Homayoun

Reference 11

WIKIPEDIA

Multi-core processor

A **multi-core processor** is a computer processor integrated circuit with two or more separate processing units, called cores, which each read and execute program instructions, as if the computer had several processors.[1] The instructions are ordinary CPU instructions (such as add, move data, and branch) but the single processor can run instructions on separate cores at the same time, increasing overall speed for programs that support multithreading or other parallel computing techniques.[2] Manufacturers typically integrate the cores onto a single integrated circuit die (known as a chip multiprocessor or CMP) or onto multiple dies in a single chip package. The microprocessors currently used in almost all personal computers are multi-core. A multi-core processor implements multiprocessing in a single physical package. Designers may couple cores in a multi-core device tightly or loosely. For example, cores may or may not share caches, and they may implement message passing or shared-memory inter-core communication methods. Common network topologies to interconnect cores include bus, ring, two-dimensional mesh, and crossbar. Homogeneous multi-core systems include only identical cores; heterogeneous multi-core systems have cores that are not identical (e.g. big.LITTLE have heterogeneous cores that share the same instruction set, while AMD Accelerated Processing Units have cores that don't even share the same instruction set). Just as with single-processor systems, cores in multi-core systems may implement architectures such as VLIW, superscalar, vector, or multithreading.

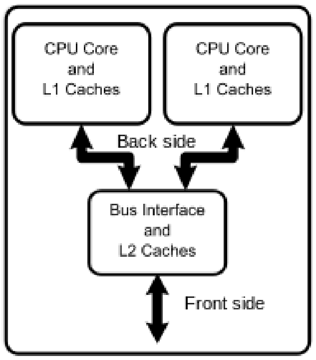

Diagram of a generic dual-core processor with CPU-local level-1 caches and a shared, on-die level-2 cache.


An Intel Core 2 Duo E6750 dual-core processor.


An AMD Athlon X2 6400+ dual-core processor.

Multi-core processors are widely used across many application domains, including general-purpose, embedded, network, digital signal processing (DSP), and graphics (GPU).

The improvement in performance gained by the use of a multi-core processor depends very much on the software algorithms used and their implementation. In particular, possible gains are limited by the fraction of the software that can run in parallel simultaneously on multiple cores; this effect is described by Amdahl's law. In the best case, so-called embarrassingly parallel problems may realize speedup factors near the number of cores, or even more if the problem is split up enough to fit within each core's cache(s), avoiding use of much slower main-system memory. Most applications, however, are not accelerated so much unless programmers invest a prohibitive amount of effort in re-factoring the whole problem.[3] The parallelization of software is a significant ongoing topic of research.

Contents

Terminology

The terms *multi-core* and *dual-core* most commonly refer to some sort of central processing unit (CPU), but are sometimes also applied to digital signal processors (DSP) and system on a chip (SoC). The terms are generally used only to refer to multi-core microprocessors that are manufactured on the *same* integrated circuit die; separate microprocessor dies in the same package are generally referred to by another name, such as *multi-chip module*. This article uses the terms "multi-core" and "dual-core" for CPUs manufactured on the *same* integrated circuit, unless otherwise noted.

In contrast to multi-core systems, the term *multi-CPU* refers to multiple physically separate processing-units (which often contain special circuitry to facilitate communication between each other).

The terms *many-core* and *massively multi-core* are sometimes used to describe multi-core architectures with an especially high number of cores (tens to thousands[4]).[5]

Some systems use many soft microprocessor cores placed on a single FPGA. Each "core" can be considered a "semiconductor intellectual property core" as well as a CPU core.

Development

While manufacturing technology improves, reducing the size of individual gates, physical limits of semiconductor-based microelectronics have become a major design concern. These physical limitations can cause significant heat dissipation and data synchronization problems. Various other methods are used to improve CPU performance. Some *instruction-level parallelism* (ILP) methods such as superscalar pipelining are suitable for many applications, but are inefficient for others that contain difficult-to-predict code. Many applications are better suited to *thread-level parallelism* (TLP) methods, and multiple independent CPUs are commonly used to increase a system's overall TLP. A combination of increased available space (due to refined manufacturing processes) and the demand for increased TLP led to the development of multi-core CPUs.

Commercial incentives

Several business motives drive the development of multi-core architectures. For decades, it was possible to improve performance of a CPU by shrinking the area of the integrated circuit (IC), which reduced the cost per device on the IC. Alternatively, for the same circuit area, more transistors could be used in the design, which increased functionality, especially for complex instruction set computing (CISC) architectures. Clock rates also increased by orders of magnitude in the decades of the late 20th century, from several megahertz in the 1980s to several gigahertz in the early 2000s.

As the rate of clock speed improvements slowed, increased use of parallel computing in the form of multi-core processors has been pursued to improve overall processing performance. Multiple cores were used on the same CPU chip, which could then lead to better sales of CPU chips with two or more cores. For example, Intel has produced a 48-core processor for research in cloud computing; each core has an x86 architecture.[6][7]

Technical factors

Since computer manufacturers have long implemented symmetric multiprocessing (SMP) designs using discrete CPUs, the issues regarding implementing multi-core processor architecture and supporting it with software are well known.

Additionally:

- Using a proven processing-core design without architectural changes reduces design risk significantly.
- For general-purpose processors, much of the motivation for multi-core processors comes from greatly diminished gains in processor performance from increasing the operating frequency. This is due to three primary factors:[8]

  1. The *memory wall*; the increasing gap between processor and memory speeds. This, in effect, pushes for cache sizes to be larger in order to mask the latency of memory. This helps only to the extent that memory bandwidth is not the bottleneck in performance.
  2. The *ILP wall*; the increasing difficulty of finding enough parallelism in a single instruction stream to keep a high-performance single-core processor busy.
  3. The *power wall*; the trend of consuming exponentially increasing power (and thus also generating exponentially increasing heat) with each factorial increase of operating frequency. This increase can be mitigated by "shrinking" the processor by using smaller traces for the same logic. The *power wall* poses manufacturing, system design and deployment problems that have not been justified in the face of the diminished gains in performance due to the *memory wall* and *ILP wall*.

In order to continue delivering regular performance improvements for general-purpose processors, manufacturers such as Intel and AMD have turned to multi-core designs, sacrificing lower manufacturing-costs for higher performance in some applications and systems. Multi-core architectures are being developed, but so are the alternatives. An especially strong contender for established markets is the further integration of peripheral functions into the chip.

Advantages

The proximity of multiple CPU cores on the same die allows the cache coherency circuitry to operate at a much higher clock rate than what is possible if the signals have to travel off-chip. Combining equivalent CPUs on a single die significantly improves the performance of cache snoop (alternative: Bus snooping) operations. Put simply, this means that signals between different CPUs travel shorter distances, and therefore those signals degrade less. These higher-quality signals allow more data to be sent in a given time period, since individual signals can be shorter and do not need to be repeated as often.

Assuming that the die can physically fit into the package, multi-core CPU designs require much less printed circuit board (PCB) space than do multi-chip SMP designs. Also, a dual-core processor uses slightly less power than two coupled single-core processors, principally because of the decreased power required to drive signals external to the chip. Furthermore, the cores share some circuitry, like the L2 cache and the interface to the front-side bus (FSB). In terms of competing technologies for the available silicon die area, multi-core design can make use of proven CPU core library designs and produce a product with lower risk of design error than devising a new wider-core design. Also, adding more cache suffers from diminishing returns.

Multi-core chips also allow higher performance at lower energy. This can be a big factor in mobile devices that operate on batteries. Since each core in a multi-core CPU is generally more energy-efficient, the chip becomes more efficient than having a single large monolithic core. This allows higher performance with less energy. A challenge in this, however, is the additional overhead of writing parallel code.[9]

Disadvantages

Maximizing the usage of the computing resources provided by multi-core processors requires adjustments both to the operating system (OS) support and to existing application software. Also, the ability of multi-core processors to increase application performance depends on the use of multiple threads within applications.

Integration of a multi-core chip can lower the chip production yields. They are also more difficult to manage thermally than lower-density single-core designs. Intel has partially countered this first problem by creating its quad-core designs by combining two dual-core ones on a single die with a unified cache, hence any two working dual-core dies can be used, as opposed to producing four cores on a single die and requiring all four to work to produce a quad-core CPU. From an architectural point of view, ultimately, single CPU designs may make better use of the silicon surface area than multiprocessing cores, so a development commitment to this architecture may carry the risk of obsolescence. Finally, raw processing power is not the only constraint on system performance. Two processing cores sharing the same system bus and memory bandwidth limits the real-world performance advantage. In a 2009 report, Dr Jun Ni showed that if a single core is close to being memory-bandwidth limited, then going to dual-core might give 30% to 70% improvement; if memory bandwidth is not a problem, then a 90% improvement can be expected; however, Amdahl's law makes this claim dubious.[10] It would be possible for an application that used two CPUs to end up running faster on a single-core one if communication between the CPUs was the limiting factor, which would count as more than 100% improvement.

Hardware

Trends

The trend in processor development has been towards an ever-increasing number of cores, as processors with hundreds or even thousands of cores become theoretically possible.[11] In addition, multi-core chips mixed with simultaneous multithreading, memory-on-chip, and special-purpose "heterogeneous" (or asymmetric) cores promise further performance and efficiency gains,[12] especially in processing multimedia, recognition and networking applications. For example, a big.LITTLE core includes a high-performance core (called 'big') and a low-power core (called 'LITTLE'). There is also a trend towards improving energy-efficiency by focusing on performance-per-watt with advanced fine-grain or ultra fine-grain power management and dynamic voltage and frequency scaling (i.e. laptop computers and portable media players).

Chips designed from the outset for a large number of cores (rather than having evolved from single core designs) are sometimes referred to as manycore designs, emphasising qualitative differences.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.