (48 reads and 32 writes), optimized read/write transaction mixing, and minimized turnarounds to maximize bandwidth. Memory scheduling has been optimized to exploit the additional bank resources available on LRDIMMs.

Off-die interconnect: QPI

IVB supports multisocket systems connected through QPI links. Each QPI link comprises 20-bit lanes running at a speed of 8 billion transfers per second (GTps). The E5 enables two bidirectional QPI links per socket providing two- and four-socket glueless system solutions, which is similar to the prior SNB E5 product line. The E7 enables three bidirectional QPI links per socket providing up to eight-socket glueless connectivity options. IVB E7 maintains a similar connectivity profile to WSM E7 despite having one fewer QPI link; this is made possible by the on-die IIO. Another change for IVB E7 is the move from QPI v1.0 source snoopy protocol to the newer QPI v1.1 home snoopy protocol.

Coherence engine

The IVB coherence engine (or HA) is responsible for handling all DRAM requests “homed” at its node. To support the E7 socket scalability while keeping latencies low, the coherence structures and algorithms needed to be expanded over the prior implementation in SNB.

IVB is based on the QPI coherence protocol, which mandates preallocated trackers in the HA for tracking requests. Preallocation necessitates a large size tracker in the HA for socket scalability. IVB uses a two-level tracker
scheme that optimizes performance while maintaining area and power efficiency. A 512-entry first-level tracker catches all requests from the caching agents and feeds into a 128-entry second-level tracker with data buffers and address content-addressable memory that hold transactions active while they are processed in the MC.

Previous E7 systems relied on snoop broadcast for multisocket coherency. This had a considerable transaction overhead for glueless eight-socket systems and would have limited the socket performance scaling. To address this, IVB uses a home snoopy protocol with directory support. It maintains 2 bits of directory information for each cache line in memory, such that each memory read operation returns both the data and directory state. The directory provides information on whether the cache line is exclusive/modified, shared (new in IVB over SNB), or invalid in other sockets in the system. The directory reduces the snoop overhead needed to maintain memory coherence and improves the load to use latency for clean memory accesses (invalid or nonownership shared) by eliminating snoop processing from the critical path.

Although in-memory directory has advantages in socket scalability, maintaining the directory involves extra memory accesses, which cuts into the memory bandwidth available to the application. Additionally, it serializes the directory lookup and probing of the peer caches, which increases the latency for cache-to-cache transfer flows. To improve the cache-to-cache transfer latency, IVB implements an opportunistic snoop broadcast (OSB) option wherein the snoops are speculatively broadcast in parallel with the memory read or in lieu of the memory read if there is sufficient QPI link bandwidth. Heuristics exist to track OSB’s usefulness and to modulate OSB, dynamically biasing it for power and/or performance. OSB enables an optimal tradeoff between snooping and memory accesses to improve memory bandwidth delivered to applications. It improves latency for cache-to-cache transfers by removing the serialization overhead of directory lookup, without penalizing data return latency for clean directory accesses. We evaluated directory mode with OSB against a directory cache implementation and determined that it was the best option because it had a lower die area cost and reduced validation complexity while maintaining favorable performance and power characteristics.

IVB implements an I/O directory cache to minimize the directory overhead for writes to remote socket memory from an I/O agent by caching the directory state between the ownership phase and the actual write to memory, saving the directory lookup and update operations.

**IVB architecture**

The features we’ve described here come together to form the IVB die shown in Figure 3. This base die contains 15 cores and 15 LLC slices, two MCs with both native DDR3 and buffered memory capabilities, two HAs, two QPI agents with three total QPI links, an IIO unit with 40 PCIe Gen3 lanes, a power control unit, and a utility box managing the global noncoherent system flows and advanced RAS (reliability, availability, serviceability) features. Each of these components has a path to a stop on the internal ring interconnect.

**Scalability**

A key requirement for the IVB server is good performance scalability at higher core and socket counts. Thus far, we’ve presented the super-set architecture and its features; this section will outline how IVB scales the base die to cover the disparate server segments and product offerings within a segment. IVB products range from four to 15 cores, with a wide range of QPI and socket support. Table 1 provides a summary of the range of SKUs the IVB is required to cover.

To manage the entire SKU stack efficiently, IVB has built in cut lines or “chops” that create multiple products from the base die. As we mentioned earlier, the architecture is meant to be scalable, with identical functional behavior with various components removed. The control and routing algorithms are updated, but the baseline architecture mechanisms remain intact. This way, multiple smaller dies can be created from the base die with limited additional validation. Figure 4a captures the die photo of the base die. The three columns of cores and caches are visible...
from this view. The die photo in Figure 4b, which is covered with two different size structures, depicts the modular blocks of the design. The shaded box on the right covers the third column of cores and caches, the second QPI agent, and the second HA and MC, which can be removed to create a 10-core die (two-column design) with only two QPI links and one HA or MC. The horizontal shaded blocks cover the two rows of cores and caches on the 10-core die. With the removal of those two additional “chopable” blocks, a six-core die can be created. In addition to the “chopable” sections, IVB can also disable cores, LLC slices, either of the two memory controllers, QPI links, or PCIe lanes independently on all dies. Both these capabilities combined allow us create a rich SKU stack.

### Power and performance

Each Intel client and server CPU has power and performance scalability goals over prior-generation CPU products. The goal for IVB was to build a single design that would provide significant performance improvement over two prior-generation products, SNB Server E5 and WSM-EX Server E7, while maintaining power targets. IVB E5 v2 has the additional challenge of fitting in the same platform as a drop in compatible CPU refresh product.

### Generational power and performance scaling

There are two key attributes for power efficiency in a server product. The first is idle power, or how much power is drawn with no load. The second is proportional computing.
in which the power draw of the server product is proportional to the load. IVB uses Intel 22-nm process technology, which provides significant power improvements. The design also incorporates microarchitecture improvements, such as increased clock gating. Figure 5 depicts the power performance load-line benchmarking results generated using an internal workload (based on the industry standard SPECpower_ssj*2008 benchmark) at varying utilization levels. Figure 5a compares a few different IVB E5 v2 products (130-W 12-core, 95-W 10-core, and 70-W 10-core SKU parts) to SNB server E5 8-core, 95-W parts. Despite a significant core count increase over the prior generation (up to 50 percent), idle power remains the same or reduces over SNB, and equal or better performance is achieved at significantly lower power levels. The power increase happens at high demand levels as system performance moves up proportionally. Figure 5b illustrates the same workload results for the IVB E7 v2 15C 155W top SKU product compared to the
WSM E7 10C 130W baseline. The E7 graph shows the dramatic reduction in power consumption on v2 at all utilization levels and a greater than 2x performance increase at the 100 percent load level, with the extra benefit mainly due to incorporation of SNB generation turbo and power management capabilities.4

Memory bandwidth and latency scaling
Low latency and high throughput have been the guiding principles for the IVB architecture and design. The outcome is a substantial increase in memory bandwidth efficiencies and low memory access latencies at increased core counts. On IVB E7 v2, we get dramatic bandwidth improvements while keeping latencies low and flat at much higher bandwidth levels. We achieve up to 130 percent local memory bandwidth increase over WSM E7 with performance mode and a 50 percent increase with lock-step mode on a four-socket, all-cores-active local memory streaming workload.5 Similarly, IVB E5 v2 achieves higher bandwidths than SNB E5 on the same platform. On E5 v2, we measure 10 percent higher bandwidth at the same memory speed and 23 percent higher bandwidth at a 16.7 percent memory speed increase.5

Benchmark scaling
Figure 6 shows IVB E5 v2 and E7 v2 top SKU benchmark publications compared against SNB E5 and WSM E7 top SKU publications. Figure 6a captures the snapshot of the best external IVB E5 v2 and SNB E5 2-socket publications; up to 50 percent performance and energy efficiency is achieved over SNB E5. Figure 6b shows the same for four-socket IVB E7 v2 publications as compared to the best WSM E7 results. Note that for the E7 comparison the baseline is the WSM-core-based WSM E7 product—hence, the greater performance scaling.

The described IVB server architectural enhancements enabled us to hit our performance and scalability goals. We will continue to face challenges as we go to future-generation servers with a larger number of components to be integrated into one server die. The challenge lies not only in improving on-die and off-die interconnects but also in maintaining modularity to enable easy product variants up and down the server SKU stack. Hence, our future innovation is targeted at continuing to improve absolute performance as well as performance scaling efficiency.
Acknowledgments

We acknowledge the contributions of the entire Ivy Bridge architecture and design teams and our management. We also recognize the Sandy Bridge server team and the Ivy Bridge core team. It’s from their hard work that we could innovate further. We thank our partners in the BIOS, platform validation, software solutions group, and TME teams for their valued contributions to the product definitions and validation. Post-silicon debug and performance validation are critical to shipping a successful product; special thanks go to John Benoit for his System Debug Lab leadership and to Andrew Sun for his post-silicon performance validation contributions.

References


Irma Esmer Papazian is a principal engineer at Intel. She was the performance lead for the Ivy Bridge server. Her research interests include server architecture development, performance analysis and modeling.
techniques, pre- and post-silicon performance validation, and workload projections. Esmer Papazian has an MS in computer engineering from University of Wisconsin–Madison. Contact her at irma.esmer@intel.com.

Sailesh Kottapalli is an Intel Fellow. He was the lead architect for Ivy Bridge Server. His research interests include uncore architecture, multOCKET scalability, and server features. Kottapalli has an MS in computer engineering from Virginia Polytechnic Institute and State University. Contact him at sailesh.kottapalli@intel.com.

Jeff Baxter is a senior principal engineer at Intel. He leads the performance evaluation team. His research interests include architecture performance techniques and evaluation methodologies. Baxter has a PhD in electrical engineering from the University of Illinois. Contact him at jeff.baxter@intel.com.

Jeff Chamberlain is a cache and coherent fabric architect, focusing on high-end server products at Intel. He was lead architect for the Ivy Bridge server on-chip interconnect and LLC. Chamberlain has an MA in mathematics from the University of California, Los Angeles. Contact him at jeffrey.d.chamberlain@intel.com.

Geetha Vedaraman is a server coherence agent architect at Intel, where she was the lead home agent architect on the Ivy Bridge server, and also works on coherence protocol specifications. Vedaraman has an MS in computer science from the University of Wisconsin–Madison. Contact her at geetha.vedaraman@intel.com.

Brian Morris is a memory controller architect at Intel. His work focuses on improving the performance and reliability of the memory subsystem. Morris has an MS in electrical engineering from Purdue University. Contact him at brian.s.morris@intel.com.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.