Computer Organization

COMP2120

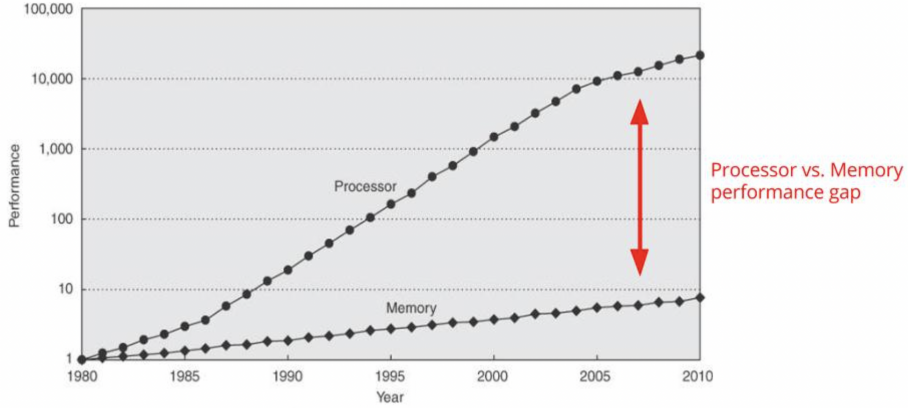
Qi Zhao

February 21, 2024

Memory Hierarchy

Performance of Logic vs Memory

Memory is a bottle neck for faster performance, because it is much slower than the CPU.



Memory system challenge

- Ideal memory: fast, cheap, and large
- No technology provides all three.
- Register, Cache: CMOS, SRAM, smaller, more expensive, faster
- Main memory, External memory: DRAM, Disk, larger, cheaper, slower
- Ideally, we would like to have all fast memory but the cost will be prohibitive.
- Would like to lower the cost, but still achieve acceptable performance.
- Use a hierarchy of memory

Memory Hierarchy

Go down the hierarchy

- Decreasing cost per bit
- Increasing capacity
- Increasing access time
- Decreasing frequency of access of the memory by the processor

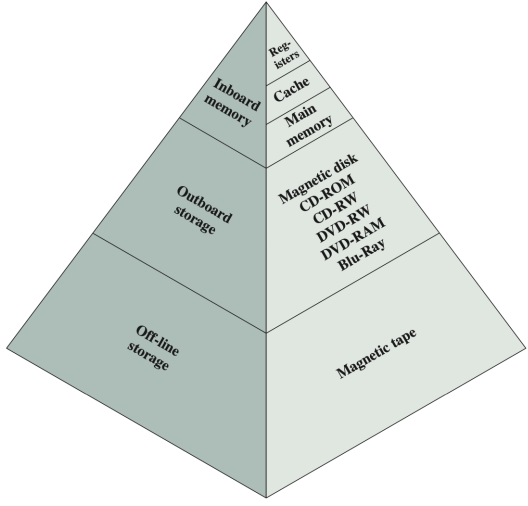
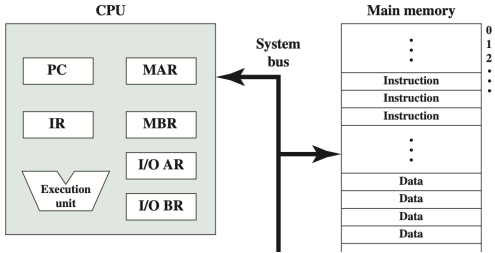


Figure 4.1 The Memory Hierarchy

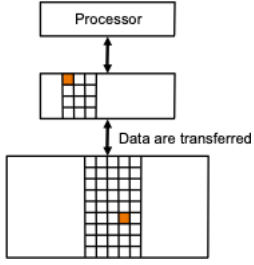
Memory Hierarchy



- The CPU reads memory by giving address and wait for data.
- It does not care what happens on the other side of the interface.
- You can add cache memory, virtual memory (hard disk), as long as the interface remains the same, i.e. the same address provides the same data (may be with different time delay).

Memory Hierarchy

- Achieve cost close to the lower level, i.e. use only a little high level, but a lot of lower level memory.
- Achieve speed close to a higher level, i.e. speed close to cache memory.
- When a datum is needed, it will be copied from the lower level to the higher level.
- Eventually, it will appear at the topmost level.



Principle of locality

A program references memory locations not uniformly. Some units are more likely to be accessed than others.

- **Temporal locality**, locality in time
  memory referenced in the recent past, e.g., instruction used in an iteration loop
- **Spatial locality**, locality in space
  memory whose addresses are near one another, e.g., arrays,
- `for(i=0; i<n; i++), {sum = sum + arr[i];}`
- So future access can be from higher level, e.g. cache memory, instead of lower level.

Characteristics of Memory Systems Location of memory

Go down the hierarchy

- Inside the processor — registers of the CPU.
- Inside the CPU chip — on-chip cache memory
- Cache Memory — on the motherboard
- Main Memory — on the motherborad
- External Memory — secondary memory: hard disks, or more recently SSDs.

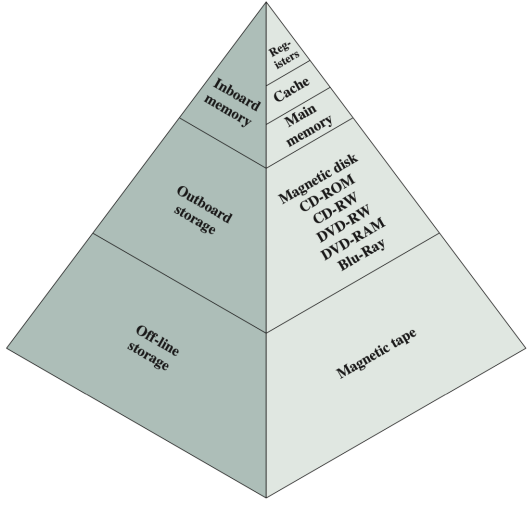


Figure 4.1 The Memory Hierarchy

Physical type and Characteristics

Physical Types:

- Semiconductor
- Magnetic disk or tape
- OpticalMagneto-optical

Characteristics:

- Volatile — data lost when power off.
- Nonvolatile — data is not lost even no power.
- Erasable/Nonerasable.

Capacity of memory

- Capacity is usually measured in MBytes/GBytes.
- Some system addresses memory by word (32 bits or 64 bits).
- Usually memory is byte addressable, i.e. each address is 1 byte.
- Hence one 32-bit word will occupy 4 addresses.

Organization

Memory Byte Ordering

- for multiple byte data, e.g. integer, or floating point numbers.
- A word may have 4 bytes. The bytes in a word can be numbered from left-to-right or right-to-left. e.g., Ox12345678, 32-bit,

Address: 100	101	102	103

- Big Endian Mode — Left-to-right ordering (numbering begins at the big. Example: IBM mainframes).

12	34	56	78
Address: 100	101	102	103

- Little Endian Mode — Right-to-left ordering Example: Intel family.

78	56	34	12
Address: 100	101	102	103

Unit of transfer

- Memory is a bottle neck for faster performance, because it is much slower than the CPU.
- CPU reads memory word by word.
- However, nowadays, CPU do not read directly from main memory, but from cache memory instead.
- Because of cache memory, unit of transfer will be 1 block (which contains, for example, 4KByte), although CPU access word by word (from cache).

Access Method

- **Sequential Access**: access the data in a sequential manner, one data after another. Have to start from the front (lower addresses) to access anything in the middle. (same as tapes)
- **Random Access**: can access any piece of data directly, by providing the address of the data. (same as array access). Constant latency
- **Associative** Content-addressable memory — address by content. use part of the content to retrieve the data. Used in cache memory.

Random Access Memory (RAM)

Traditional RAM is volatile. Must be provided with a constant power supply, otherwise the data are lost.

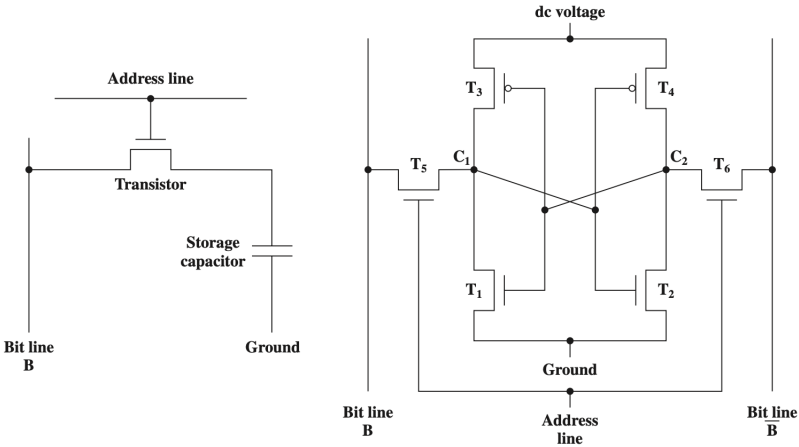
Dynamic RAM

- using transistors to store electric charges (by capacitance effect).
- need refreshing every few milliseconds (because charge will leak away)
- slower (capacitance causes a delay in from 0 to 1 and vice versa).
- used in Main memory
- 1 transister per cell (bit), much cheaper, e.g. 8GB DRAM cost HK$200-300, with access time: 50-60ns

Static RAM

- use logic gates to store data (e.g. Latch).
- no refreshing needed
- much faster
- used in cache memory
- 16Mbit static RAM costs about HK$200-300, with access time: about 10ns

Random Access Memory (RAM)



(a) Dynamic RAM (DRAM) cell

(b) Static RAM (SRAM) cell

Figure 5.2 Typical Memory Cell Structures

Characteristics of Memory devices

- **Registers**, CMOS, word
- **Cache**, SRAM, DRAM, block
- **Main memory**, DRAM, virtual memory page
- **External memory**, magnetic disk

ROM: read-only memory

ROM

- A ROM is created like any other integrated circuit chip, part of the fabrication process
- non-volatile memory. We need non-volatile memory to store the start-up program of the system, e.g. booting.
- **PROM**, programmable ROM, may be written into only once. The writing process is performed electrically and may be performed by a supplier or customer later than the original chip fabrication.
- ROM/PROM — cannot change content.

ROM: read-only memory

Read-mostly memory: read operations are far more frequent than write operations, nonvolatile storage is required

- Erasable programmable read-only memory (**EPROM**)- content of entire chip can be erased by intense ultraviolet light. Before a write operation, all erased to the same initial state
- Electrically erasable programmable read-only memory (**EEPROM**) — can be erased by using electric current (but slow). Only the byte or bytes addressed are updated.

Flash

Flash memory, non-volatile memory.

- Read-mostly memory
- Intermediate between **EPROM** and **EEPROM**
- faster than EEPROM in writing
- used in handheld device/mobile phones.
- has replaced EPROM/EEPROM for storing BIOS of PCs.
- Recently, used as SSD (Solid State Drives) in notebook computers
- Have a limited number of write cycles.

RAM, ROM, Flash

Table 5.1 Semiconductor Memory Types

Memory Type	Category	Erasure	Write Mechanism	Volatility
Random-access memory (RAM)	Read-write memory	Electrically, byte-level	Electrically	Volatile
Read-only memory (ROM)	Read-only memory	Not possible	Masks	Nonvolatile
Programmable ROM (PROM)			Electrically	
Erasable PROM (EPROM)	Read-mostly memory	UV light, chip-level		
Electrically Erasable PROM (EEPROM)		Electrically, byte-level		
Flash memory		Electrically, block-level		

Performance

- **Access time** — time to perform a read/write operation.
- **Memory cycle time** — access time + transfer time (i.e. time between two memory access).
- **Transfer rate** — how fast data can be transferred. It depends on how much data that can be addressed at the same time.
- For example, we can double "memory bandwidth" by providing 64-bit memory bus, instead of 32 bit, so that 2 words can be transferred at the same time.
- After an initial access time, a memory can provide consecutive data much faster – burst mode.
  e.g., transfer the first 4 words from main memory to cache is $55ns$, while each subsequent 4 words require $10ns$.

Performance of a two-levels of memory

Example: Consider a two level hierarchy (say main memory vs cache memory, note that the values are not those in main and cache memory, just for illustration purpose).

- Let the lower level has an access time of $T_2 = 0.1\mu s$,
- the upper level $T_1 = 0.01\mu s$.
- Furthermore, suppose 95% of the time, we can find the data at the upper level. Note that we still have to access the higher level first and if it cannot be found there, then go to the lower levels.
- The average time

$$0.95 \times 0.01 + 0.05 \times (0.1 + 0.01) = 0.015$$

  which is much closer to the upper level than the lower level.
- 95% hit-rate for cache memory is quite common.

Error detection Correction

- error may due to voltage spikes (during lightning), or electromagnetic interference (cosmic ray from outer space, e.g. in aeroplane, satellite), power supply problem
- Extra bits are required to perform error-detection or correction.
- Usually, not used in main memory, but used in secondary storage (i.e. Hard Disks).

Error detection Correction

- a single parity bit is attached to the bit pattern
- the parity bit is chosen such that the number of 1 s in the bit pattern (including the parity bit) is even (even parity) or odd (odd parity).
- Example: (parity in parenthesis)

$$\text{Even parity } 11010011(1)$$
$$\text{Odd parity } 11010011(0)$$

- it can detect 1-bit error, because if one of the bit is wrong, (i.e. changing from 1 to 0 or vice versa), the number of 1's will change from odd to even or from even to odd.
- need more bits to correct an error than detection.
- e.g. Hamming code is a kind of error-correcting code(ECC).