THE AMD OPTERON NORTHBRIDGE ARCHITECTURE

To increase performance while operating within a fixed power budget, the AMD Opteron processor integrates multiple x86-64 cores with a router and memory controller. AMD’s experience with building a wide variety of system topologies using Opteron’s HyperTransport-based processor interface has provided useful lessons that expose the challenges to be addressed when designing future system interconnect, memory hierarchy, and I/O to scale with both the number of cores and sockets in future x86-64 CMP architectures.

In 2005, Advanced Micro Devices introduced the industry’s first native 64-bit x86 chip multiprocessor (CMP) architecture combining two independent processor cores on a single silicon die. The dual-core Opteron chip featuring AMD’s Direct Connect architecture provided a path for existing Opteron shared-memory multiprocessors to scale up from 4- and 8-way to 8- and 16-way while operating within the same power envelope as the original single-core Opteron processor.1,2 The foundation for AMD’s Direct Connect architecture is its innovative Opteron processor northbridge.

In this article, we discuss the wide variety of system topologies that use the Direct Connect architecture for glueless multiprocessing, the latency and bandwidth characteristics of these systems, and the importance of topology selection and virtual-channel-buffer allocation to optimizing system throughput. We also describe several extensions of the Opteron northbridge architecture, planned by AMD to provide significant throughput improvements in future products while operating within a fixed power budget. AMD has also launched an initiative to provide industry access to the Direct Connect architecture. The “Torrenza Initiative” sidebar summarizes the project’s goals.

The x86 blade server architecture

Figure 1a shows the traditional front-side bus (FSB) architecture of a four-processor (4P) blade, in which several processors share a bus connected to an external memory controller (the northbridge) and an I/O controller (the southbridge). Discrete external memory buffer chips (XMBs) provide expanded memory capacity. The single memory controller can be a major bottleneck, preventing faster CPUs or additional cores from improving performance significantly.

In contrast, Figure 1b illustrates AMD’s Direct Connect architecture, which uses industry-standard HyperTransport technol-
Torrenza Initiative

AMD’s Torrenza is a multiyear initiative to create an innovation platform by opening access to the AMD64 Direct Connect architecture to enhance acceleration and coprocessing in homogeneous and heterogeneous systems. Figure A shows the Torrenza platform, illustrating how custom designed accelerators, say for the processing of Extensible Markup Language (XML) documents or for service-oriented architecture (SOA) applications, can be tightly coupled with Opteron processors. As the industry’s first open, customer-centered x86 innovation platform, Torrenza capitalizes on the Direct Connect architecture and HyperTransport technology advances of the AMD64 platform.

The Torrenza Initiative includes the following elements:

- **Innovation Socket.** In September 2006 AMD announced it would license the AMD64 processor socket and design specifications to OEMs to allow collaboration on specifications so that they can take full advantage of the x86 architecture. Cray, Fujitsu, Siemens, IBM, and Sun have publicly stated their support and are designing products for the Innovation Socket.

- **Coprocessor enablement.** Leveraging the strengths of HyperTransport, AMD is working with various partners to create an extensive partner ecosystem of tools, services, and software to implement coprocessors in silicon. HyperTransport is the only open, standards-based, extensible system bus.

- **Direct Connect platform enablement.** AMD is encouraging standards bodies and operating system suppliers to support accelerators and coprocessors directly connected to the processor. To help drive innovation across the industry, AMD is opening access to HyperTransport.

Torrenza is designed to create an opportunity for a global innovation community to develop and deploy application-specific coprocessors to work alongside AMD processors in multisocket systems. Its goal is to help accelerate industry innovation and drive new technology, which can then become mainstream. It gives users, original equipment manufacturers, and independent software vendors the ability to leverage billions in third-party investments.
Northbridge microarchitecture

In the Opteron processor, the northbridge consists of all the logic outside the processor core. Figure 2 shows an Opteron processor with a simplified view of the northbridge microarchitecture, including system request interface (SRI) and host bridge, crossbar, memory controller, DRAM controller, and HyperTransport ports.

The northbridge is a custom design that runs at the same frequency as the processor core. The command flow starts in the processor core with a memory access that misses in the L2 cache, such as an instruction fetch. The SRI contains the system address map, which maps memory ranges to nodes. If the memory access is to local memory, an address map lookup in the SRI sends it to the on-chip memory controller; if the memory access is off-chip, a routing table lookup routes it to a HyperTransport port.

The northbridge crossbar has five ports: SRI, memory controller, and three HyperTransport ports. The processing of command packet headers and data packets is logically separated. There is a command crossbar dedicated to routing command packets, which are 4 or 8 bytes in size, and a data crossbar for routing the data payload associated with commands, which can be 4 or 64 bytes in size.

Figure 3 depicts the northbridge command flow. The command crossbar routes coherent HyperTransport commands. It can deliver an 8-byte HyperTransport packet header at a rate of 1 per clock (one every 333 ps with a 3-GHz CPU). Each input port has a pool of command-size buffers, which are divided between four virtual channels (VCs): Request, Posted request, Probe, and Response. A static allocation of command buffers occurs at each of the five crossbar input ports. (The next section of this article discusses how buffers should be allocated across different virtual channels to optimize system throughput.)

The data crossbar, shown in Figure 4, supports cut-through routing of data packets. The cache line size is 64 bytes, and all buffers are sized in multiples of 64 bytes to optimize the transfer of cache-line-size data packets. Data packets traverse on-chip data paths in 8 clock cycles. Transfers to different output ports are time multiplexed.

Figure 1. Evolution of x86 blade server architecture: traditional front-side bus architecture (a) and AMD’s Direct Connect architecture (b). MCP: multichip package; Mem.: memory controller.
Figure 2. Opteron 800 series processor architecture.

clock by clock to support high concurrency; for example, two concurrent transfers from CPU and memory controller input ports to different output ports are possible. The peak arrival rate from a HyperTransport port is 1 per 40 CPU clock cycles or 16 ns (64 bytes at 4 Gbytes/s), and the on-chip service rate is 1 per 8 clock cycles, or 3 ns.

HyperTransport routing is table driven to support arbitrary system topologies, and the crossbar provides separate routing tables for routing Requests, Probes, and Responses. Messages traveling in the Request and Response VCs are always point-to-point, whereas messages in the Probe VC are broadcast to all nodes in the system.

**Coherent HyperTransport protocol**

The Opteron processor northbridge supports a coherent shared-memory address space. AMD’s development of a coherent HyperTransport was strongly influenced by prior experience with the Scalable Coherent Interface (SCI), the Compaq EV6, and various symmetric multiprocessor systems. A key lesson guiding the development of the coherent HyperTransport protocol was that the high-volume segment of the server market is two to four processors, and although supporting more than four processors is important, it is not a high-volume market segment. The SCI protocol supports a single shared-address space for an arbitrary number of nodes in a distributed shared-memory architecture. It does so through the creation and maintenance of lists of sharers for all cached lines in doubly linked queues with mechanisms for sharer insertion and removal. The protocol is more complex than required for the volume server market, and its wide variance in memory latency would require a lot of application tuning for nonuniform memory access. On the other hand, bus-based systems with snoopy bus protocols can achieve only limited transfer rates. The coherent HyperTransport protocol was designed to support cache coherence in a distributed shared-memory system with an arbitrary number of nodes using a broadcast-based coherence protocol. This provides good scaling in the one-, two-, four-, and even eight-socket range, while avoiding the serialization overhead, storage overhead, and complexity of directory-based coherence schemes.

In general, cacheable processor requests are unordered with respect to one another in the coherent HyperTransport fabric. Each processor core must maintain the program order of its own requests. The Opteron processor core implements processor consistency, in which loads and stores are always
ordered behind earlier loads, stores are ordered behind earlier stores, but loads can be ordered ahead of earlier stores (all requests to different addresses). Opteron-based systems implement a total-store-order memory-ordering model.\textsuperscript{7} The cores and the coherent HyperTransport protocol support the MOESI states: \textit{modified}, \textit{owned}, \textit{exclusive}, \textit{shared}, and \textit{invalid}, respectively.\textsuperscript{6}

Figure 5 is a transaction flow diagram illustrating the operation of the coherent HyperTransport protocol. It shows the message flow resulting from a cache miss for a processor fetch, load, or store on node 3. Initially, a request buffer is allocated in the SRI of source node 3. The SRI looks up the system address map on node 3, using the physical address to determine that node 0 is the home node for this physical address. The SRI then looks up the crossbar routing table, using destination node 0 to determine which HyperTransport port to forward the read request (RD) to. Node 2 forwards RD to home node 0, where the request is delivered to the memory controller. The memory controller starts a DRAM access and broadcasts a probe (PR) to nodes 1 and 2. Node 1 forwards the probe to source node 3. The probe is delivered to the SRI on each of the four nodes. The SRI probes the processor cores on each node and combines the probe responses from each core into a single probe response (RP), which it returns to source node 3 (if the line is \textit{modified} or \textit{owned}, the SRI returns a read response to the source node instead of a probe response).

Once the source node has received all probe and read responses, it returns the fill data to the requesting core. The request buffer in the SRI of source node 3 is deallocated, and a source done message (SD) is sent to home node 0 to signal that all the transaction’s side effects, such as invalidating all cached copies for a store, have completed and the data is now globally visible. The memory controller is then free to process a subsequent request to the same node.
Figure 4. Northbridge data flow. All buffers are 64-byte cache lines.

Figure 5. Traffic for an Opteron processor read transaction.
address. The memory latency of a request is the longer of two paths: the time it takes to access DRAM and the time it takes to probe all caches in the system.

The coherent HyperTransport protocol message chain is essentially three messages long: Request → Probe → Response. The protocol avoids deadlock by having a dedicated VC per message class (Request, Probe, and Response). Responses are always unconditionally accepted, ensuring forward progress for probes, and in turn ensuring forward progress for requests.

One unexpected lesson that emerged during the bring-up and performance tuning of Opteron multiprocessors was that improper buffer allocation across VCs has a surprising negative impact on performance. Why? The Opteron microprocessor has a flexible command buffer allocation scheme. The buffer pool can be allocated across the four VCs in a totally arbitrary way, with only the requirement that each VC have at least one buffer allocated. The optimum allocation turns out to be a function of the number of nodes in a system, the system topology, the coherence protocol, the relative mix of different transaction and the routing tables. As a rule, the number of buffers allocated to the different VCs should be in the same proportion as the traffic on these VCs. After exhaustive traffic analysis, factoring in the cache coherence protocol, the topology, and the routing tables, we determined optimum BIOS settings for four- and eight-node topologies.

**Opteron-based system topologies**

The Opteron processor was designed to have low memory latency for one-, two-, and four-node systems. For example, a four-node machine’s worst-case latency (two hops) is lower than that typically achievable with an external memory controller. Even so, a processor’s performance is a strong function of system topology, as Figure 6 shows. The figure illustrates the performance scaling achieved for five commercial workloads on five common Opteron system topologies. The figure shows the topologies with different node counts, along with their average network diameter and memory latency. For example, the four-node topology (“4-node square”) is a 2 × 2 2D mesh with a network diameter of 2 hops, an average diameter of 1 hop, and an average memory latency of $44 \text{ ns}$, where $x$ is the latency of a one-node system using a 2.8-GHz processor, a 400-MHz DDR2 PC3200 memory, and a HyperTransport-based processor interface operating at 2 giga-transfers per second (GT/s). The system performance is normalized to that of one node × one core. We see positive scaling from one to eight nodes, but the normalized processor performance decreases with increasing average diameter. The difference in normalized processor performance among this set of workloads is mainly due to differences in L2 cache miss rates. SPECjbb2000 has the lowest miss rate and the best performance scaling, whereas OLTP1 has the highest L2 miss rate and the worst performance scaling. It is worth noting that processor performance is a strong function of average diameter. For example, the processor performance in the eight-node twisted ladder, with a 1.5 hop average diameter, is about 10 percent higher than in the eight-node ladder (a 2 × 4 2D mesh), with a 1.8 hop average diameter. This observation strongly influenced our decision to consider fully connected 4- and 8-node topologies in AMD’s next-generation processor architecture.

The most direct way to reduce memory latency and increase coherent memory bandwidth is to use better topologies and faster links. The argument for fully connected topologies is simple: The shortest distance between two points is a direct path, and fully connected topologies provide a direct path between all possible sources and destinations.

Future generations of Opteron processors will have a new socket infrastructure that will support HyperTransport 3.0 with data rates of up to 6.4 GT/s. We will enable fully connected four-socket systems by adding a fourth HyperTransport port, as Figure 7 shows. We will enable fully connected eight-socket systems by supporting a feature called HyperTransport link unganging as shown in Figure 8. A HyperTransport link is typically 16 bits wide (denoted ×16) in
Each direction. These same HyperTransport pins can also be configured at boot time to operate as two logically independent links, each 8 bits wide (denoted \( \times 8 \)). Thus, the processor interface can be configured to provide a mix of \( \times 16 \) and \( \times 8 \) HyperTransport ports, each of which can be configured to be either coherent or noncoherent. Link unganging provides system builders with a high degree of flexibility by expanding the number of logical HyperTransport ports from 4 to 8.

Fully connected topologies provide several benefits: Network diameter (memory latency) is reduced to a minimum, links are more evenly utilized, packets traverse fewer

Figure 6. Performance versus memory latency in five Opteron topologies (systems use a single 2.8-GHz core, a 400-MHz DDR2 PC3200, a 2-GT/s HyperTransport, and a 1-Mbyte L2 cache).
links, and there are more links. Reduced link utilization lowers queuing delay, in turn reducing latency under load. Two simple metrics for memory latency and coherent memory bandwidth demonstrate the performance benefit of fully connected multiprocessor topologies:

- **Average diameter**—average number of hops between any two nodes in the network (network diameter is the maximum number of hops between any pair of nodes in the network).
- **Xfire memory bandwidth**—link-limited, all-to-all communication bandwidth (data only). All processors read data from all nodes in an interleaved manner.

We can statically compute these two metrics for any topology, given routing tables, message visit counts, and packet sizes.

Figure 7. Four-socket, 16-way topologies: The 4-node square topology (a) has a network diameter of 2 hops, an average diameter of 1.0 hop, and a Xfire bandwidth using 2.0 GT/s HyperTransport of 14.9 Gbytes/s. The 4-node fully connected topology (b) with two extra links and a fourth HyperTransport port yields a network diameter of 1 hop, an average diameter of .75 hop, and a Xfire bandwidth of 29.9 Gbytes/s. Using HyperTransport 3.0 at 4.4 GT/s, that topology achieves a Xfire bandwidth of 65.8 Gbytes/s.

Figure 8. Eight-socket, 32-way topologies. The 8-node twisted-ladder topology (a) has a network diameter of 3 hops, an average diameter of 1.62 hops, and a Xfire bandwidth using HyperTransport 1 at 2.0GT/s of 15.2 Bytes/s. The 8-node 2×4 topology (b) has a diameter of 2 hops, an average diameter of 1.12 hops, and a Xfire bandwidth using HyperTransport 3.0 at 4.4GT/s of 72.2 Gbytes/s. The 8-node fully connected topology (c) has a diameter of 1 hop, an average diameter of 0.88 hop, and a Xfire bandwidth using HyperTransport 3.0 at 4.4GT/s of 94.4 Gbytes/s.
the four-node system in Figure 7, adding one extra HyperTransport port doubles the XFire bandwidth. In addition, this number scales linearly with link frequency. Thus, with HyperTransport 3.0 running at 4.4 GT/s, the XFire bandwidth increases by a factor of 4 overall. In addition, the average diameter (memory latency) decreases from 1 hop to 0.75 hop.

The benefit of fully connected topologies is even more dramatic for the eight-node topology in Figure 8. The XFire bandwidth increases by a factor of 6 overall. In addition, the average diameter decreases significantly from 1.6 hops to 0.875 hop. Furthermore, this access pattern, which is typical of many multithreaded commercial workloads, evenly utilizes the links.

**Next-generation processor architecture**

AMD’s next-generation processor architecture will be a native quad-core upgrade that is socket- and thermal-compatible with the Opteron processor 800 series socket F. It will contain about 450 million transistors and will be manufactured in a 65-nm CMOS silicon-on-insulator process. At some point, AMD will introduce a four-HyperTransport-port version in a 1,207-contact, organic land grid array package paired with a surface-mount LGA socket with a 1.1-mm pitch and a 40 x 40-mm body.

Core enhancements include out-of-order load execution, in which a load can pass other loads and stores that are known not to alias with the load. This mitigates L2 and L3 cache latency. The translation look-aside buffer adds support for 1-Gbyte pages and a 48-bit physical address. The TLB’s size increases to 512 4-Kbyte page entries plus 128 2-Mbyte page entries for better support of virtualized workloads, large-footprint databases, and transaction processing.

The design provides a second independent DRAM controller to provide more concurrency, additional open DRAM banks to reduce page conflicts, and a longer burst length to improve efficiency. DRAM paging support in the controller uses history-based pattern prediction to increase the frequency of page hits and decrease page conflicts. The DRAM prefetcher tracks positive, negative, and non-unit strides and has a dedicated buffer for prefetched data. Write bursting minimizes read and write turnaround time.

The design has a three-level cache hierarchy as shown in Figure 9. Each core has separate L1 data and instruction caches of 64 Kbytes each. These caches are two-way set-associative, linearly indexed, and physically tagged, with a 64-byte cache line. The L1 has the lowest latency and supports two 128-bit loads per cycle. Locality tends to keep the most critical data in the L1 cache. Each core also has a dedicated 512-Kbyte L2 cache, sized to accommodate most workloads. This cache is dedicated to eliminating conflicts common in shared caches and is better than shared caches for virtualization. All cores share a common L3 victim cache that resides logically in the northbridge SRI unit. Cache lines are installed in the L3 when they are cast out from the L2 in the processor core. The L3 cache is noninclusive, allowing a line to be present in an upper level L1 or L2 cache and not be present in the L3. This increases the maximum number of unique cache lines that can be cached on a node to the sum of the individual L3, L2, and L1 cache capacities (in contrast, the maximum number of distinct cache lines that can be cached with an inclusive L3 is simply the L3 capacity). The L3 cache has a sharing-aware replacement policy to optimize the movement, placement, and replication of data for multiple cores.

As Figure 10 shows, the next-generation design has seven clock domains (phase-locked loops) and two separate power planes for the northbridge and the core. Separate CPU core and northbridge power planes allow processors to reduce core voltage for power savings while the northbridge continues to run, thereby retaining system bandwidth and latency characteristics. For example, core 0 could be running at normal operating frequency, core 1 could be running at a lower frequency, and cores 2 and 3 could be halted and placed in a low-power state. It is also possible to apply higher voltage to the northbridge to raise its frequency for a performance boost in power-constrained platforms.
Fine-grained power management (enhanced AMD PowerNow! technology) provides the capability of dynamically and individually adjusting core frequencies to improve power efficiency.

Figure 9. AMD next-generation processor’s three-level cache hierarchy.

In summary, the AMD Opteron processor integrates multiple x86-64 cores with an on-chip router, memory controller and HyperTransport-based processor interface. The benefits of this system integration include lower latency, cost, and power use. AMD’s next-generation processor extends the Opteron 800 series architecture by adding more cores with significant instructions per cycle (IPC) enhancements, an L3 cache, and fine-grained power management to create server platforms with improved memory latency, higher coherent memory bandwidth, and higher performance per watt.

Acknowledgments

We thank all the members of the AMD Opteron processor northbridge team, including Nathan Kalyanasundharam, Gregg Donley, Jeff Dwork, Bob Aglietti, Mike Fertig, Cissy Yuan, Chen-Ping Yang, Ben Tsien, Kevin Lepak, Ben Sander, Phil Madrid, Tahsin Askar, and Wade Williams.
References

Pat Conway is a principal member of technical staff at AMD where he is responsible for developing scalable, high-performance server architectures. His work experience includes the design and development of server hardware, cache coherence and message passing protocols. He has a M.Eng.Sc from University College Cork, Ireland, and an MBA from Golden Gate University. He is a member of the IEEE.

Bill Hughes is a senior fellow at AMD. He was one of the initial Opteron architects working on HyperTransport and the on-chip memory controller and also worked on the load-store and data cache units on Athlon. He currently leads the Northbridge and HyperTransport microarchitecture and RTL team. He has a BS from Manchester University, England, and a PhD from Leeds University, England.

Direct questions and comments about this article to Pat Conway, Advanced Micro Devices, 1 AMD Place, Sunnyvale, CA 94085; pat.conway@amd.com.

For further information on this or any other computing topic, visit our Digital Library at http://www.computer.org/publications/dlib.

Engineering and Applying the Internet

IEEE Internet Computing reports emerging tools, technologies, and applications implemented through the Internet to support a worldwide computing environment.

In 2007, we'll look at:
- Autonomic Computing
- Roaming
- Distance Learning
- Dynamic Information Dissemination
- Knowledge Management
- Media Search

www.computer.org/internet/