TEMPLATE FOR DEFINITIONAL ENTRY

Main Memory

Peter Boncz
Centrum voor Wiskunde en Informatica (CWI)
Kruislaad 413, 1098 SJ Amsterdam, The Netherlands
boncz@cwi.nl

SYNONYMS
Primary Memory, Random Access Memory (RAM)

DEFINITION
Primary storage, presently known as main memory, is the largest memory directly accessible to the CPU in the prevalent Von Neumann model and stores both data and instructions (program code). The CPU continuously reads instructions stored there and executes them. Also called Random Access Memory (RAM), to indicate that load/store instructions can access data at any location at the same cost, ir is usually implemented using DRAM chips, which are connected to the CPU and other peripherals (disk drive, network) via a bus.

MAIN TEXT
The earliest computers used tubes, then transistors and since the 1970s in integrated circuits. RAM chips generally store a bit of data in either the state of a flip-flop, as in SRAM (static RAM), or as a charge in a capacitor (or transistor gate), as in DRAM (dynamic RAM). Some types have circuitry to detect and/or correct random faults called memory errors in the stored data, using parity bits or error correction codes (ECC). RAM of the read-only type, ROM, instead uses a metal mask to permanently enable/disable selected transistors, instead of storing a charge in them.

The main memory available to a program in most operating systems, while primariy relying on RAM, can be increased by disk memory. That is, the memory access instructions supported by a CPU work on so-called virtual memory, where an abstract virtual memory space is divided into pages. At any time, a page either resides in a swap-file on disk or in RAM, where it must be in order for the CPU to access it. When memory is accessed, the Memory Management Unit (MMU) of the CPU transparently translates the virtual address into its current physical addres. If the memory page is not in RAM, it generates a page fault, to be a handled by the OS which then has to perform I/O to the swap file. If a high percentage of the memory access generates a page fault, this is called thrashing, and severely lowers performance.

Over the past decades, the density of RAM chips has increased, following a planned evolution of ever finer chip production process sizes, popularly known as `Moore's Law'. This has led to an increase in RAM capacity as well as bandwidth. Access latency has also decreased, however, the physical distance on the motherboard between DRAM chips and CPU results in a minimum access latency of around 50ns (real RAM latencies are often higher). In current multi-GHz CPUs this means that a memory acess instruction takes hundreds of cycles to execute. As typically a high percentage of instructions in a program can be memory access instructions (up to 33%) the RAM latency can seriously impact performance. This problem is known as the ``memory wall''.

To counter the performance problems of the memory wall, modern computer architecture now features a memory hierarchy that besides DRAM also includes SRAM cache memories, typically located on the CPU chip. Memory access instructions transfer memory in units of cache-lines, typically 64 bytes at a time (this cache line size is also related to the width of the memory bus). Memory access instruction first check whether the accessed cache line is in the highest (fastest/smallest) L1 cache. This takes just a few CPU cycles. If a cache miss occurs, the memory access instruction checks the next cache level. Only if no cache contains the cache line, memory access is performed. Therefore, just like virtual memory page thrashing, the CPU cache hit ratio achieved by a program now materially affects performance.

While in the past access to the DRAM chips over the bus was typically performed by a chipset, in between CPU and memory, some modern CPU architectures have moved the memory controller logic onto the CPU chip itself, which tends to reduce access latency. Also, to better serve the memory bandwidth requirement multi-CPU systems, modern architectures often have a dedicated memory bus between the CPU and DRAM. In a Symmetric Multi-Processing (SMP) this leads to a so-called Non-Uniform Memory Access architecture (NUMA), where access to the memory directly connected to a CPU is faster than access to the memory connected to another CPU.

While database systems traditionally focus on the disk access pattern (i.e. I/O), modern database systems, as well as main-memory database systems (that do not rely on I/O in the first place) now must carefully plan the in-memory data storage format used as well as the memory access patterns caused by query processing algorithms, in order to optimize the use of the CPU caches and avoid high cache miss ratios. The increased RAM sizes as well as the increased impact of I/O latency also leads to a trend to rely more on main memory as the preferred storage medium in database processing.

CROSS REFERENCES
Cache memory
Disk
CPU