System & Service Management
Prozessorarchitekturen

Multiprocessing and Multithreading
Computer architects have become stymied by the growing mismatch in CPU operating frequencies and DRAM access times. None of the techniques that exploited instruction-level parallelism within one program could make up for the long stalls that occurred when data had to be fetched from main memory. Additionally, the large transistor counts and high operating frequencies needed for the more advanced ILP techniques required power dissipation levels that could no longer be cheaply cooled. For these reasons, newer generations of computers have started to exploit higher levels of parallelism that exist outside of a single program or program thread.

This trend is sometimes known as throughput computing. This idea originated in the mainframe market where online transaction processing emphasized not just the execution speed of one transaction, but the capacity to deal with massive numbers of transactions. With transaction-based applications such as network routing and web-site serving greatly increasing in the last decade, the computer industry has re-emphasized capacity and throughput issues.

One technique of how this parallelism is achieved is through multiprocessing systems, computer systems with multiple CPUs. Once reserved for high-end mainframes and supercomputers, small scale (2-8) multiprocessors servers have become commonplace for the small business market. For large corporations, large scale (16-256) multiprocessors are common. Even personal computers with multiple CPUs have appeared since the 1990s.

With further transistor size reductions made available with semiconductor technology advances, multicore CPUs have appeared where multiple CPUs are implemented on the same silicon chip. Initially used in chips targeting embedded markets, where simpler and smaller CPUs would allow multiple instantiations to fit on one piece of silicon. By 2005, semiconductor technology allowed dual high-end desktop CPUs CMP chips to be manufactured in volume. Some designs, such as Sun Microsystems' UltraSPARC T1 have reverted back to simpler (scalar, in-order) designs in order to fit more processors on one piece of silicon.

Another technique that has become more popular recently is multithreading. In multithreading, when the processor has to fetch data from slow system memory, instead of stalling for the data to arrive, the processor switches to another program or program thread which is ready to execute. Though this does not speed up a particular program/thread, it increases the overall system throughput by reducing the time the CPU is idle.

Conceptually, multithreading is equivalent to a context switch at the operating system level. The difference is that a multithreaded CPU can do a thread switch in one CPU cycle instead of the hundreds or thousands of CPU cycles a context switch normally requires. This is achieved by replicating the state hardware (such as the register file and program counter) for each active thread.

A further enhancement is simultaneous multithreading. This technique allows superscalar CPUs to execute instructions from different programs/threads simultaneously in the same cycle.

http://en.wikipedia.org/wiki/Multiprocessing

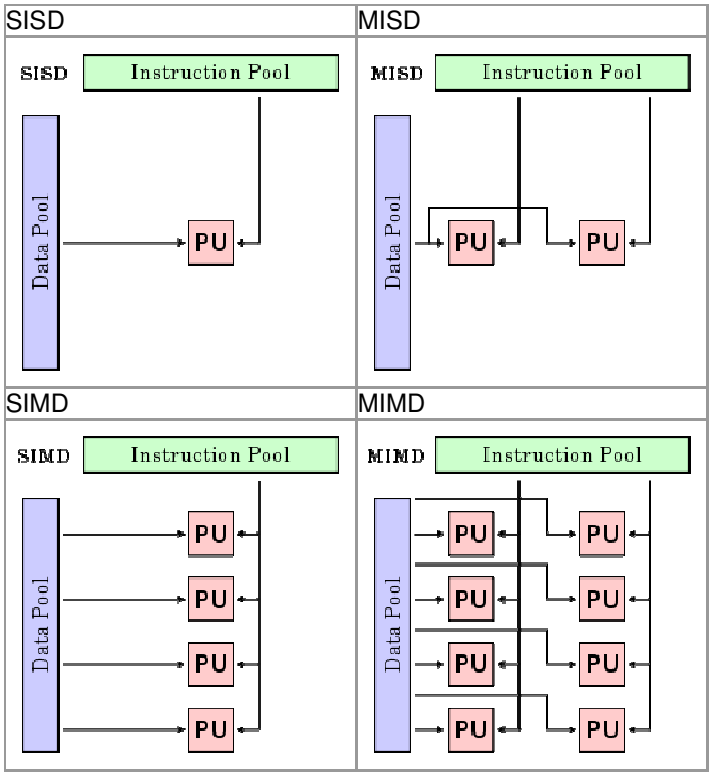
Instruction Pipelining
An instruction pipeline is a technique used in the design of computers and other digital electronic devices to increase their instruction throughput (the number of instructions that can be executed in a unit of time).

The fundamental idea is to split the processing of a computer instruction into a series of independent steps, with storage at the end of each step. This allows the computer's control circuitry to issue instructions at the processing rate of the slowest step, which is much faster than the time needed to perform all steps at once. The term pipeline refers to the fact that each step is carrying data at once (like water), and each step is connected to the next (like the links of a pipe.)

http://en.wikipedia.org/wiki/Instruction_pipeline

Flynn's Taxonomy
The four classifications defined by Flynn are based upon the number of concurrent instruction (or control) and data streams available in the architecture:

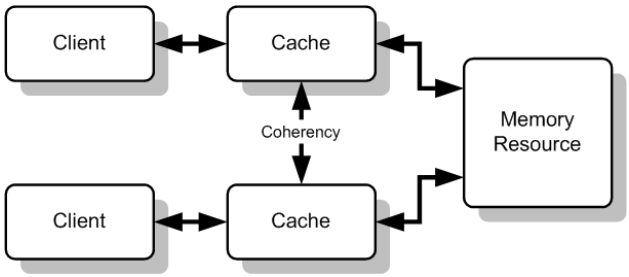
- Single Instruction, Single Data stream (SISD)
  A sequential computer which exploits no parallelism in either the instruction or data streams. Examples of SISD architecture are the traditional uniprocessor machines like a PC (currently manufactured PC's have multiple processors) or old mainframes.
- Single Instruction, Multiple Data streams (SIMD)
  A computer which exploits multiple data streams against a single instruction stream to perform operations which may be naturally parallelized. For example, an array processor or GPU.
- Multiple Instruction, Single Data stream (MISD)
  Multiple instructions operate on a single data stream. Uncommon architecture which is generally used for fault tolerance. Heterogeneous systems operate on the same data stream and must agree on the result. Examples include the Space Shuttle flight control computer.
- Multiple Instruction, Multiple Data streams (MIMD)
  Multiple autonomous processors simultaneously executing different instructions on different data. Distributed systems are generally recognized to be MIMD architectures; either exploiting a single shared memory space or a distributed memory space.



Non-Uniform Memory Access
Non-Uniform Memory Access or Non-Uniform Memory Architecture (NUMA) is a computer memory design used in multiprocessors, where the memory access time depends on the memory location relative to a processor. Under NUMA, a processor can access its own local memory faster than non-local memory, that is, memory local to another processor or memory shared between processors.
http://en.wikipedia.org/wiki/Non-Uniform_Memory_Access

Uniform Memory Access
Uniform Memory Access (UMA) is a shared memory architecture used in parallel computers. All the processors in the UMA model share the physical memory uniformly. In an UMA architecture, access time to a memory location is independent of which processor makes the request or which memory chip contains the transferred data. Uniform Memory Access computer architectures are often contrasted with Non-Uniform Memory Access (NUMA) architectures.
In the UMA architecture, each processor may use a private cache. Peripherals are also shared in some fashion, The UMA model is suitable for general purpose and time sharing applications by multiple users. It can be used to speed up the execution of a single large program in time critical applications. Unified Memory Architecture (UMA) is a computer architecture in which graphics chips are built into the motherboard and part of the computer's main memory is used for video memory.
http://en.wikipedia.org/wiki/Uniform_Memory_Access

Cache-Kohärenz
Durch die Sicherstellung von Cache-Kohärenz wird bei Mehrprozessorsystemen mit mehreren CPU-Caches verhindert, dass die einzelnen Caches für die gleiche Speicheradresse unterschiedliche (inkonsistente) Daten zurückliefern.



Eine temporäre Inkonsistenz zwischen Speicher und den Caches ist zulässig, sofern diese spätestens bei lesenden Zugriffen identifiziert und behoben wird. Inkonsistenzen werden z. B. durch das Rückschreibeverfahren (Write-Back) erzeugt, das im Gegensatz zu einem Durchschreibeverfahren (Write-Through) beim Schreiben in den Cache-Speicher nicht sofort den Hauptspeicher aktualisiert.
http://de.wikipedia.org/wiki/Cache-Koh%C3%A4renz

Crossbar Switch
Ein Koppelfeld, früher auch als Kreuzschienenverteiler (engl. crossbar switch) bezeichnet, dient zur Vermittlung und Durchschaltung von digitalen oder analogen Signalen. Es zählt zu den so genannten Raummultiplexverfahren in der Kommunikationstechnik. Dabei werden Eingangssignale völlig transparent (ohne Änderungen oder Verfälschungen) auf entsprechende Ausgänge geschaltet.

Ein Koppelfeld bezeichnet eine zusammen geschaltete Matrix (so genannte Koppelvielfache) von kommenden und gehenden Leitungen. Das Koppelfeld hat hierbei die Aufgabe, die Eingangsleitungen über so genannte Koppelpunkte (Switch) auf die Ausgangsleitungen durchzuschalten.

Wenn das Koppelfeld mehr Eingangsleitungen hat als Ausgangsleitungen wird es „blockierend“ genannt, weil nicht alle Eingangsleitungen gleichzeitig auf Ausgangsleitungen durchgeschaltet werden können.
http://de.wikipedia.org/wiki/Crossbar_Switch
http://en.wikipedia.org/wiki/Crossbar_Switch

More on this…
http://en.wikipedia.org/wiki/UNIX_System_V
http://en.wikipedia.org/wiki/BSD