Multi-core processor

From Wikipedia, the free encyclopedia

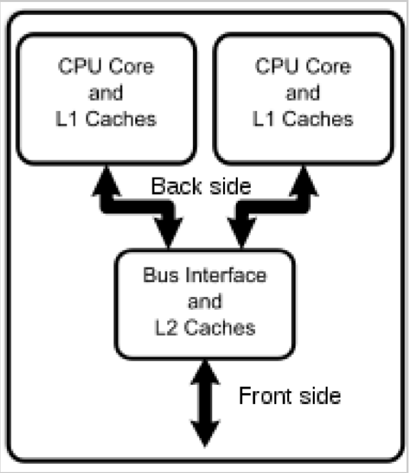

Diagram of a generic dual-core processor, with CPU-local level 1 caches, and a shared, on-die level 2 cache.

A **multi-core processor** is a single computing component with two or more independent actual processors (called "cores"), which are the units that read and execute program instructions.[1] The data in the instruction tells the processor what to do. The instructions are very basic things like reading data from memory or sending data to the user display, but they are processed so rapidly that human perception experiences the results as the smooth operation of a program. 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.

Processors were originally developed with only one core. A **many-core** processor is a multi-core processor in which the number of cores is large enough that traditional multi-processor techniques are no longer efficient[*citation needed*] — largely because of issues with congestion in supplying instructions and data to the many processors. The many-core threshold is roughly in the range of several tens of cores; above this threshold network on chip technology is advantageous.

An Intel Core 2 Duo E6750 dual-core processor.

An AMD Athlon X2 6400+ dual-core processor.

A **dual-core processor** has two cores (e.g. AMD Phenom II X2, Intel Core Duo), a **quad-core processor** contains four cores (e.g. AMD Phenom II X4, the Intel 2010 core line that includes three levels of quad-core processors, see i3, i5, and i7 at Intel Core), and a **hexa-core processor** contains six cores (e.g. AMD Phenom II X6, Intel Core i7 Extreme Edition 980X). 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 which are not identical. Just as with single-processor systems, cores in multi-core systems may implement architectures such as superscalar, VLIW, vector processing, SIMD, or multithreading.

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

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 be parallelized to run on multiple cores simultaneously; 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[2]. 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). Additionally, some use these terms to refer only to multi-core microprocessors that are manufactured on the *same* integrated circuit die. These people generally refer to separate microprocessor dies in the same package 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 or hundreds).

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[citation needed].

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 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 dual-core architectures. For decades, it was possible to improve performance of a CPU by shrinking the area of the integrated circuit, which drove down the cost per device on the IC. Alternatively, for the same circuit area, more transistors could be utilized in the design, which increased functionality, especially for CISC architectures.

As manufacturing techniques reach theoretical limits in miniaturization, increased use of parallel computing in the form of multi-core processors has been perused to improve overall processing performance. Multiple cores were used on the same CPU chip, sale which could then fund further research and development of multiple-core processors. Intel has produced a 48-core processor for research in cloud computing.[3]

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:

- Utilizing 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:
    1. The *memory wall*; the increasing gap between processor and memory speeds. This effect pushes cache sizes 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 instructions stream to keep a high-performance single-core processor busy.
    3. The *power wall*; the trend of consuming exponentially increasing power 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 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.

The largest boost in performance will likely be noticed in improved response-time while running CPU-intensive processes, like antivirus scans, ripping/burning media (requiring file conversion), or file searching. For example, if the automatic virus-scan runs while a movie is being watched, the application running the movie is far less likely to be starved of processor power, as the antivirus program will be assigned to a different processor core than the one running the movie playback.

Assuming that the die can fit into the package, physically, the 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.[*citation needed*]

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 multi-core is generally more energy-efficient, the chip becomes more efficient than having a single large monolithic core. This allows to get higher performance with less energy. The challenge of writing parallel code clearly offsets this benefit.[4]

Disadvantages

Maximizing the utilization 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. The situation is improving: for example the Valve Corporation's Source engine offers multi-core support,[5][6] and Crytek has developed similar technologies for CryEngine 2, which powers their game, *Crysis*. Emergent Game Technologies' Gamebryo engine includes their Floodgate technology[7] which simplifies multicore development across game platforms. In addition, Apple Inc.'s second latest OS, Mac OS X Snow Leopard has a built-in multi-core facility called Grand Central Dispatch for Intel CPUs.

Integration of a multi-core chip drives chip production yields down and they are more difficult to manage thermally than lower-density single-chip designs. Intel has partially countered this first problem by creating its quad-core designs by combining two dual-core 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. 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. If a single core is close to being memory-bandwidth limited, going to dual-core might only give 30% to 70% improvement. If memory bandwidth is not a problem, a 90% improvement can be expected[*citation needed*]. It would be possible for an application that used two CPUs to end up running faster on one dual-core if communication between the CPUs was the limiting factor, which would count as more than 100% improvement.

Hardware

Trends

The general trend in processor development has moved from dual-, tri-, quad-, hexa-, octo-core chips to ones with tens or even hundreds of cores. In addition, multi-core chips mixed with simultaneous multithreading, memory-on-chip, and special-purpose "heterogeneous" cores promise further performance and efficiency gains, especially in processing multimedia, recognition and networking applications. There is also a trend of 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).

Architecture

The composition and balance of the cores in multi-core architecture show great variety. Some architectures use one core design repeated consistently ("homogeneous"), while others use a mixture of different cores, each optimized for a different, "heterogeneous" role.

The article *CPU designers debate multi-core future*[8] by Rick Merritt, EE Times 2008, includes comments:

> *"Chuck Moore [...] suggested computers should be more like cellphones, using a variety of specialty cores to run modular software scheduled by a high-level applications programming interface.*
> *[...] Atsushi Hasegawa, a senior chief engineer at Renesas, generally agreed. He suggested the cellphone's use of many specialty cores working in concert is a good model for future multi-core designs.*
> *[...] Anant Agarwal, founder and chief executive of startup Tilera, took the opposing view. He said multi-core chips need to be homogeneous collections of general-purpose cores to keep the software model simple."*

Software impact

An outdated version of an anti-virus application may create a new thread for a scan process, while its GUI thread waits for commands from the user (e.g. cancel the scan). In such cases, a multicore architecture is of little benefit for the application itself due to the single thread doing all heavy lifting and the inability to balance the work evenly across multiple cores. Programming truly multithreaded code often requires complex co-ordination of threads and can easily introduce subtle and difficult-to-find bugs due to the interleaving of processing on data shared between threads (thread-safety). Consequently, such code is much more difficult to debug than single-threaded code when it breaks. There has been a perceived lack of motivation for writing consumer-level threaded applications because of the relative rarity of consumer-level multiprocessor hardware. Although threaded applications incur little additional performance penalty on single-processor machines, the extra overhead of development has been difficult to justify due to the preponderance of single-processor machines. Also, serial tasks like decoding the entropy encoding algorithms used in video codecs are impossible to parallelize because each result generated is used to help create the next result of the entropy decoding algorithm.

Given the increasing emphasis on multicore chip design, stemming from the grave thermal and power consumption problems posed by any further significant increase in processor clock speeds, the extent to which software can be multithreaded to take advantage of these new chips is likely to be the single greatest constraint on computer performance in the future. If developers are unable to design software to fully exploit the resources provided by multiple cores, then they will ultimately reach an insurmountable performance ceiling.

The telecommunications market had been one of the first that needed a new design of parallel datapath packet processing because there was a very quick adoption of these multiple-core processors for the datapath and the control plane. These MPUs are going to replace[9] the traditional Network Processors that were based on proprietary micro- or pico-code.

Parallel programming techniques can benefit from multiple cores directly. Some existing parallel programming models such as Cilk++, OpenMP, OpenHMPP, FastFlow, Skandium, and MPI can be used on multi-core platforms. Intel introduced a new abstraction for C++ parallelism called TBB. Other research efforts include the Codeplay Sieve System, Cray's Chapel, Sun's Fortress, and IBM's X10.

Multi-core processing has also affected the ability of modern computational software development. Developers programming in newer languages might find that their modern languages do not support multi-core functionality. This then requires the use of numerical libraries to access code written in languages like C and Fortran, which perform math computations faster than newer languages like C#. Intel's MKL and AMD's ACML are written in these native languages and take advantage of multi-core processing.

Managing concurrency acquires a central role in developing parallel applications. The basic steps in designing parallel applications are:

Partitioning

The partitioning stage of a design is intended to expose opportunities for parallel execution. Hence, the focus is on defining a large number of small tasks in order to yield what is termed a fine-grained decomposition of a problem.

Communication

The tasks generated by a partition are intended to execute concurrently but cannot, in general, execute independently. The computation to be performed in one task will typically require data associated with another task. Data must then be transferred between tasks so as to allow computation to proceed. This information flow is specified in the communication phase of a design.

Agglomeration

In the third stage, development moves from the abstract toward the concrete. Developers revisit decisions made in the partitioning and communication phases with a view to obtaining an algorithm that will execute efficiently on some class of parallel computer. In particular, developers consider whether it is useful to combine, or agglomerate, tasks identified by the partitioning phase, so as to provide a smaller number of tasks, each of greater size. They also determine whether it is worthwhile to replicate data and/or computation.

Mapping

In the fourth and final stage of the design of parallel algorithms, the developers specify where each task is to execute. This mapping problem does not arise on uniprocessors or on shared-memory computers that provide automatic task scheduling.

On the other hand, on the server side, multicore processors are ideal because they allow many users to connect to a site simultaneously and have independent threads of execution. This allows for Web servers and application servers that have much better throughput.

Licensing

Typically, proprietary enterprise-server software is licensed "per processor". In the past a CPU was a processor and most computers had only one CPU, so there was no ambiguity.

Now there is the possibility of counting cores as processors and charging a customer for multiple licenses for a multi-core CPU. However, the trend seems to be counting dual-core chips as a single processor: Microsoft, Intel, and AMD support this view. Microsoft have said they would treat a socket as a single processor.[10]

Oracle counts an AMD X2 or Intel dual-core CPU as a single processor but has other numbers for other types, especially for processors with more than two cores. IBM and HP count a multi-chip module as multiple processors. If multi-chip modules count as one processor, CPU makers have an incentive to make large expensive multi-chip modules so their customers save on software licensing. It seems that the industry is slowly heading towards counting each die (see Integrated circuit) as a processor, no matter how many cores each die has.

Embedded applications

Embedded computing operates in an area of processor technology distinct from that of "mainstream" PCs. The same technological drivers towards multicore apply here too. Indeed, in many cases the application is a "natural" fit for multicore technologies, if the task can easily be partitioned between the different processors.

In addition, embedded software is typically developed for a specific hardware release, making issues of software portability, legacy code or supporting independent developers less critical than is the case for PC or enterprise computing. As a result, it is easier for developers to adopt new technologies and as a result there is a greater variety of multicore processing architectures and suppliers.

As of 2010, multi-core network processing devices have become mainstream, with companies such as Freescale Semiconductor, Cavium Networks, Wintegra and Broadcom all manufacturing products with eight processors.

In digital signal processing the same trend applies: Texas Instruments has the three-core TMS320C6488 and four-core TMS320C5441, Freescale the four-core MSC8144 and six-core MSC8156 (and both have stated they are working on eight-core successors). Newer entries include the Storm-1 family from Stream Processors, Inc (http://www.streamprocessors.com) with 40 and 80 general purpose ALUs per chip, all programmable in C as a SIMD engine and Picochip with three-hundred processors on a single die, focused on communication applications.

Hardware examples

Commercial

- Adapteva Epiphany, a many-core processor architecture with up to 4096 processors on-chip
- Aeroflex Gaisler LEON3, a multi-core SPARC that also exists in a fault-tolerant version.
- Ageia PhysX, a multi-core physics processing unit.
- Ambric Am2045, a 336-core Massively Parallel Processor Array (MPPA)
- AMD
  - Athlon 64, Athlon 64 FX and Athlon 64 X2 family, dual-core desktop processors.
  - Athlon II, dual-, triple-, and quad-core desktop processors.
  - Opteron, dual-, quad-, hex-, 8-, and 12-core server/workstation processors.
  - Phenom, dual-, triple-, quad-core processors.
  - Phenom II, dual-, triple-, quad-, and hex-core desktop processors.
  - Sempron X2, dual-core entry level processors.
  - Turion 64 X2, dual-core laptop processors.
  - Radeon and FireStream multi-core GPU/GPGPU (10 cores, 16 5-issue wide superscalar stream processors per core)
- Analog Devices Blackfin BF561, a symmetrical dual-core processor
- ARM MPCore is a fully synthesizable multicore container for ARM11 MPCore and ARM Cortex-A9 MPCore processor cores, intended for high-performance embedded and entertainment applications.
- ASOCS ModemX, up to 128 cores, wireless applications.
- Azul Systems
  - Vega 1, a 24-core processor, released in 2005.
  - Vega 2, a 48-core processor, released in 2006.
  - Vega 3, a 54-core processor, released in 2008.
- Broadcom SiByte SB1250, SB1255 and SB1455.
- ClearSpeed
  - CSX700, 192-core processor, released in 2008 (32/64-bit floating point; Integer ALU)
- Cradle Technologies CT3400 and CT3600, both multi-core DSPs.
- Cavium Networks Octeon, a 16-core MIPS MPU.
- Freescale Semiconductor QorIQ series processors, up to 8 cores, Power Architecture MPU.
- Hewlett-Packard PA-8800 and PA-8900, dual core PA-RISC processors.
- IBM
  - POWER4, the world's first non-embedded dual-core processor, released in 2001.
  - POWER5, a dual-core processor, released in 2004.
  - POWER6, a dual-core processor, released in 2007.
  - POWER7, a 4,6,8-core processor, released in 2010.
  - PowerPC 970MP, a dual-core processor, used in the Apple Power Mac G5.
  - Xenon, a triple-core, SMT-capable, PowerPC microprocessor used in the Microsoft Xbox 360 game console.
- Sony/IBM/Toshiba's Cell processor, a nine-core processor with one general purpose PowerPC core and eight specialized SPUs (Synergystic Processing Unit) optimized for vector operations used in the Sony PlayStation 3
- Infineon Danube, a dual-core, MIPS-based, home gateway processor.

- Intel
  - Celeron Dual-Core, the first dual-core processor for the budget/entry-level market.
  - Core Duo, a dual-core processor.
  - Core 2 Duo, a dual-core processor.
  - Core 2 Quad, 2 dual-core dies packaged in a multi-chip module.
  - Core i3, Core i5 and Core i7, a family of multi-core processors, the successor of the Core 2 Duo and the Core 2 Quad.
  - Itanium 2, a dual-core processor.
  - Pentium D, 2 single-core dies packaged in a multi-chip module.
  - Pentium Extreme Edition, 2 single-core dies packaged in a multi-chip module.
  - Pentium Dual-Core, a dual-core processor.
  - Teraflops Research Chip (Polaris), a 3.16 GHz, 80-core processor prototype, which the company originally stated would be released by 2011.[11]
  - Xeon dual-, quad-, hexa-, octo- and 12-core processors.

- IntellaSys
  - SEAforth 40C18, a 40-core processor[12]
  - SEAforth24, a 24-core processor designed by Charles H. Moore

- NetLogic Microsystems
  - XLP, a 32-core, quad-threaded MIPS64 processor
  - XLR, an eight-core, quad-threaded MIPS64 processor
  - XLS, an eight-core, quad-threaded MIPS64 processor

- Nvidia
  - GeForce 9 multi-core GPU (8 cores, 16 scalar stream processors per core)
  - GeForce 200 multi-core GPU (10 cores, 24 scalar stream processors per core)
  - Tesla multi-core GPGPU (10 cores, 24 scalar stream processors per core)

- Parallax Propeller P8X32, an eight-core microcontroller.
- picoChip PC200 series 200–300 cores per device for DSP & wireless
- Plurality HAL series tightly coupled 16-256 cores, L1 shared memory, hardware synchronized processor.
- Rapport Kilocore KC256, a 257-core microcontroller with a PowerPC core and 256 8-bit "processing elements". Is now out of business.
- SiCortex "SiCortex node" has six MIPS64 cores on a single chip.
- Sun Microsystems
  - MAJC 5200, two-core VLIW processor
  - UltraSPARC IV and UltraSPARC IV+, dual-core processors.
  - UltraSPARC T1, an eight-core, 32-thread processor.
  - UltraSPARC T2, an eight-core, 64-concurrent-thread processor.
  - UltraSPARC T3, an sixteen-core, 128-concurrent-thread processor.

- Texas Instruments
  - TMS320C80 MVP, a five-core multimedia video processor.
  - TMS320TMS320C66, 2,4,8 core dsp.

- Tilera
  - TILE64, a 64-core 32-bit processor
  - TILE-Gx, a 100-core 64-bit processor

- XMOS Software Defined Silicon quad-core XS1-G4

Free

- OpenSPARC

Academic

- MIT, 16-core RAW (http://groups.csail.mit.edu/cag/raw/) processor
- University of California, Davis, Asynchronous array of simple processors (AsAP)
    - 36-core 610 MHz AsAP
    - 167-core 1.2 GHz AsAP2
- University of Washington, Wavescalar (http://wavescalar.cs.washington.edu/) processor
- University of Texas, Austin, TRIPS processor

Notes

1. ^ Digital signal processors (DSPs) have utilized multi-core architectures for much longer than high-end general-purpose processors. A typical example of a DSP-specific implementation would be a combination of a RISC CPU and a DSP MPU. This allows for the design of products that require a general-purpose processor for user interfaces and a DSP for real-time data processing; this type of design is common in mobile phones. In other applications, a growing number of companies have developed multi-core DSPs with very large numbers of processors.
2. ^ Two types of operating systems are able to utilize a dual-CPU multiprocessor: partitioned multiprocessing and symmetric multiprocessing (SMP). In a partitioned architecture, each CPU boots into separate segments of physical memory and operate independently; in an SMP OS, processors work in a shared space, executing threads within the OS independently.

See also

- Race condition
- Multicore Association
- Multithreading (computer architecture)
- Multiprocessing
- Hyper-threading
- Symmetric multiprocessing (SMP)
- Simultaneous multithreading (SMT)
- Multitasking
- OpenHMPP HPC Open Standard for Manycore Programming (http://www.openhmpp.org)
- Parallel computing
- PureMVC MultiCore – a modular programming framework
- XMTC
- Parallel Random Access Machine
- Partitioned global address space (PGAS)
- Thread

- GPGPU
- CUDA
- OpenCL (Open Computing Language), a framework for heterogeneous execution
- Ateji PX, an extension of the Java language for parallelism

References

1. ^ TechTarget > multi-core processor (http://searchdatacenter.techtarget.com/sDefinition/0,,sid80_gci1015740,00.html) LAST UPDATED: 27 Mar 2007
2. ^ "What makes parallel programming hard? (http://www.futurechips.org/tips-for-power-coders/parallel-programming.html) bu FutureChips
3. ^ Intel has unveiled a single-chip version of a 48-core CPU (http://www.pcper.com/article.php?aid=825) for software and circuit research in cloud computing: accessdate=2009-12-02. Intel has loaded Linux on each core; each core has an X86 architecture (http://news.bbc.co.uk/2/hi/technology/8392392.stm) : accessdate=2009-12-3
4. ^ Do multicores save energy? Not really.|url=http://www.futurechips.org/chip-design-for-all/a-multicore-save-energy.html
5. ^ Multi-core in the Source Engine (http://www.bit-tech.net/gaming/2006/11/02/Multi_core_in_the_Source_Engin/1.html)
6. ^ AMD: dual-core not for gamers... yet (http://www.theregister.co.uk/2005/04/22/amd_dual-core_games/)
7. ^ Gamebryo's Floodgate page (http://www.emergent.net/index.php/homepage/products-and-services/floodgate)
8. ^ "CPU designers debate multi-core future" (http://www.eetimes.com/showArticle.jhtml?articleID=206105179) , by Rick Merritt, EE Times 2008
9. ^ Multicore packet processing Forum (http://multicorepacketprocessing.com/)
10. ^ Multicore Processor Licensing (http://www.microsoft.com/licensing/highlights/multicore.mspx)
11. ^ 80-core prototype from Intel (http://techfreep.com/intel-80-cores-by-2011.htm)
12. ^ "40-core processor with Forth-based IDE tools unveiled" (http://www.embedded.com/products/integratedcircuits/210603674)

External links

This article's use of external links **may not follow Wikipedia's policies or guidelines**. Please improve this article (http://en.wikipedia.org/w/index.php?title=Multi-core_processor&action=edit) by removing excessive and inappropriate external links. *(September 2009)*

- Embedded moves to multicore (http://embedded-computing.com/embedded-moves-multicore)
- Multicore Packet Processing Forum (http://multicorepacketprocessing.com)
- Parallel Computing Research wiki: "Chip Multiprocessor Comparison Chart" (http://view.eecs.berkeley.edu/wiki/Chip_Multi_Processor_Watch) *(Additions welcome)*
- A Berkeley View on the Parallel Computing Landscape (http://view.eecs.berkeley.edu) Argues for the desperate need to innovate around "manycore".
- Multicore: Multi-nonsense or Multi-opportunity? (http://www.futurechips.org/thoughts-for-researchers/multi-core-multi-nonsense-multi-opportunit.html)
- BMDFM (http://bmdfm.com) : Binary Modular Dataflow Machine – Multi-core Runtime Environment (BMDFM)
- Intel Tera-scale Computing Research Program (http://www.intel.com/go/terascale/)
- Overview of Intel's Dual Core CPUs' Specifications (Intel's website)

(http://www.intel.com/products/processor/core2duo/specifications.htm?iid=prod_core2duo+tab_spec)

- Multicore Programming blog (http://www.cilk.com/multicore-blog/)
- Multicore News blog (http://www.multicoreinfo.com)
- e-Book on Multicore Programming (http://www.cilk.com/multicore-e-book/) e-Book outlining multicore programming challenges, and the leading programming approaches to deal with them.
- XMTC: PRAM-like Programming – Software release (http://sourceforge.net/projects/xmtc/)
- Online multicore community (http://multicore.ning.com/)
- IEEE: Multicore Is Bad News For Supercomputers (http://spectrum.ieee.org/nov08/6912) for some computing tasks, 8 cores aren't (yet) much better than 4
- Muticore short course at MIT (http://web.mit.edu/professional/short-programs/courses/multicore_programming.html)
- Diploma thesis: A Virtual Platform for High Speed Message-Passing-Hardware Research (http://rechner-architektur.de/mpi-research/) A virtual network interface for many core CPUs
- IBM X10 - The New Concurrent Programming Language for Multicore and Petascale computing (http://x10-lang.org/)
- Ateji PX, an extension of the Java language for parallelism (http://www.ateji.com/px/)

	CPU technologies	[hide]
Architecture		ISA : CISC · EDGE · EPIC · MISC · OISC · RISC · VLIW · NISC · ZISC · Harvard architecture · von Neumann architecture · 4-bit · 8-bit · 12-bit · 16-bit · 18-bit · 24-bit · 31-bit · 32-bit · 36-bit · 48-bit · 64-bit · 128-bit · Comparison of CPU architectures
Parallelism	**Pipeline**	Instruction pipelining · In-order & out-of-order execution · Register renaming · Speculative execution · Hazards
	Level	Bit · Instruction · Superscalar · Data · Task
	Threads	Multithreading · Simultaneous multithreading · Hyperthreading · Superthreading
	Flynn's taxonomy	SISD · SIMD · MISD · MIMD
Types		Digital signal processor · Microcontroller · System-on-a-chip · Vector processor
Components		Arithmetic logic unit (ALU) · Barrel shifter · Floating-point unit (FPU) · Back-side bus · Multiplexer · Demultiplexer · Registers · Memory management unit (MMU) · Translation lookaside buffer (TLB) · Cache · Register file · Microcode · Control unit · Clock rate
Power management		APM · ACPI · Dynamic frequency scaling · Dynamic voltage scaling · Clock gating

Retrieved from "http://en.wikipedia.org/wiki/Multi-core_processor"

Categories: Computer architecture | Digital signal processing | Microprocessors | Parallel computing | Flynn's Taxonomy

- This page was last modified on 1 September 2011 at 11:21.
-