The Opteron Microprocessor

Tom Van Cutsem
tvcutsem@vub.ac.be
Vrije Universiteit Brussel

November 30, 2003

Abstract

This document gives a broad overview of AMD's [1] [Dev03a] Opteron microprocessor. The document highlights several parts of the architecture, such as its 64-bit features, its memory hierarchy and its pipeline architecture. Furthermore, the Opteron is analyzed in the context of multiprocessors, featuring Hypertransport [Con03] as an innovative communication link.

Contents

[1] Advanced Micro Devices

1 Introduction

The Opteron[2] microprocessor is an implementation of AMD's newest, 8^{th} generation *Hammer Architecture* announced to finally "bring 64-bit computing to the desktop world". The Hammer, also marketed as the AMD64 Architecture, has spawned some interesting microprocessors (section 2) of which the Opteron is the most powerful and will have to match Intel's Itanium family of microprocessors.

The Opteron is an out-of-order, 3-way superscalar processor, meaning it can decode, execute and retire three x86-instructions per cycle. Each clock tick the Opteron can work on three instructions in parallel, although this of course does not mean that the instructions are entirely processed in one cycle. Section 3.2 will explain in detail how x86-instructions will flow through the microprocessor.

The Opteron was designed as a server or enterprise microprocessor, which explains why AMD has architected it as a multiprocessor. Section 6 explains the capabilities of the Opteron in such a multiprocessor setup, and the importance of Hypertransport (section 5) in this context. The Opteron has been cited as the "first-ever multiprocessor designed for the commercial market".

2 Model Overview

This section highlights the general Hammer Architecture before elaborating on the details of the more Opteron-specific architecture.

2.1 The Hammer Architecture

AMD has scheduled two versions of its Hammer Architecture, named ClawHammer and SledgeHammer [Lab02]. The first is intended for desktop PCs and low-end dual-processor servers, marketed under the well-known Athlon name. SledgeHammer focuses on the server-side of the market and is marketed under the name of Opteron.

[2]Opteron is derived from the Latin word *Optimus* meaning "the best"

The Sledgehammer models are known as the Athlon64 and the Athlon64-FX. Both are "scaled down" versions of the Opteron, suited for desktop models and targeting especially the gaming audience. The Athlon64-FX shares more features with the Opteron than the Athlon64, such as a 128-bit connection with the memory, as opposed to Athlon64's 64-bit connection. Also, the Athlon64 has only a 754-pin CPU package, as opposed to Athlon64-FX and Opteron's 940-pin packages.

All Hammer processors feature clock speeds up to 2GHz. This is much less than Intel's Xeon family, who have already surpassed the 3Ghz boundary. However, customers for the server market are usually more interested in performance, scalability, and architecture than plain core speed. Moreover, the higher the clock speed, the more problems a microprocessor architecture runs into (like heat conduction).

Compared to the seventh-generation Athlon and Athlon XP, the Hammer brings about two more pipeline stages, enhanced branch-prediction algorithms, support for SSE2 streaming multimedia instructions, a memory controller integrated into the CPU instead of located in the Northbridge (see section 4.1) and a full 64-bit extension of the x86 instruction set.

2.2 Opteron Models

The Opteron microprocessor itself has different available models, ranging in clock speed and performance. The Opterons are manufactured at Fab 30 in Dresden, Germany. They are produced on a 0.13 micron scale and are expected to be produced on a 0.09 micron (90-nanometer) fabrication process in the near future.

The Opteron series are logically subdivided into the 100-, 200- and 800-series, the first digit indicating the maximum number of processors supported in a multiprocessor setup. Thus, the Opteron-100 series are useful for uniprocessor workstations, while the 200 series can form 2-way dual-processor workstations and the 800-series can form an 8-way cluster [Fre03]. The core clock speeds of all three models range from 1.4Ghz up to 2.0Ghz.

The last two digits of each model number indicate relative performance within the series itself. Whether this performance gain is related to a higher clock speed, a bigger cache or a faster Hypertransport link, AMD does not specify.

3 CPU Architecture

3.1 The world of 64-bit computing

Perhaps the most remarkable feature in comparison with AMD's 7^{th} generation of microprocessors is the transcendence to 64-bit computing. This implies that the Opteron is able to address much more memory than its predecessors. Opteron uses 40-bit physical and 48-bit virtual addressing, thereby being able

to address up to one terabyte of physical memory and 256 terabytes of virtual addressing space, in contrast to the 4 Gigabyte limit imposed by most 32-bit processors.

What users would benefit from having a 64-bit processor? CAD tools, large databases and complex simulation software tools often require over 4 Gb of RAM. A lot of scientific calculations involve 64-bit representations and also require large amounts of RAM. Cryptographic applications will especially benefit from 64-bit integers [Kar03].

Apart from better memory addressability, having a 64-bit core also allows for extending the instruction set. At this point, the differences between AMD's and Intel's 64-bit strategy are significantly different [De 02]. AMD has chosen only to **extend** the x86 instruction set with new 64-bit features. This evolutionary approach ensures backward compatibility with all 32-bit x86 software written thus far, which is important from an economic point of view. The Opteron can thus be profiled as allowing a smooth transition into the world of 64-bit computing.

On the other hand Intel, featuring the IA-64 has adopted to break with the x86 instruction set, but allows for emulation (via an x86-to-IA-64 decoder [De 02]) of legacy x86 code. Obviously, such emulation scheme will have its effect on performance. In the Opteron, a legacy x86 program will be able to run at full speed and will just not use the added features of the 64-bit architecture.

3.1.1 Opmodes

The Opteron is quite flexible in its compatibility with older 16- or 32-bit applications. The microprocessor features a number of opmodes to handle compatibility issues.

The operation modes of the Opteron are subdivided into two large categories:

Legacy Mode In Legacy Mode, the Opteron acts as though it was an ordinary 32-bit processor. It thus provides compatibility with 16- and 32-bit Operating Systems and applications.

Long Mode Long Mode is further subdivided into *Compatibility Mode* and *64-bit Mode*. Both require a 64-bit Operating System.

- Compatibility mode provides binary compatibility for existing 16- and 32-bit applications on a 64-bit OS. Although running applications view the processor as a 32-bit processor, the OS itself works at the 64-bit level.
- Full 64-bit mode gives access to all the advantages of the 64-bit architecture, including 64-bit memory addressability for programs, as well as access to the extended x86 instruction set, discussed in the next section.

3.1.2 Instruction Set changes

AMD includes instruction set support at the 64-bit level very naturally by just extending the 32-bit architecture in a simple way. The number of general-purpose registers are extended from 8 up to 16. Also, 8 new registers are added to the Streaming SIMD extensions (SSE) unit for the support of SSE2. Furthermore the old registers are widened from 32 to 64 bits, the same goes for the instruction pointer. Programmers can address the new registers via a new instruction prefix byte called REX (Register extension). Figure 1 shows additions of the x86-64 architecture.

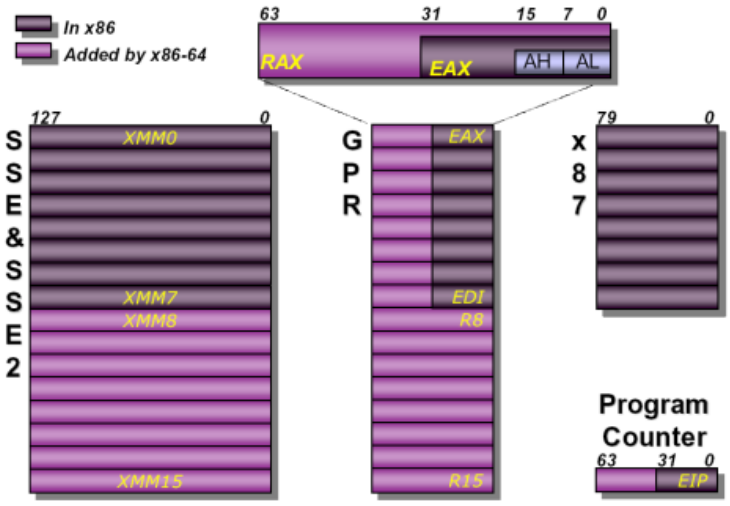

Figure 1: Extended x86 instruction set architecture [Web01]

x86-64 also features a single contiguous address space where instructions, data and stack are all stored in the same place. We go into some more detail in section 4.4.

In contrast to Intel's Itanium, AMD has not added a vast amount of registers. While the Opteron is now equipped with 16 programmable registers, Intel's Itanium features 128 integer and 128 floating-point registers. AMD's argument is that more than 80 % of current-day code uses only a maximum of 16 registers [KMAC03].

3.2 Detailed Architecture

The Opteron, just like Intel's Itanium, has 9 execution units, featuring three Arithmetic and Logic units (ALUs), three Address Generation Units (AGUs) and three Floating Point units. Opteron features a 12-stage Integer pipeline and a 17-stage Floating point pipeline. These will both be discussed in a little more detail below. Figure 2 gives a general overview of Hammer's architecture, on which the Opteron is based.

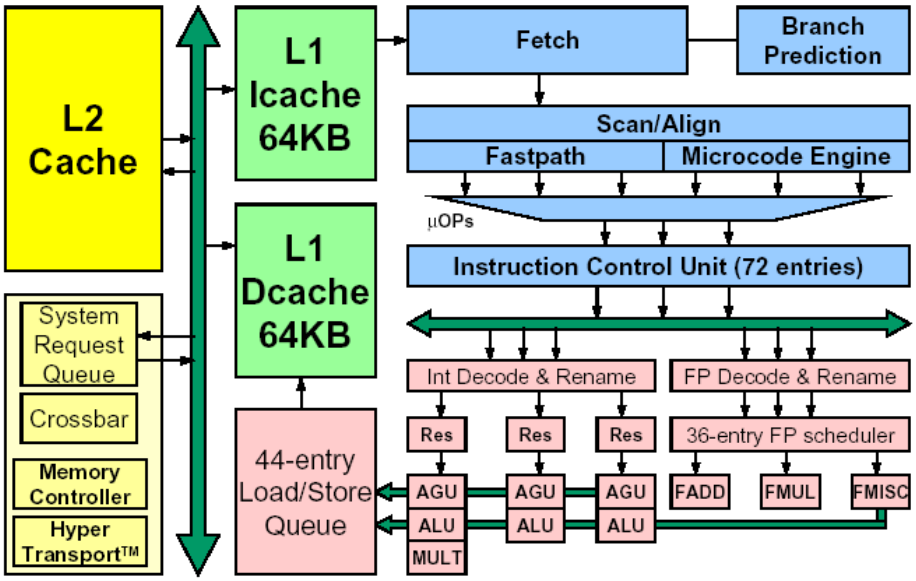

Figure 2: Hammer Core Overview [Kel02]

Opteron's main pipeline can be subdivided into a *fetch* pipeline, where instructions are fetched and decoded into smaller RISC-like micro-ops. Next, the *execute* pipeline performs the useful work in the execution units, after which the Opteron enters its *L2* and *DRAM* pipelines to communicate the results to memory. Figure 3 gives an overview of the most important stages for the integer and floating point pipelines.

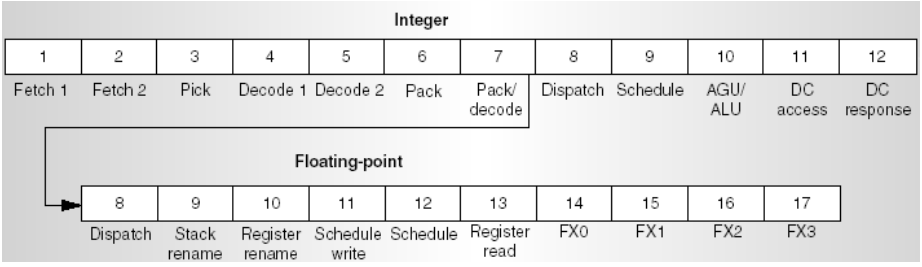

Figure 3: Integer and floating point pipeline stages [KMAC03]

3.2.1 Integer Pipeline

Instruction Fetch Each cycle, up to 16 instruction bytes can be handled by the decoder. Also, at this point, branch prediction in case of branch instructions is performed. This will be explained in section 3.3.

Instruction Decode One of the points in the architecture where the Opteron is improved compared to the Athlon processors is the addition of two more decoding stages. During this stage, x86 instructions are transformed into internal macro-ops. Instructions can be decoded in two different ways: either by using the hardware *fastpath* or *directpath* decoder or by using a small microprogram (called the *Microcode instruction sequencer*), also called the 'vector path' for vector instructions. Most simple x86 instructions can use the fastpath decoder.

The more complex x86 instructions (usually of variable length) are decoded into fixed-length macro-ops by the MIS. In comparison with the Athlon processor, the extra stages pay off: more instructions now use the fastpath decoder (even SSE/SSE2 instructions use this decoder [de 03]). Worth noting is that AMD scaled down every instruction uniformly to 64-bits. This means that 128-bit SSE instructions are decoded into two separate 64-bit instructions.

Note that such transformation into fixed-length gives the Opteron a RISC flavour, since the macro-ops are much simpler than x86 instructions [Les03].

Each cycle, three x86 instructions can be decoded simultaneously by three different decoding pipelines.

Dispatching and scheduling Each cycle, the decoders send macro-ops to three 8-entry schedulers (called *reservation stations*) where they are packed in groups of three. Each group of macro-ops then gets dispatched over the three integer pipelines. However, before they get dispatched, the macro-ops are further subdivided into micro-ops, the most elementary operations [Dev03c]. Each cycle, one macro-op can give rise to two micro-ops. A micro-op can be either an integer operation or an address-generation operation.

Since each integer pipeline has its own ALU and AGU who can simultaneously execute a micro-operation, it can issue two micro-ops per cycle, giving rise to a maximum of six active micro-ops per cycle. The ALU and AGU of one integer pipe can even execute two micro-ops that were issued from two different macro-ops in parallel. Figure 4 gives a general overview of this part of the pipeline.

Instruction Execution During the execution phase, the execution devices (AGU and ALU) perform the real work. The Opteron can execute all logical integer operations (add, shift, logic, ...) except for multiplication in one clock cycle, on 64-bit data. 32-bit multiplication takes 3 cycles, while 64-bit multiplication takes 5 cycles. All operations can be started every clock, except for 64-bit multiplication which can only be started every second cycle [de 03].

Each of the three ALUs performs general purpose arithmetic. An AGU typically calculates logical addresses from the base register and some index, also taking into account program segments etc.

Instruction Retirement Retirement is not so straightforward in processors that allow out-of-order execution (this out-of-order execution was introduced by the schedulers, who launch micro-ops whenever their operands and the execution

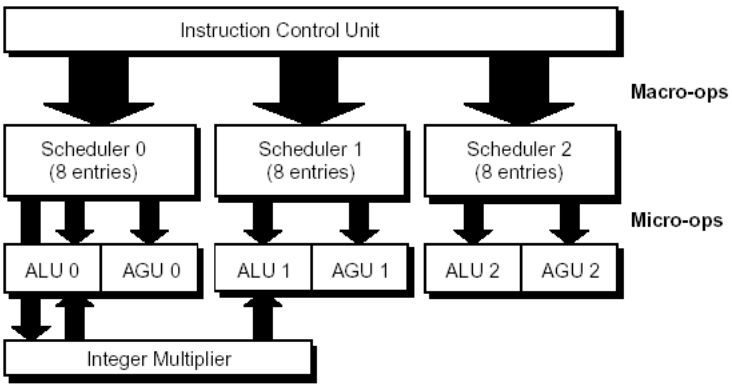

Figure 4: The integer execution pipeline [Dev03c]

unit are available). Retirement is handled by the use of a *reorder buffer* [de 03]. This buffer re-orders the instructions back into the original program sequence.

The buffer also makes sure no exceptions (eg. interrupts, branch-miss-prediction, ...), have occurred since the issuing of the instructions. Recall that complex x86 instructions were broken down in several multiple macro-ops. The buffer must make sure that the micro-ops generated for such vector path instructions are retired simultaneously.

3.2.2 Floating Point Pipeline

Up to the scheduling stage, the floating-point pipeline coincides with the integer pipeline. From there on, instead of the 3x8-entry integer schedulers, the floating point pipeline has one 36-entry scheduler organized as twelve lines of three macro-ops each. After registers are renamed (to minimize data dependencies), execution proceeds to the floating-point execution unit (the FPU).

The floating-point unit is a fully pipelined, superscalar, out-of-order execution unit. It is capable of accepting three macro-ops per cycle and can handle a mixture of different types of instructions (x87 floating point, MMX, 3DNow!, SSE and SSE2). The FPU consists of a stack renaming unit, a register renaming unit, a scheduler, a register file, and three parallel execution units [de 03]:

- the adder pipe (FADD), which contains an MMX ALU/shifter and floating-point *add* execution units.
- the multiplier (FMUL), which contains the floating-point multiplier, divider, square root unit and an MMX ALU.
- the floating-point load/store (FMISC) unit, which handles floating-point stores. It is also responsible for float to int and int to float conversion.

Figure 5 shows the layout of the FPU.

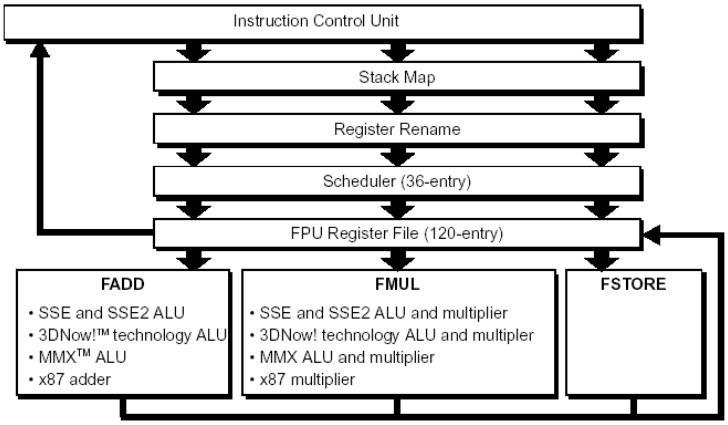

Figure 5: Floating-point unit architecture [Dev03c]

Floating-point instructions receive their operands by reading them from the *renamed register file*. This register file can be accessed directly after an instruction is dispatched by the scheduler. Up to three instructions can access the register file simultaneously (one instruction for each of the three execution units).

3.3 Branch Prediction

A significant amount of work was put into the improvement of the Branch Prediction unit (BPU). The longer the execution pipeline, the more important branch prediction becomes, since a misprediction will lead to a bigger pipeline stall. AMD realized this, since Branch Prediction has been dramatically improved in comparison with the Athlon processors.

First of all, the *global history counter buffer* is now four times bigger, compared to the Athlon [Lab02]. This buffer stores 16 kbytes of transitions. This makes the Opteron "remember" more branches and predict future branches more accurately. It also keeps 2kbytes of previous destination addresses (*branch targets*) in memory.

The global history counter is a large array of 2-bit counters [de 03]. The counters determine whether or not a branch should be taken. When a branch instruction enters, the BPU uses the address of the branch instruction as an index into the global history counter buffer. The BPU next examines the counter and if its value is ≥ 2 the branch is predicted as "taken". If this choice leads to a misprediction, the counter is decremented by one, otherwise it is incremented by one [de 03].

Since a 2-bit counter can only count from 0 to 3, there are four possible states, being *strongly not taken, weakly not taken, weakly taken* and *strongly taken*. This scheme of incrementing the counter with successful predictions works very good with looping structures.

One problem with this scheme is that two branch instructions can *interfere* when they "hash" to the same index value, thereby inspecting and changing the counter without any semantic value. The bigger the global history counter, the less likely interference will occur.

Finally, the Branch Prediction unit also contains a 12-entry *return address stack* [KMAC03]. Such RAS optimizes `CALL/RETURN` instruction pairs by storing the return address of each `CALL` and retrieving it when the `RETURN` is executed.

4 Memory Hierarchy

4.1 Integrated Memory Controller

Historically, a CPU chip worked together with (usually two) other chips on the motherboard. These chips are usually called the *northbridge* and the *southbridge*. The northbridge is responsible mostly for memory and AGP access. The southbridge manages other (usually slower) peripherals like hard disks, PCI devices, USB devices, etc...

The northbridge, being responsible for memory access, plays a crucial role in the performance of a computer system. This is why AMD has decided to *integrate* this northbridge functionality directly on the chip itself. The main advantage of this operation is a significant reduction in memory access latency. Less cycles are needed for accessing the main memory. The results are promising, given that memory access has improved by approximately 20% compared to the Athlon processor [Kel02].

The Integrated DDR Memory Controller has a 128-bit connection to system memory. Moreover, it runs at the same frequency as the CPU itself. This allows memory bandwidth and capacity to improve whenever the CPU speed improves. Moreover, it automatically scales memory bandwidth as the number of CPUs themselves increases, since adding a CPU also implies adding a memory controller. The memory controller also handles cache coherence.

All Opteron chips include an integrated memory controller capable of supporting DDR200, DDR266, or DDR333 SDRAM, with a peak bandwidth of up to 5.3GBytes/sec for DDR333 [KMAC03].

Other northbridge functionality, like AGP communication is moved to an external chipset (the AMD-8151 Hypertransport AGP Graphics Tunnel, see section 5.3). The integrated memory controller is connected to the motherboard through three Hypertransport links.

4.2 L1-Cache

The Opteron features a Level 1 cache of 128 kbytes, physically and logically subdivided into a 64 kbyte instruction cache and data cache. The L1 caches are two-way set associative and have a block size of 64 bytes.

The L1 cache can be accessed twice each clock cycle. These accesses can be loads and/or stores. Each L1 cache actually consists of 8 individual banks.

Two accesses can occur simultaneously only if their addresses are located in different banks. 64 byte cache blocks are subdivided into 8 64-bit banks. Since the principle of data locality implies that cache access is usually within the same cache block, this division scheme makes sure each adjacent 64-bit word maps to a different bank and is thereby simultaneously accessible [de 03].

Virtual to Physical address translation is performed by the Translate Look aside Buffers. L1 caches have their own TLB, which is fully associative and contains 32 4kbyte and 8 2Mbyte-4Mbyte page translations [KMAC03].

4.3 L2-Cache

The Level 2 on-chip cache is 1 Mbyte large. What is important is that all data in the L2 cache is **mutually exclusive** relative to the L1 cache [KMAC03]. The L2 cache is 16-way set-associative. It uses a pseudo-Least Recently Used scheme to decide which cache line to evict first.

The Level 2 TLB is 4-way set associative and contains 512 4kbyte page translations. A remarkable feature of the Opteron is its hardware flush filter. Normally, when there is a change to the page directory base, the TLB is cleared. Opteron uses the flush filter to maximally reuse TLB entries [KMAC03]. This keeps as much address translations in the TLB as possible and is advantageous in multiprocessing schemes.

4.4 Memory Segmentation

The legacy x86 architecture featured memory segmentation support so that programs could be isolated from each other. However, most operating systems handle segmentation of programs entirely in software. AMD realized this and therefore has dispensed the x86 segmentation model. This leaves segmentation entirely to the operating system and gives the processor more space for optimization [Tec03].

Memory segmentation is treated differently according the Opteron's opmode. In 64-bit mode, Opteron uses a flat virtual memory model. This means that the virtual memory space is treated as a single unsegmented address space [Tec03]. The operating system can still specify offsets for code, stack and data segments, but the base segment offset is always 0.

In compatibility and legacy mode, memory segmentation is still used. The address space is treated as a set of segments, each with their own base address [Tec03].

5 Hypertransport

5.1 What is Hypertransport?

Hypertransport is a new I/O technology developed by AMD itself, but now managed by the Hypertransport Consortium [Con03]. It is a high-performance alternative to current-day system bus technologies. Hypertransport uses dual,

unidirectional point-to-point links [Dev01b] to connect components with one another, achieving high bandwith and low latencies.

A hypertransport link's bandwidth is determined by its signaling speed and its width. A link can be 2 to 32 bits wide and can operate at frequencies from 400Mhz up to 1.6Ghz. Data is transmitted using a packet-based protocol, where packets are multiples of four bytes. Packet sizes range from 4 up to 64 bytes [Dev01b].

Hypertransport links connect three generic types of devices. A *cave*, which is a single-link device at the end of a HT chain, a *tunnel*, which is a dual-link device and a *bridge*, which can be used to connect Hypertransport links to other types of devices. A major advantage of Hypertransport is its compatibility with other technologies like PCI, which are important for introducing it in the industry [Neu02].

Hypertransport can work in two modes: *coherent* and *non-coherent* mode. Coherent mode is used for interprocessor connections. In coherent mode, delay is minimized and cache-coherency information is communicated between multiple processors. Non-coherent mode is optimized for maximum performance and is used for I/O communication.

5.2 Hypertransport in the Opteron Architecture

HyperTransport is used in the Opteron to link the CPU (the integrated memory controller) to the system memory, to connect north- and southbridge chipsets and to connect several Opteron CPUs (using coherent HT) to form a multiprocessor Opteron configuration.

More specifically, each Opteron is equipped with three Hypertransport links. The type of link differs depending on the model. The 100 series are equipped with three non-coherent links. Coherency is not required here since it is a uniprocessor. The 200 series are equipped with two non-coherent links and one coherent for multiprocessor support. Finally, the 800 series are equipped with three coherent links [Sta03].

Opteron's Hypertransport links are 16-bit wide, bidirectional, with frequencies ranging from 200Mhz up to 800Mhz. This results in a maximum bandwidth of up to 6.4Gbytes/sec (up to 3.2GBytes/s in each direction). Since the Opteron has three of these, it can provide a total bandwidth of 19.2Gbytes/sec to the rest of the system [Sta03].

5.3 Chipset Support

AMD has developed its own chipset, the 8000 series, for Opteron processors. This chipset incorporates standard northbridge and southbridge functionality, except for, of course, the part already handled by the integrated memory controller. These chips are all connected with the Opteron using Hypertransport. The three chips are:

AMD8151 Hypertransport AGP Tunnel. An AGP 3.0 graphics controller. This chip can be thought of as the remains of the standard northbridge.

AMD8131 Hypertransport PCI-X Tunnel. Provides two PCI-X bus bridges.

AMD8111 Hypertransport I/O Hub. Provides standard southbridge functionality, including PCI control, BIOS, USB, hard disk, network, audio, ...

The strategy of developing three different chips is important from an economic point of view. Not everybody needs all features of the three chipsets. Ordinary desktop models do not require a PCI-X bus for example, while servers usually do not require extensive AGP features. Also, making the chips standalone has the advantage that one can be used multiple times. Ie. one can imagine a server with multiple PCI-X bridges (AMD8131 chips).

6 Opteron Multiprocessor Systems

6.1 Multiprocessor Configurations

The Opteron was designed to be a multiprocessor. More specifically, it needed to be a scalable multiprocessor: one where price and performance scale reasonable with the number of processors used. The Opteron is designed to work in systems from 1 up to 8 processors.

Why is the Opteron such a good scalable multiprocessor? In the past, AMD has already architected multiprocessors (eg. Athlon MP), where they have studied the bottlenecks of the system. The major bottleneck was the memory, which was shared between all processors. This also meant that processors had to share the front-side bus in connecting to the northbridge. The more processors in the system, the less FSB bandwidth each individual CPU gets [Lal03].

AMD's solution with the Athlon MP was to provide each CPU with its own connection to the northbridge, but this was not economically scalable. The Opteron avoids the same problems using the integrated memory controller. As mentioned before, adding a CPU means adding a memory controller, so each Opteron has its own, private, 128-bit connection with its own memory, with a peak bandwidth of 5.3Gbytes/sec.

Thanks to Hypertransport, which is highly scalable, the CPUs can access their neighbour's memory at a maximum transfer rate of 3.2GBytes/second. The CPU can thus both use its local controller and a non-local one to acquire data. The net effect is that creating a 2-way multiprocessor system is as "easy" to implement as an 8-way system, since the system itself is built up of scalable parts [Lal03]. AMD likes to call this "glueless multiprocessing" [Web01], since processors can be *gluelessly* interconnected with coherent HT links [KMAC03].

In order not to kill performance in a multiprocessor system, the Operating System must be smart enough to distribute the data across the memory in a balanced fashion. If, for example, we have a 4-way Opteron server working on a dataset, this set should not entirely be stored in the memory of one

CPU. Instead, it should be distributed by either copying or dividing it over all processor's memory [Lal03].

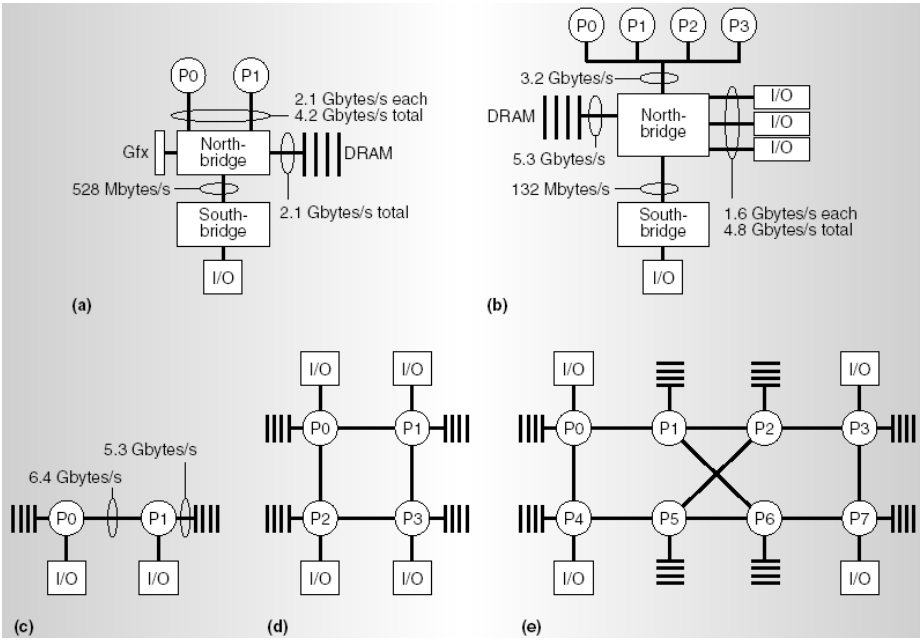

Figure 6: Multiprocessor architectures. **a** two-way Athlon. **b** four-way Xeon. **c,d,e** two-way, four-way and eight-way Opteron. [KMAC03]

Figure 6 gives an overview of the different approaches in the Athlon, Intel Xeon and Opteron multiprocessors. Note how both Athlons (**a**) are sharing the same memory controller. The Xeon processors (**b**) even have to share the front-side bus. Finally, the Opteron scales from 2 up to 8 processors, always using the same components: the integrated memory controller and hypertransport links.

It is possible to scale the Opteron beyond 8 processors, by using Hypertransport switches connecting clusters of 8 multiprocessor Opteron systems. These can then for example be connected in a hypercube topology [KMAC03].

6.2 Distributed Memory Access

When working with a multiprocessor system where each processor is assigned its own memory, there is an interesing difference in how the programmer perceives this address space.

One model is called NUMA (*Non-Uniform Memory Access*), where memory access time depends on what memory is accessed. NUMA is a scalable model, but its software management is difficult [Dev03b] since the programmer has to be careful where to store the data to ensure good performance. Another model is called SMP (*Symmetric Multi Processing*) where each memory has the same

access time. Adding processors is more difficult in this model, but the software view is much simpler since it doesn't matter where we store our data.

What class does the Opteron belong to? On one hand, the Opteron is clearly a NUMA architecture because access times will vary depending on the fact whether memory access is local to the processor or not. On the other hand, AMD considers its architecture SMP from the programmer's point of view, because the difference in access times between local and remote memory is not very large [Kar03].

Concretely, memory access times for Opterons are less than 50ns for uniprocessors, 70ns for 2-way and 110ns for 4-way Opteron configurations. The difference in latency between a local and a remote memory access is comparable to the difference between a DRAM page hit and a DRAM page conflict [KMAC03]. [Kar03] reports the following access times for a 2Ghz DDR333 Opteron in a four-way MP system: local acces time takes **100ns**, remote access to the neighbour's memory takes **118ns** and remote access to the opposite neighbour (requiring 2 hops) takes **136ns**.

AMD calls Opteron's model SUMO (*Sufficiently Uniform Memory Organization*). It is a combination of the previous two. The programmer perceives the address space as flat, while additional processors can easily be added with their own memory. Thanks to Hypertransport (with its bandwidth of 3.2Gbyte/sec), processors can so quickly access a neighbour's memory that the difference in access time becomes negligible, thus, the programmer doesn't really have to worry about exact memory locations and can program code for the Opteron as if it were an SMP model.

Cache coherency is solved by the use of the MOESI protocol. MOESI stands for the states in which a cache line can be. These states are *Modified, Owner, Exclusive, Shared* and *Invalid*. Upon reading from and writing to the memory, processors will snoop the caches of other processors, thereby taking the necessary actions according to the state.

7 Conclusion

The Opteron is an implementation of the 8^{th} generation Hammer architecture, aimed to provide high-level server performance. Moreover, the Opteron was designed to become a scalable multiprocessor. Such features as an integrated memory controller and Hypertransport links have ensured this scalability. Combined with the shift to 64-bit computing and its efficient architecture, Opteron proves to be an ideal server processor, useful for running a wide range of applications.

References

[AMD03] AMDBoard.com. Amd opteron special, 2003. http://www.amdboard.com/opteron_special.html.

[Con03] Hypertransport Consortium. Hypertransport consortium website, 2003. http://www.hypertransport.org.

[De 02] Johan De Gelas. Opteron and itanium: Two roads to 64-bit computing. July 2002. http://www.aceshardware.com/read.jsp?id=50000240.

[de 03] Hans de Vries. Understanding the detailed architecture of amd's 64 bit core. September 2003. http://www.chip-architect.com/news/2003_09_21_Detailed_Architecture_of_%AMDs_64bit_Core.html.

[Dev01a] Advanced Micro Devices. Amd eighth-generation processor architecture. October 2001. White Paper.

[Dev01b] Advanced Micro Devices. Hypertransport technology i/o link. July 2001. White Paper.

[Dev03a] Advanced Micro Devices. Amd official website, 2003. http://www.amd.com.

[Dev03b] Advanced Micro Devices. Amd opteron processor for mp server systems. April 2003.

[Dev03c] Advanced Micro Devices. Software optimization guide for amd athlon64 and amd opteron processors. September 2003.

[Fre03] Vince Freeman. Amd's opteron: The road ahead. April 2003. http://www.cpuplanet.com/features/article.php/2195741.

[Kar03] Viktor Kartunov. Facts and assumptions about the architecture of amd opteron and athlon 64, 2003.

[Kel02] Chetana Keltcher. The amd hammer processor core. August 2002. Presentation at Hotchips 14 Symposium.

[KMAC03] Chetana Keltcher, Kevin McGrath, Ardsher Ahmed, and Pat Conway. The amd opteron processor for multiprocessor servers. *IEEE Computer Society*, pages 66–76, March-April 2003.

[Lab02] X-Bit Labs. A glance at the future: Amd hammer processors and x86-64 technology. August 2002. http://www.xbitlabs.com/articles/cpu/display/hammer-preview.html.

[Lal03] Anand Lal Shimpi. Amd opteron coverage - part 1: Intro to opteron/k8 architecture. April 2003. http://www.anandtech.com/cpu/showdoc.html?i=1815.

[Les03] Darren Lester. 64-bit architectures, December 2003. http://www.just2good.co.uk/index.htm?ITFrameSet.htm?cpu64bit.htm.

[Neu02] Chris Neuts. Introduction to hypertransport technology. 2002. Presentation at AMD Developer Symposium 2002.

[RP02] Omid Rahmat and Thomas Pabst. Amd's opteron comes down hard. April 2002. http://www17.tomshardware.com/cpu/20020424/index.html.

[Sta03] Nick Stam. Inside amd's opteron processor. April 2003. http://www.extremetech.com/article2/0,3973,1036433,00.asp.

[Tec03] AMD64 Technology. Amd64 architecture programmer's manual volume 1: Application programming. September 2003.

[Web01] Fred Weber. Amd's next generation microprocessor architecture. October 2001.