CSC227: Operating Systems
Spring 2017

CHAPTER 1

MEMORY STRUCTURE

Dr. Soha S. Zaghloul

LAYOUT

- Memory Types
- Memories' Speeds
- Direct Memory Access (DMA)
- Cache Memory

MEMORY TYPES (I)

Read-Only Memory (ROM)

- Cannot be erased.
- Used to store static programs such as bootstrap program & game cartridges.

Electrically Erasable Programmable ROM (EEPROM)

- Can be erased, but not frequently.
- Contains static programs such as those installed by the factory in smartphones.
- Flash memory is a special form of EEPROM.

Registers

- High-speed memory located on the processor chip (on-chip).
- Holds data for immediate use by the processor.
- They are the fastest amongst memory types. They have the same speed as the processor.
- They are the most expensive.
- They have very low capacity, only a few bytes.
- They are volatile.

Cache Memory

Main Memory

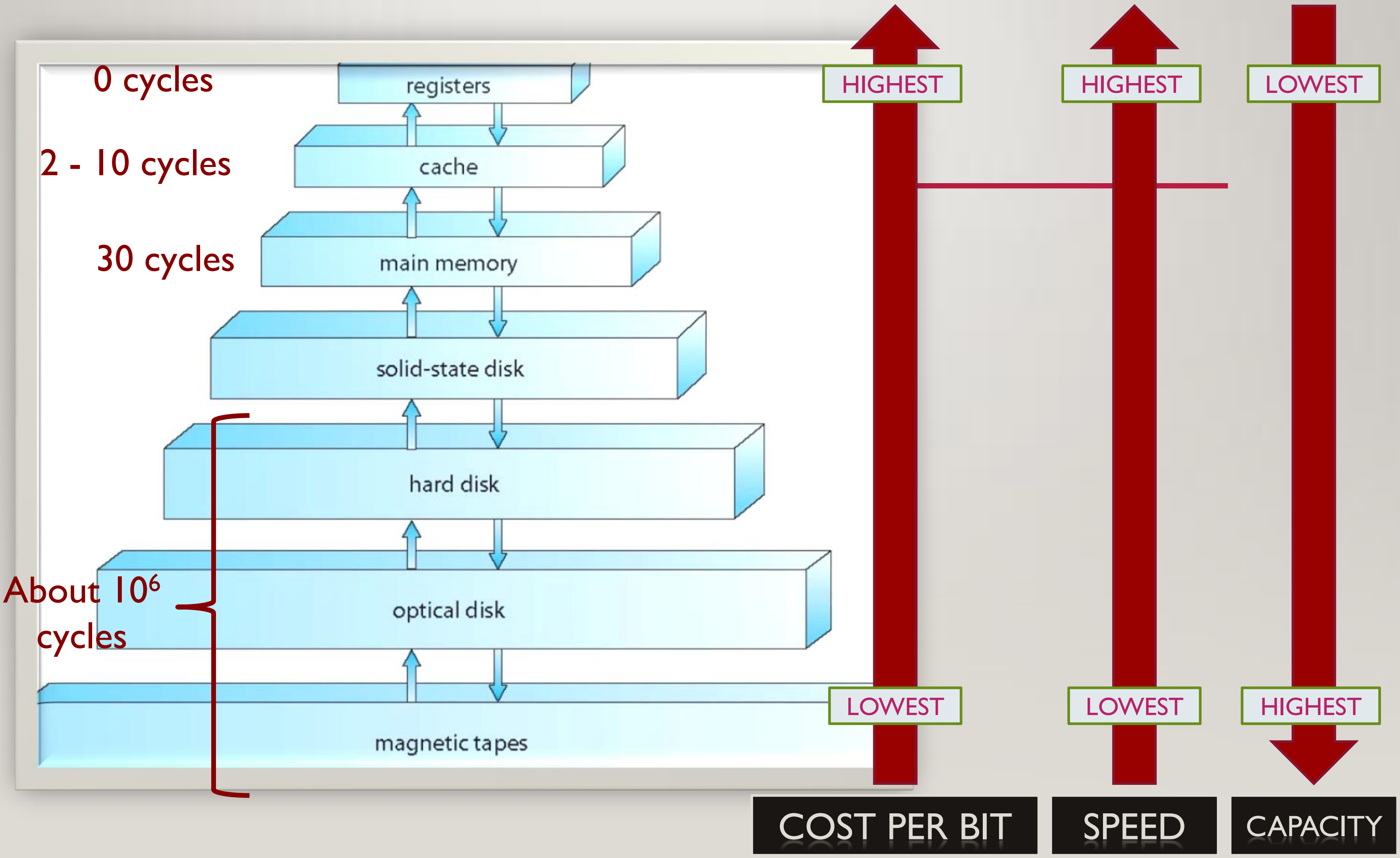
Hard Disk

Magnetic, Optical Disks and Magnetic Tapes

MEMORY TYPES (2)

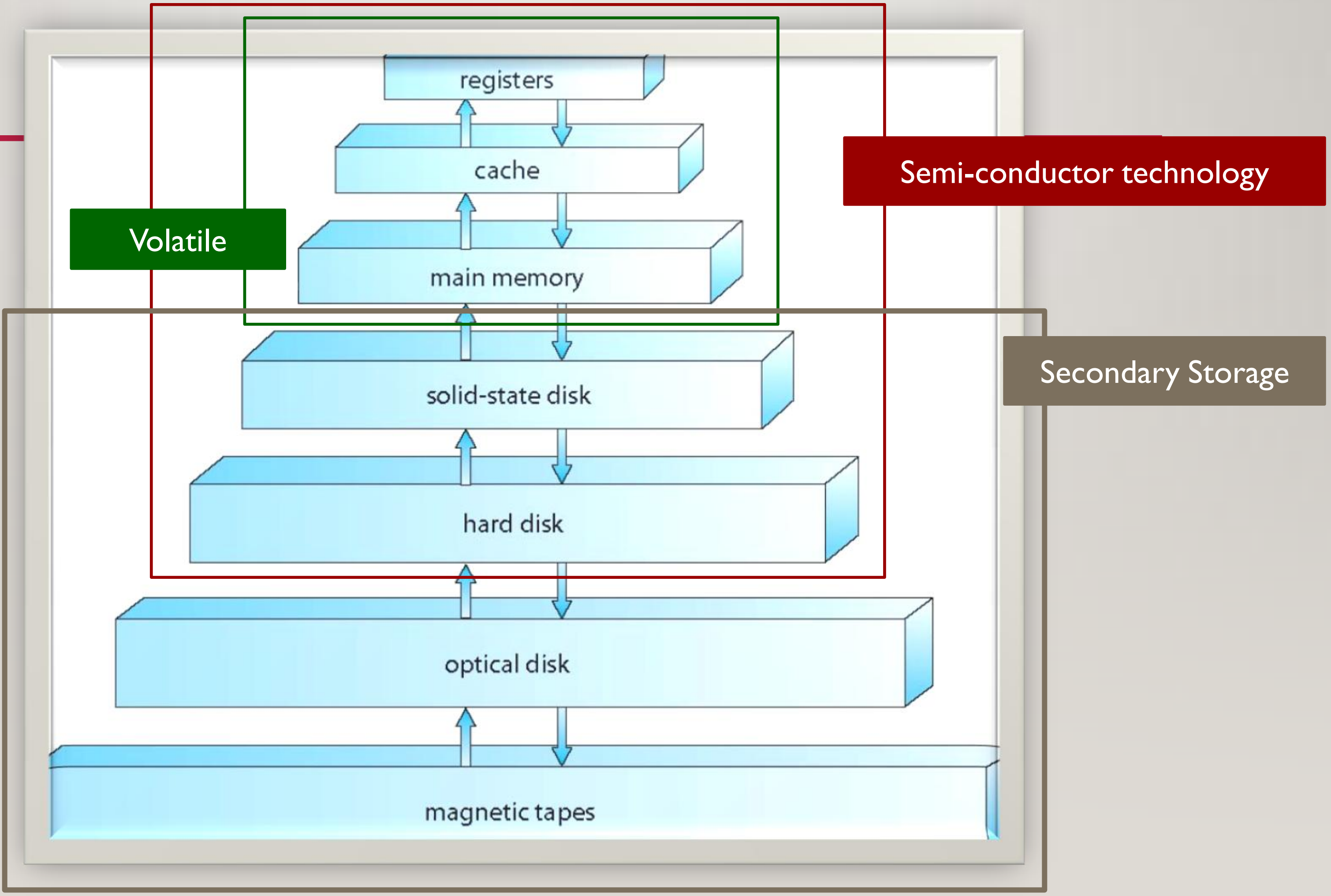
- Read-Only Memory (ROM)
- Electrically Erasable Programmable ROM (EEPROM)
- Registers
- Cache Memory
  - Relatively high-speed memory as compared to main memory.
  - located close to the processor (on-chip or off-chip).
  - Auxiliary memory, rewritable and volatile.
- Main Memory
  - Any program to be executed should be first loaded into the main memory.
  - Also, any data to be processed should be first loaded into the main memory.
  - Slower than the cache memory.
  - It has high capacity, rewritable and volatile.
- Hard Disk
  - A secondary storage device.
  - It has a low speed as compared to the main memory.
  - It has high capacity, rewritable and non-volatile.
- Magnetic, Optical Disks and Magnetic Tapes
  - A secondary storage device.
  - It has a lower speed as compared to the hard disk.
  - It has high capacity, rewritable and non-volatile.

MEMORIES' SPEEDS



Latency is measured in processor cycles.

MEMORIES – MAIN CHARACTERISTICS



Direct Memory Access (DMA) Technique for I/O

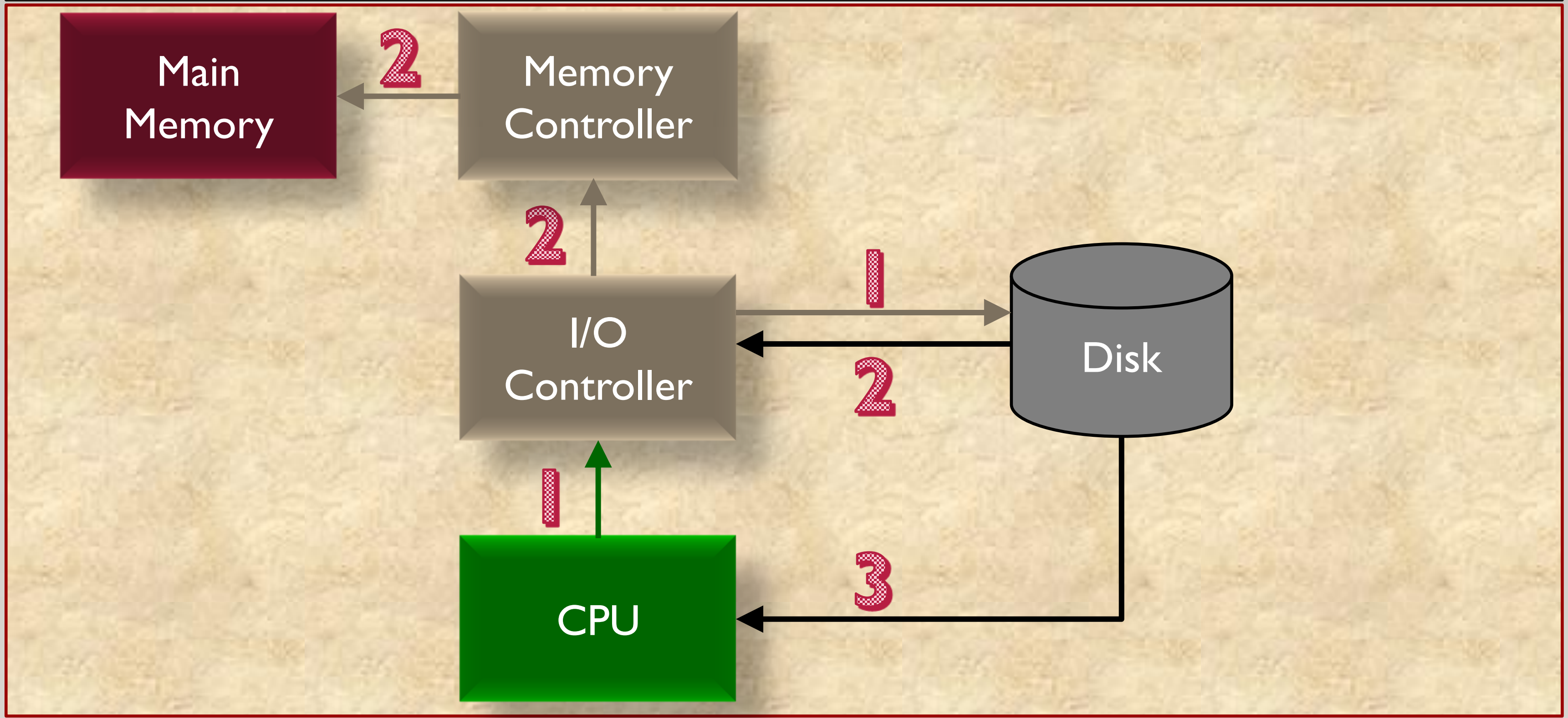
DMA is used for high-speed I/O devices that are able to transmit information at a speed close to that of the main memory.

In this technique, the device controller transfers an entire block of data (rather than bytes) directly to or from its own buffer storage to memory, with no intervention from the CPU.

Only one interrupt is generated per block, rather than per byte, to tell the device driver that the operation is completed.

While the device controller is performing these operations, the CPU is available to accomplish other work.

DIRECT MEMORY ACCESS (DMA) TECHNIQUE FOR I/O – DETAILED STEPS

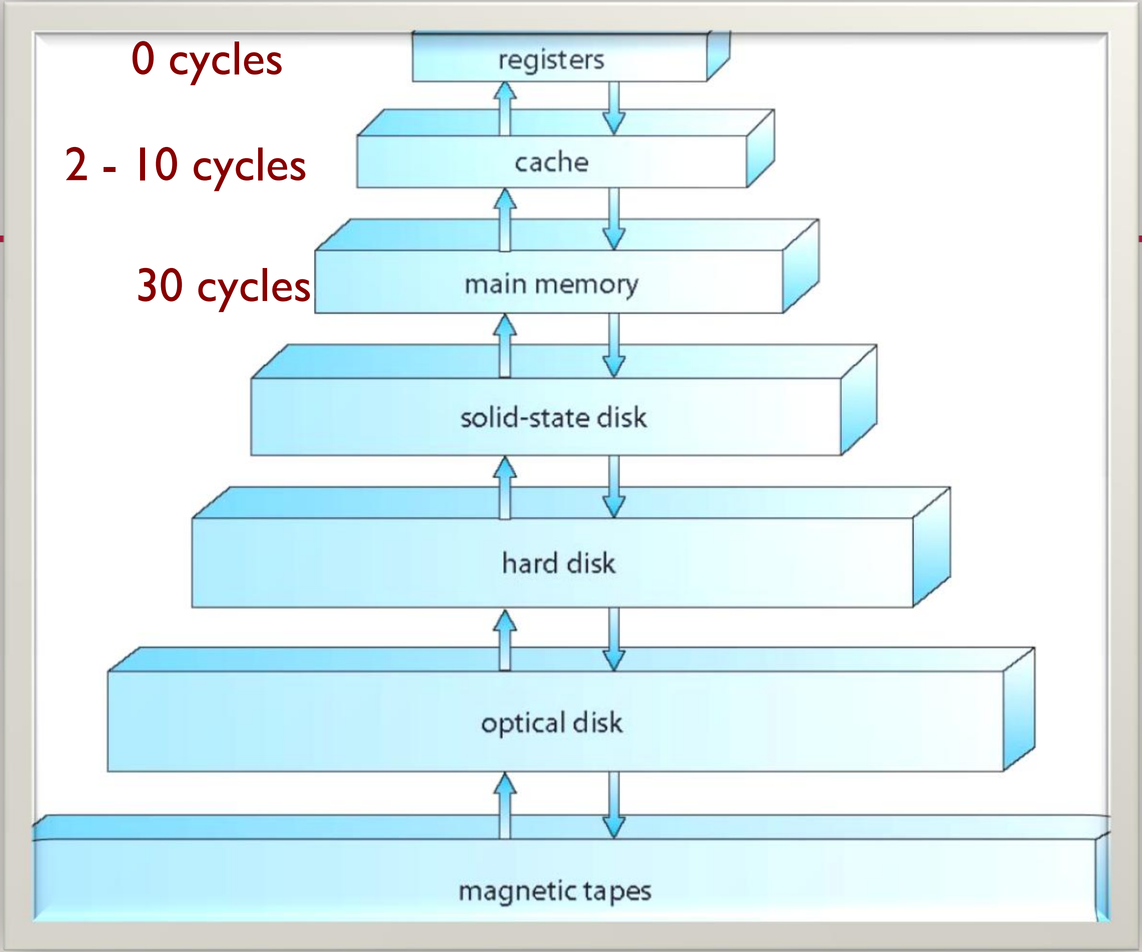


Step 1:
A processor sends an I/O request to the I/O controller, which sends the request to the disk. The processor continues executing instructions.

Step 2:
The disk sends data to the I/O controller; the data is placed at the memory address specified by the DMA command.

Step 3:
The disk sends an interrupt to the processor to indicate that the I/O is done.

CACHE MEMORY - INTRODUCTION



If the CPU needs to read/write from/into the memory, it stays idle until the needed variables are fetched (accessed).

The cache speeds the access of the CPU to the needed variables.

CACHE MEMORY – HIT/MISS

Caches are used in the following way:

Step 1:
When the CPU needs an address (data/instruction), it looks for it in the cache.

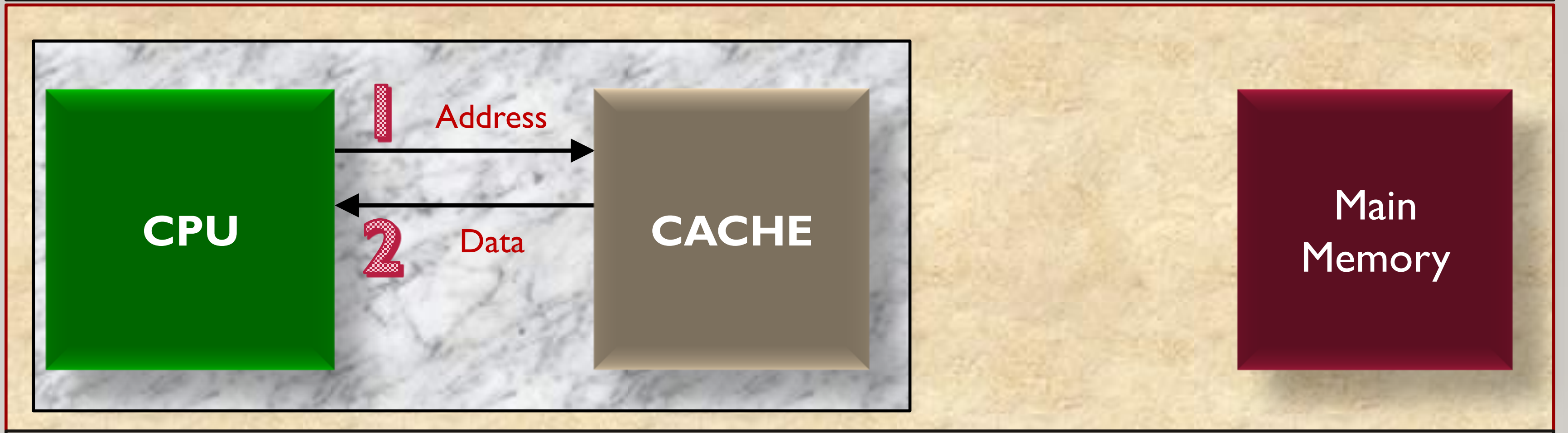
Step 2:
If found, we say that we have a cache hit. The address contents are returned to the CPU. The CPU uses them in its processing. Thus, multiple cycles are saved as compared to fetching the address contents from the main memory.

Step 3:
If not found, we say that we have a cache miss. The CPU then searches for the address in the main memory.

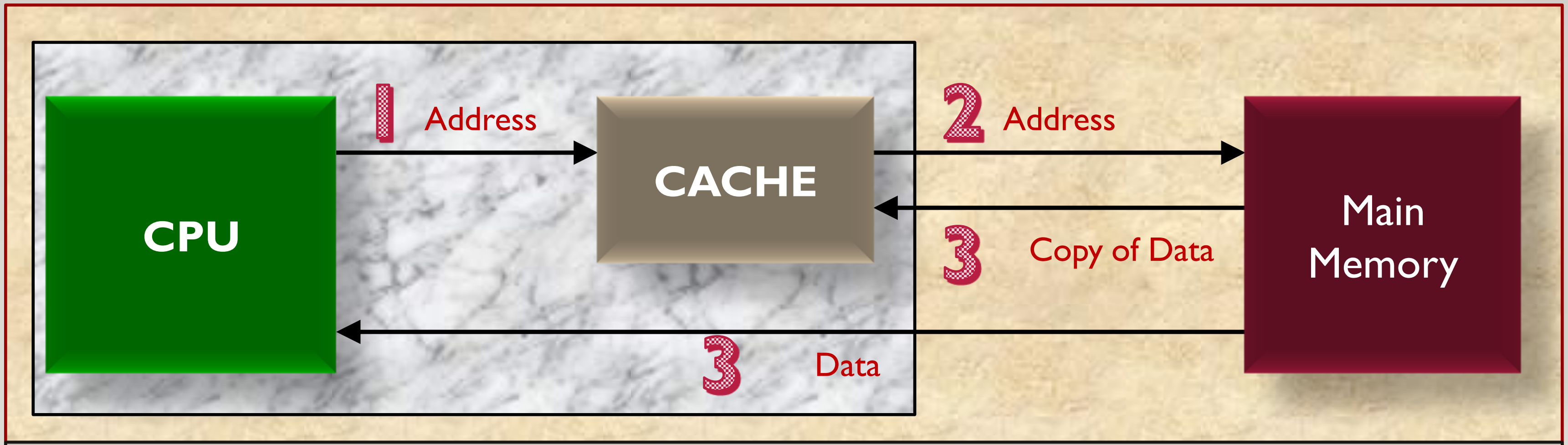
Step 4:
In the latter case, the CPU fetches the address contents from the main memory. A copy is also stored in the cache for possible future reference.

Thus, the contents of a cache is always a subset of those of a main memory.

Cache Memory – Illustration



(1) CACHE HIT



(2) CACHE MISS

Cache Memory – Performance

The performance of a cache memory is calculated as follows:

$$\text{Performance} = \frac{\text{Number of cache hits}}{\text{Total number of cache access}} * 100\%$$

Example:
A CPU accesses the cache 1000 times. The addresses are found 500 times. What is the performance of the cache?

Solution:

$$\text{Performance} = \frac{500}{1000} * 100\% = 50\%$$

Cache Memory – Multi-Level Caches

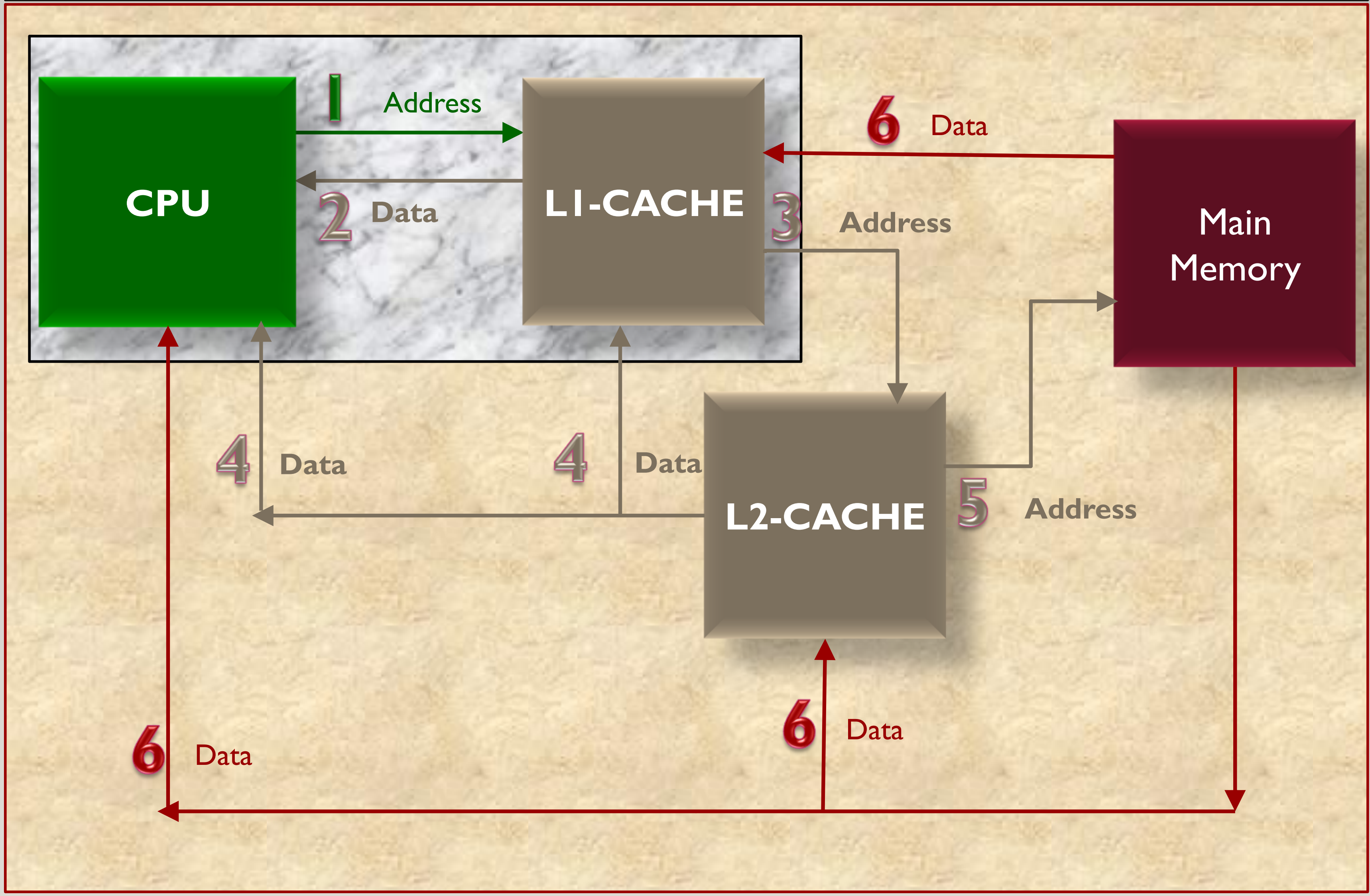
Multiple caches may be used. In this case, they are used as follows:

- Step 1:
  The CPU searches for an address in Level-1 cache.
- Step 2:
  If found, the address contents are returned to the CPU.
- Step 3:
  If not found, the CPU searches for the address in Level-2 cache.
- Step 4:
  If the address is found in Level-2 cache, then the address contents are returned to the processor, and a copy is made in Level-1 cache.
- Step 5:
  If the address is not found in Level-2 cache, then the CPU searches for it in the main memory. Address contents are submitted to the CPU. Two copies are made for both Level-1 and Level-2 caches.

In such an architecture, the contents of L1-cache are subset of L2-cache. The latter is subset of the main memory.

Note that the access to L1 cache is higher than that to L2 cache. The latter is higher than the access time to main memory.

CACHE MEMORY – MULTI-LEVEL CACHE ARCHITECTURE



Cache Memory – Performance of Multi-Level Caches

The miss rate of level-1 cache:

$$L1\ \text{Miss rate}\ (L1_{miss}) = \frac{\text{Number of L1-cache misses}}{\text{Total number of cache access}} * 100\%$$

The miss rate of level-2 cache:

$$L2\ \text{Miss rate}\ (L2_{miss}) = \frac{\text{Number of L2-cache misses}}{\text{Total number of cache access}} * L1_{miss} * 100\%$$

Note that the following formula is always true:

Miss ratio = 1 – Hit Ratio