Dual Core Processors – A brief overview[1]

Anirban Sinha

anirbans@cs.ubc.ca

Abstract:

There has been an ever increasing demand for higher and higher processing speeds. However, in this growing competition of making processors faster and faster, CPU designers have nearly exhausted their collective bag of tricks to get more performance out of additional transistors on a chip by increasing parallelism at the instruction level. Speculative execution and deep pipelining are by now very standard features, and CPU designs are getting increasingly complex and hard to manage. This problem is further exacerbated by the fact that chips are sucking up large amounts of power and expending much of it as heat, and the problem grows more acute as clock speeds ramp up. However, both AMD and Intel have found solution to this problem by redesigning the processor architecture differently. By dialing back clock speeds and putting multiple CPU cores on a chip, the theory goes, processor performance can rise as transistor counts do. This paper takes a brief peek at the dual core processor technology and what chip designers plans to do.

Keywords:

Processor, dual core processors, AMD, Intel, CPU, architecture, instruction cycle.

Introduction:

In the October 1989 issue of IEEE Spectrum, an article titled “Microprocessors Circa 2000” predicted that multi-core processors could come to market soon after the turn of the century. The paper was the work of four Intel Corporation technologists, including Pat Gelsinger, vice president and general manager of the Digital Enterprise Group, who envisioned the future through the lens of Moore’s Law. Fifteen years later, their predictions are proving to be true and multi-core processor capability development has become one of the top business and product initiatives for Intel. In April of 2005, Intel announced the Intel® Pentium® processor Extreme Edition, featuring an Intel dual-core processor, which can provide immediate advantages for people looking to buy systems that boost multitasking computing power and improve the throughput of multi threaded applications. AMD has not been far from this. AMD released its dual-core Opteron server/workstation processors on 22 April 2005, and its dual-core desktop processors, the Athlon 64 X2 family, were released on 31 May 2005. But why is this sudden shift in design? Is this solely for commercial and marketing purposes? No.

The problem with winding up clock speeds is heat. At present the processor engine can operate at only so much RPM before the engine will seize. Heat is the enemy of any processor and high clock speeds

[1] First Uploaded: April 12th 2006. Last Modified : April 12th 2006. Credits: 5

mean high heat and that means errors. A Windows PC running at 10GHz isn't much good if it can't make it past booting up before crashing.

That heat comes from power. It takes a lot of juice to crank up a processor to high clock speeds and a processor with that much electricity running around the die is prone to noise. It's not audible noise like a high RPM cooling fan but electrical noise otherwise akin to interference. The pathways on a processor are microscopically close together. The more power that runs through these pathways due to the requirement of higher clock speeds means that there will be a small amount of electrical radiation from one pathway to the next. That leakage could corrupt the data in another pathway. Corrupted data means errors which means a program could get cranky. Thus, multi-core processing appears to be the obvious answer to solve these problems.

Hyperthreading and Dual-Core Processing:

To start from beginning, a thread is simply a single stream of data through the processor on the system. Each application generates its own or multiple threads depending upon how it is running. With current multitasking, a processor can only handle a single thread at a time, so the system rapidly switches between the threads to process the data in a seemingly concurrent manner.

The benefit of having multiple processors is that the system can handle more than one thread. Each processor can handle a separate stream of data. This greatly increases the performance of a system that is running concurrent applications such as a server.

Prior to designing the actual dual core processors, Intel implemented something they called Hyperthreading. This is not the same as multithreading. Instead it is a technology embedded within a single core processor to make it appear to the system as if it had multiple processors. With HT Technology, two threads can execute on the same single processor core simultaneously in parallel rather than context switching between the threads. Scheduling two threads on the same physical processor core allows better use of the processors resources. HT Technology adds circuitry and functionality into a traditional processor to enable one physical processor to appear as two separate processors. Each processor is then referred to as a logical processor. The added circuitry enables the processor to maintain two separate architectural states and separate Advanced Programmable Interrupt Controllers (APIC) which provides multi-processor interrupt management and incorporates both static and dynamic symmetric interrupt distribution across all processors. The shared resources include items such as cache, registers, and execution units to execute two separate programs or two threads simultaneously. Requirements to enable HT Technology are system equipped with a processor with HT Technology, an OS that supports HT Technology and BIOS support to enable/disable HT Technology.

According to Intel, "*Hyper-Threading Technology (HT Technology) provides thread-level parallelism on each processor, resulting in more efficient use of processor resources, higher processing throughput, and improved performance on today's multi-threaded software. The combination of an*

Intel® processor and chipset that support HT Technology, an operating system that includes optimizations for HT Technology, and a BIOS that supports HT Technology and has it enabled, delivers unmatched system performance and responsiveness".

What this really did was speed up the rate at which the system could switch between multiple threads thus boosting multitasking on personal computers.

The goal of a dual-core CPU however is to take two physical processors and integrate them on one physical chip. The idea behind this implementation of the chip's internal architecture is in essence a "divide and conquer" strategy. In other words, by dividing up the computational work performed by the single Pentium microprocessor core in traditional microprocessors and spreading it over multiple execution cores, a multi-core processor can perform more work within a given clock cycle. Thus, it is designed to deliver a better overall user experience. A processor equipped with thread-level parallelism can execute completely separate threads of code. This can mean one thread running from an application and a second thread running from an operating system, or parallel threads running from within a single application.

It is also possible to have a dual processor system that contains two HT Technology enabled processors which would provide the ability to run up to 4 programs or threads simultaneously. This capability is currently available on Intel Xeon processors.

The Caveat:

While all these looks very promising and appealing, there is however a major caveat. n order for the true benefits of the multiple processors to be seen, the software that is running on the computer must be written to support multithreading. Without the software supporting such a feature, threads will be primarily run through a single processor thus degrading the efficiency. Though almost all operating systems today support multicore and multiprocessors, most user level applications are not written to take advantage of the existing multiprocessors. As a result, any speed benefits will solely be from the operating system being able to separate applications between the processors. Thus application level parallelism is required to take the full advantage of a multi core system. For example, most gaming systems today consists of a graphics rendering engine and a AI engine that predicts the behavior of the gaming application depending on user inputs. These two essentially are parallel in nature and can be easily executed on different processors. However, with a single processor, both of these must function by switching between the two. This is not necessarily efficient. Thus a gaming application can benefit from having a dual core processor immensely.

Thus, like most applications, gaming is not designed to take advantage of the multiple processors. As a result, both the rendering and AI happen on a single processor leaving the second processor essentially unused. As a result, a multiple core PC will not have any speed benefit. For the average user the difference in performance will be most noticeable in multi-tasking until more software is SMT aware.

If the game is designed with multiple threads, then a dual-core processor would be advantageous over a single processor. Adobe Photoshop is an example of SMT (simultaneous multi-threading technology) aware software. SMT is also used with multi-processor systems common to servers.

The Architecture:

Conceptually, a dual core processor architecture can be described as shown in the figure 1. Integrated circuit (IC) chips contain two complete physical computer processors (cores) in the same IC package. Typically, this means that two identical processors are manufactured so they reside side-by-side on the same die. It is also possible to (vertically) stack two separate processor die and place them in the same IC package. Each of the physical processor cores has its own resources (architectural state, registers, execution units, etc.). The multiple cores on-die may or may not share several layers of the on-die cache.

A dual core processor design could provide for each physical processor to: 1) have its own on-die cache, or 2) it could provide for the on-die cache to be shared by the two processors, or 3) each processor could have a portion of on-die cache that is exclusive to a single processor and then have a portion of on-die cache that is shared between the two dual core processors. The two processors in a dual core package could have an on-die communication path between the processors so that putting snoops and requests out on the FSB is not necessary. Both processors must have a communication path to the computer system front-side bus.

In terms of architecture, AMD and Intel have quite different ways of dealing with this issue of multicore systems. Figure 2 shows very simplified diagram of a dual-core Opteron designed by AMD. Each of the K8 cores has its own, independent L2 cache onboard, but the two cores share a common system request queue. They also share a dual-channel DDR memory controller and a set of HyperTransport links to the outside world. Access to these I/O resources is adjudicated via a crossbar, or switch, so that each CPU can talk directly to memory or I/O as efficiently as possible. In some respects, the dual-core Opteron acts very much like a sort of SMP system on a chip, passing data back and forth between the two cores internally. To the rest of the system I/O infrastructure, though, the dual-core Opteron looks more or less like the single-core version.

The Opteron's system architecture remains very different from that of its primary competitor, Intel's Xeon. AMD says its so-called Direct Connect architecture was over-designed for single-core Opterons with an eye to the dual-core future. Each processor (whether dual core or single) has its own local dual-channel DDR memory controller, and the processors talk to one another and to I/O chips via point-to-point HyperTransport links running at 1GHz. This arrangement makes for a network-like system topology with gobs of bandwidth. The total possible bandwidth flowing through the 940 pins of an Opteron 875 is 30.4GB/s. With one less HyperTransport link, the Opteron 275 can theoretically hit 22.4GB/s. MD uses a cache coherency protocol called MOESI, A CPU that "owns" certain data has that data in its cache, has modified it, and yet makes it available to other CPUs. Data flagged as Owner in an Opteron cache can be delivered directly from the cache of CPU 0 into the cache of CPU 1 via a CPU-to-CPU HyperTransport link, without having to be written to main memory. This interface runs at the speed of the CPU, so transfers from the cache on core 0 into the cache on core 1 should happen very, very quickly.

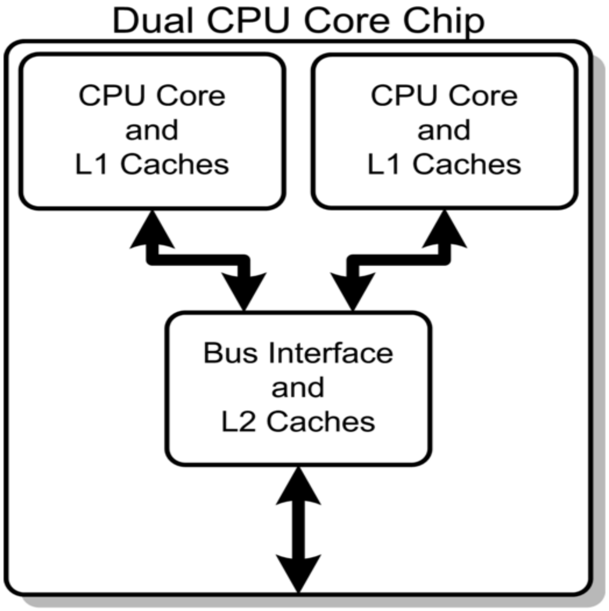

Figure 1 : Generic Diagram of a dual core processor architecture. Source: Wikipedia.

This is in stark contrast to the Intel design where MESI updates are communicated over the front-side bus. There is no alternative internal on-chip data path. Current Xeons have a shared front-side bus on which the north bridge chip (with memory controller) and both processors reside. At 800MHz, its total bandwidth is 6.4GB/s—a possible bottleneck in certain situations.

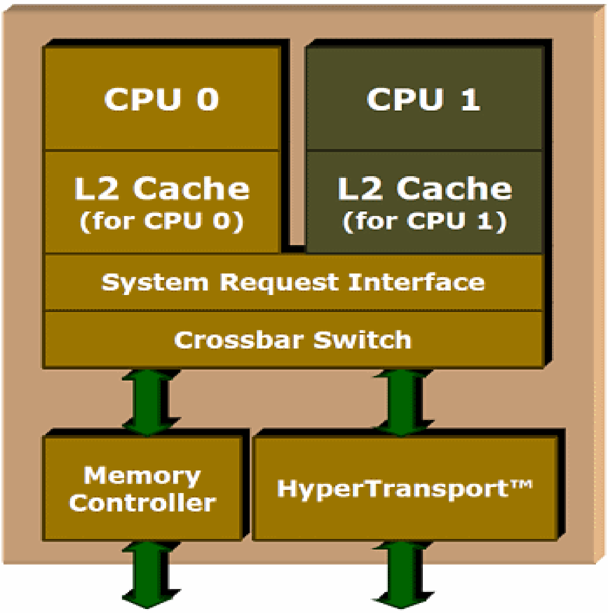

Figure 2: AMD Dual core processor architecture. Source: AMD.

The Conclusion:

A dual core processor is between a single core processor and a dual processor system for architecture. A dual core processor has two cores but will share some of the other hardware like the memory controller and bus. A dual processor system has completely separate hardware and shares nothing with the other processor. A dual core processor is different from a multi-processor system. In the latter there are two separate CPUs with their own resources. In the former, resources are shared and the cores reside on the same chip. A multi-processor system is faster than a system with a dual core processor, while a dual core system is faster than a single-core system, all else being equal.

Most of the early dual-core processors runs at lower clock speeds compared to single core processors. The rational behind it is that a dual-core processor with each running at 1GHz should be equivalent to a single processor running at 2GHz.

However, the requirements for successfully delivering hardware-enhanced threading and multi-core processing capability go beyond critical silicon manufacturing capacity and technology. The promise of a better user experience also depends on software as well. Unless we develop parallel user level applications, it will be difficult to harness the full power of multi core processor technology. For the majority of people, there is not going to be much of a benefit for the dual-core over a single core processor. This will gradually change as the dual-core model becomes more common, but it will likely take some time.

References:

Parts of this writeup has been shamelessly stolen from some of the following sources:

1. Dual-Core Processors: Are Two Really Better Than One? - http://compreviews.about.com/od/cpus/a/dualcore.htm
2. AMD's dual-core Opteron processors; Because four is better than two: http://www.techreport.com/reviews/2005q2/opteron-x75/index.x?pg=1
3. Intel® Multi-Core Processor Architecture Development Backgrounder: http://cache-www.intel.com/cd/00/00/20/57/205707_205707.pdf
4. Dual Core Processing: Over-simplified, demystified and explained: http://www.short-media.com/review.php?r=261
5. Dual core shoot-out: Intel versus AMD: http://reviews.zdnet.co.uk/hardware/processorsmemory/0,39024015,39233885,00.htm
6. Understanding Dual Processors, Hyper-Threading Technology, and Multi Core Systems: http://www.devx.com/Intel/Article/27399
7. Wikipedia Resources: www.wikipedia.org
8. What is a Dual Core Processor?: http://www.wisegeek.com/what-is-a-dual-core-processor.htm
9. Intel Dual Core Processors: http://www.intel.com/technology/computing/dual-core/
10. Intel Hyper Threading Technology: http://www.intel.com/technology/hyperthread/