

Understanding Dual-processors, Hyper-Threading Technology, and Multicore Systems

This paper will provide you with a basic understanding of the differences among several computer system architectures—dual-processor systems, Intel® Xeon® processors with Hyper-Threading Technology, dual-core processor systems, and multicore processor systems—capable of executing two or more programs, for example in a multitasking environment or multiple threads within one program.

**by Tom Martinez,
Sunish Parikh**
February 28, 2005

This paper is intended to provide software professionals with a basic understanding of the differences between several computer system architectures capable of executing two (or more) software programs, for example in a multitasking environment or multiple threads within one program. The architectures that will be discussed are: dual-processor systems, Intel® Xeon® processors with Hyper-Threading Technology, dual-core processor systems, and multicore processor systems.

Definitions and Architectural Details

The Dual-processor

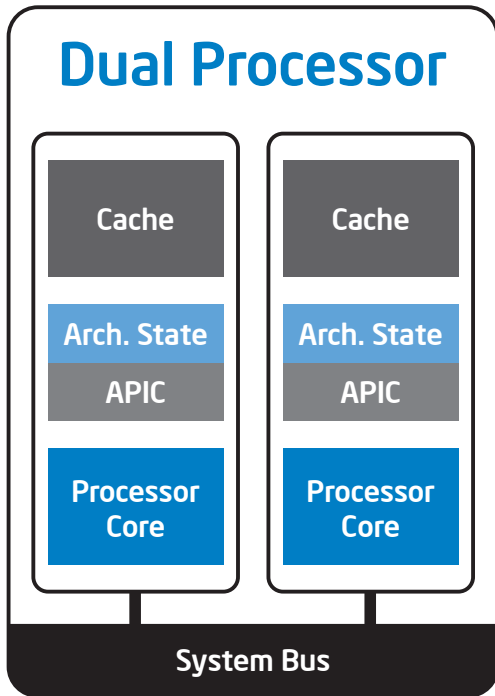


Figure 1. Dual-processor system.

A traditional dual-processor system contains two separate physical computer processors in the same chassis. The two processors are usually located on the same circuit board (motherboard) but occasionally will be located on separate circuit boards. In this case, each of the processors will reside in its own socket. A dual-processor (DP) system can also be considered a subset of the larger set of a symmetric multiprocessor (SMP) system. A multiprocessor capable operating system can schedule two separate computer processes or two threads within a process to run simultaneously on these separate processors.

Hyper-Threading Technology

Hyper-Threading Technology (HT Technology) was developed by Intel Corporation to bring the simultaneous multithreading approach to the Intel architecture. 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 is available on Intel Xeon processors and some Intel® Pentium® 4 processors. 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 operating system that supports HT Technology and BIOS support to enable/disable HT Technology.

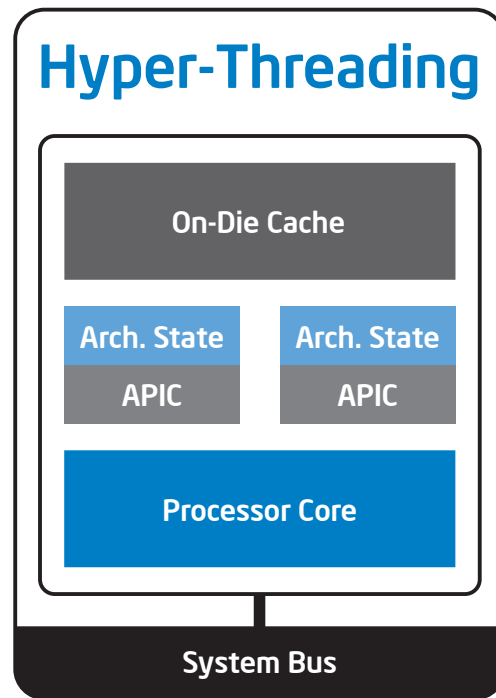


Figure 2. Processor equipped with Hyper-Threading Technology.

You can find some additional, more complete, and technical descriptions of HT Technology in the Intel Technology Journal.

Note that it is also possible to have a dual-processor system that contains two HT Technology-enabled processors that would provide the ability to run up to four programs or threads simultaneously. This capability is currently available on Intel Xeon processors, and these systems are currently available from several OEMs making and selling Intel Xeon processor-based DP systems.

Dual-core

This term refers to integrated circuit (IC) chips that 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, and so on). 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.

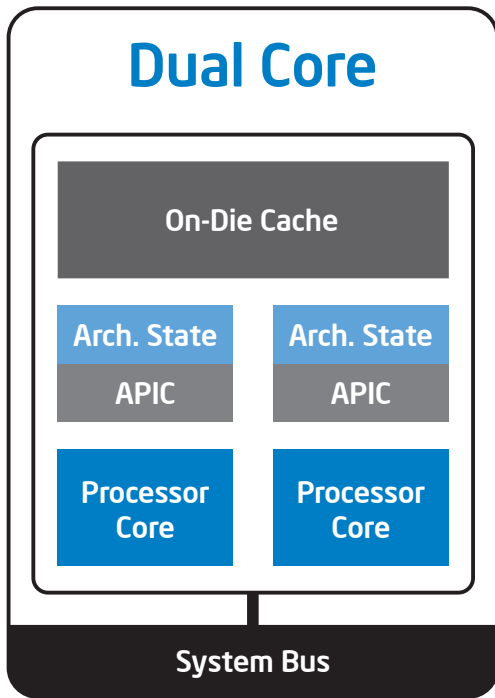


Figure 3. Dual-core processor system.

Note that dual-core processors could also contain HT Technology that would enable a single processor IC package, containing two physical processors, to appear as four logical processors capable of running four programs or threads simultaneously. At the Intel Developer Forum in Fall 2004, Intel publicly announced that it planned to release dual-core processors for mobile, desktop, and server platforms in 2005.

Multicore

The multicore system is an extension to the dual-core system except that it would consist of more than 2 processors. The current trends in processor technology indicate that the number of processor cores in one IC chip will continue to increase. If we assume that the number of transistors per processor core remains relatively fixed, it is reasonable to assume that the number of processor cores could follow Moore's Law, which states that the number of transistors per a certain area on the chip will double approximately every 18 months. Even if this trend does not follow Moore's Law, the number of processor cores per chip appears destined to steadily increase—based on statements from several processor manufacturers. The optimal number of processors is yet to be determined but will probably change over time as software adapts to effectively use many processors, simultaneously. However, a software program that is only capable of running on one processor (or very few processors) will be unable to take full advantage of future processors that contain many processors cores. For example, an application running on a 4-processor system with each socket containing quad-core processors has 16 processor cores available to schedule 16 program threads simultaneously. At the Intel Developer Forum Intel also publicly announced its intention to develop and manufacture multicore processors in its Itanium Processor Family.

Operating System Support

Software Licensing Models

In order for multiprocessor systems to be effective the operating system must be able to detect multiple processors and provide a mechanism to schedule separate processes or threads on the physical and logical processors present. This is true for all the architectural systems discussed in this paper. Microsoft provides several versions of Windows* that have this capability, as do many Unix/Linux*-based operating systems.

Software vendors use a wide variety of software licensing models. Some offer a single flat fee without regard to the number of processors present on the computer system running the software. Some software vendors charge a fee based on the number of processors reported by the operating system. With the introduction of HT Technology from Intel, many software vendors changed their licensing scheme to charge based on the number of physical processors. In this case, it is best to have an operating system that is HT Technology aware so the operating system will report physical processors as opposed to or in addition to logical processors. There are also many software vendors who have agreed to charge license fees based on physical processor as opposed to the number of logical processor reported by the operating system. Furthermore, with the introduction of dual-core and multicore processors, a number of software vendors have committed to license based on the number of processor sockets, instead of based on the total number of processor cores.

Price/Performance

There are many factors that affect the cost of processors and computer systems and therefore affect the price that a computer system vendor will charge for a system.

Traditional dual-processor systems require two separate processors in two sockets. Both may be physically located on one main computer board or they may each have their own separate board. Using separate boards may make systems more modular (and possibly easier to maintain). Dual-processor systems provide excellent performance because both processors can operate independently since each has their own computing resources.

A simple computer system containing a single processor with HT Technology appears to an operating system as having two logical processors. Since this is a single processor system, it would likely cost much less than a dual-processor system since the cost of each processor is a significant portion of the overall system. The HT Technology capable processor is actually a single processor that appears as two logical processors. It is not expected that it would perform as two distinct processors. For example, a HT Technology-enabled processor might perform about 1.15x to 1.3x the performance of an equivalent processor with HT Technology disabled. As mentioned before, a dual-processor system could have two HT Technology capable/enabled

processors. The price of this system would be expected to be similar to that of a traditional dual-processor system. The performance of this system would be expected to be equal to, or greater than, a traditional dual-processor system. A dual-core processor would be expected to take up more die area and consequently cost more than a single processor. However, the cost of a dual-core processor, in a single IC package, would be expected to be less than the cost of two separate processors in separate IC packages.

The performance difference of dual-core and dual-processor designs depends on the underlying implementation. If the same amount/speed interconnects exist on a dual-core processor as exist on a dual-processor system, the improvement is bound by the effective utilization of multithreading in its architecture, whereas in dual-processor designs there needs to be an efficient way to route data to two different CPUs and then reroute the data into a result. With current dual-processor arrangements, single-threaded apps are expected to see a 10% increase in performance whereas multi threaded apps are expected to perform better by at least 40-80% over a uni-processor configuration for a majority of applications. As new dual-core architecture become available, efficiency will continue to climb to the point where a dual-core will be as efficient as dual CPU. With dual-core, the two chips might communicate at a much faster rate and share resources. At the same time, power consumption goes down because the electrical pathway between chips has been dramatically reduced. Dual-core systems are capable of taking advantage of parallelism for improving performance rather than trying to increase performance of a single thread using frequency bumps and processor complexity. Power consumption increases exponentially with processor speed so a small drop in processor speed allows significant savings in power consumption. HT Technology, on the other end, saves energy by increasing performance by 18%-30% in a multi-tasking environment depending on the application. The circuitry for HT Technology takes up less than 5% of the chip space thereby giving a substantial increase in performance for a very small investment in die size.

Software Development Considerations

Assuming the operating system is appropriate for the hardware system, software that runs on dual-processor systems should run on HT Technology capable/enabled processors and on dual-core processor systems without modification. Even if the software is not multi-threaded, it can still take advantage of multiple physical and/or logical processors in a multitasking environment. For example, a software developer could answer e-mail or research a technical problem on the internet while a large software application is being compiled in the background. Although all applications should run on multi-processor systems, multithreaded applications should benefit the most from the multi-processor systems discussed above. In order to get the best performance, it may be necessary to tune or optimize the application to take advantage of a specific architecture or multiprocessor implementation.

Advances in computer architecture technology are enabling hardware vendors to put multiple cores on a single socket. The challenge is to architect software applications to fully exploit the capabilities of the compute power of a multicore system. In order to take advantage of the trend toward dual-core and multicore processors, software developer should multithread their applications and could use, for example, processor detection APIs to determine how many threads to launch in an application to maximize the use of all available physical and logical processors.

Tom Martinez is a software performance engineer for the Core Software Division of the Software Solutions Group. He joined Intel in 2000. He has a BS in computer engineering from the University of New Mexico and an MBA from the University of Southern California. Prior to joining Intel, he worked for the US Department of Defense and also worked for major defense contractors. Tom supports software vendors to leverage the performance and features on Intel Xeon and Itanium processors designed for servers.

Sunish Parikh is a software performance engineer for the Core Software Division of the Software Solutions Group. He joined Intel in 2000 after graduating with a masters' degree in computer engineering from Florida Atlantic University. Sunish works with major software vendors developing enterprise server applications, enabling their applications on upcoming Intel architectures and optimizing the software to take advantages of the hardware platform.

