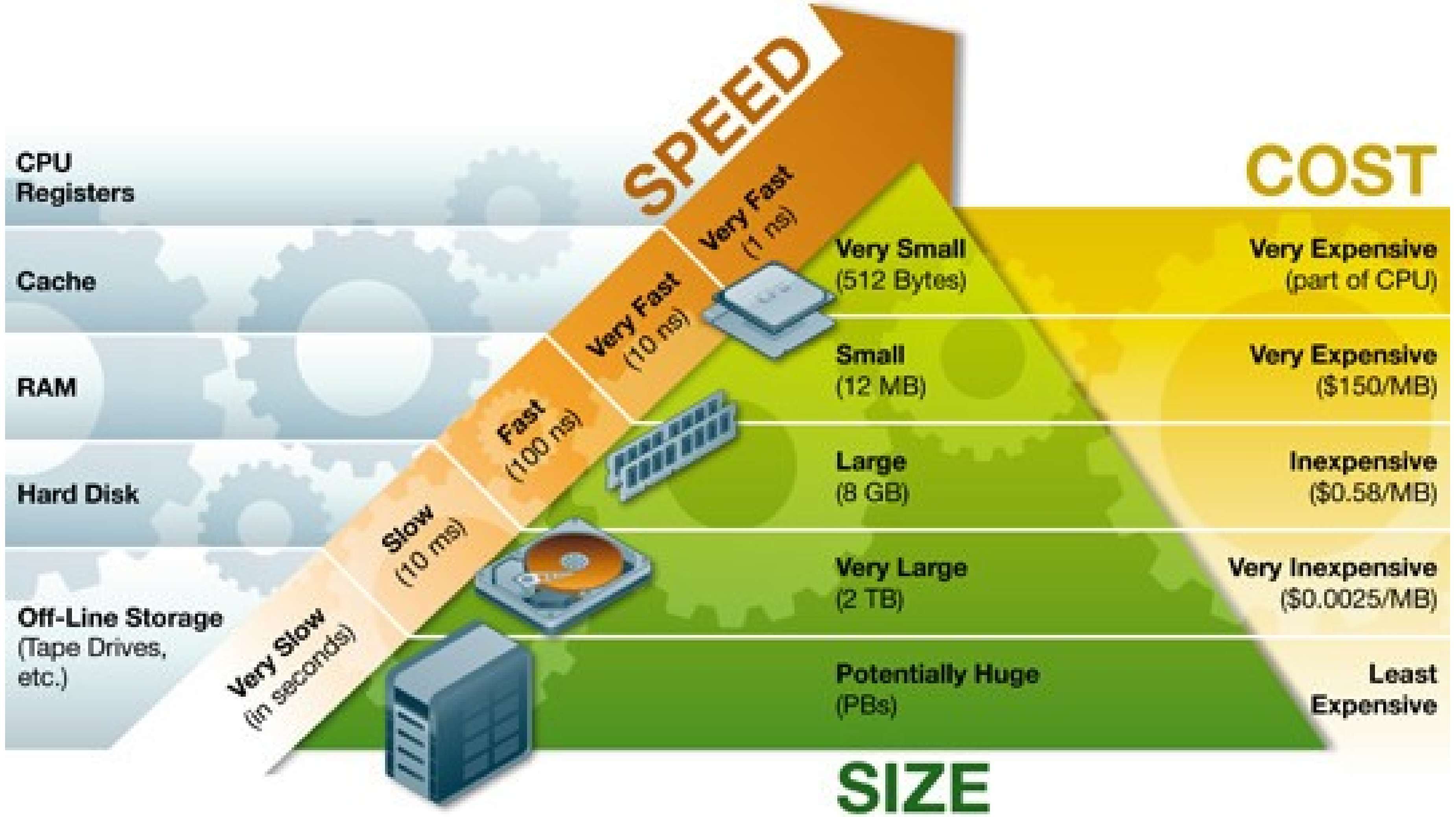
Functional Aspects of Memory & Memory Hierarchy

ICT 2203 Computer Architecture

Partially based on Computer Organization and Architecture 8th Edition by William Stallings

Memory Hierarchy

Memory Hierarchy



Location

- CPU
    - Registers, Control Memory
    - Cache – L1, L2, …
- Internal
    - Main memory (RAM), consists of DRAMs
- External
    - Secondary memory – hard disks
    - Removable media – ZIP, CD-ROM, Tape

Unit of Transfer

- Internal
  - #of bits read or written to/from main memory at a time
  - Determined by data bus width
  - May be different from word or addressable unit
  - On a 32 bit processor, this will be 32 bits. On a 64 bit processor, this will be 64 bits
- External
  - Usually a block which is much larger than a word

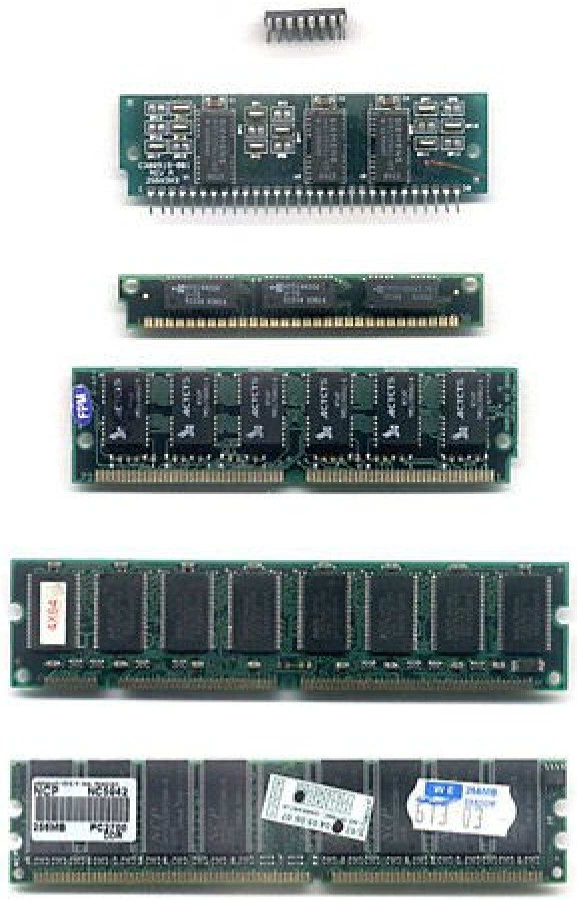
Memory Access Methods

- Sequential
  - Start at the beginning and read through in order
  - Access time depends on location of data and previous location
  - e.g. tape

- Direct
  - Individual blocks have unique address
  - Access is by jumping to block plus sequential search
  - Access time depends on location and previous location, but much faster than sequential access
  - e.g. disk

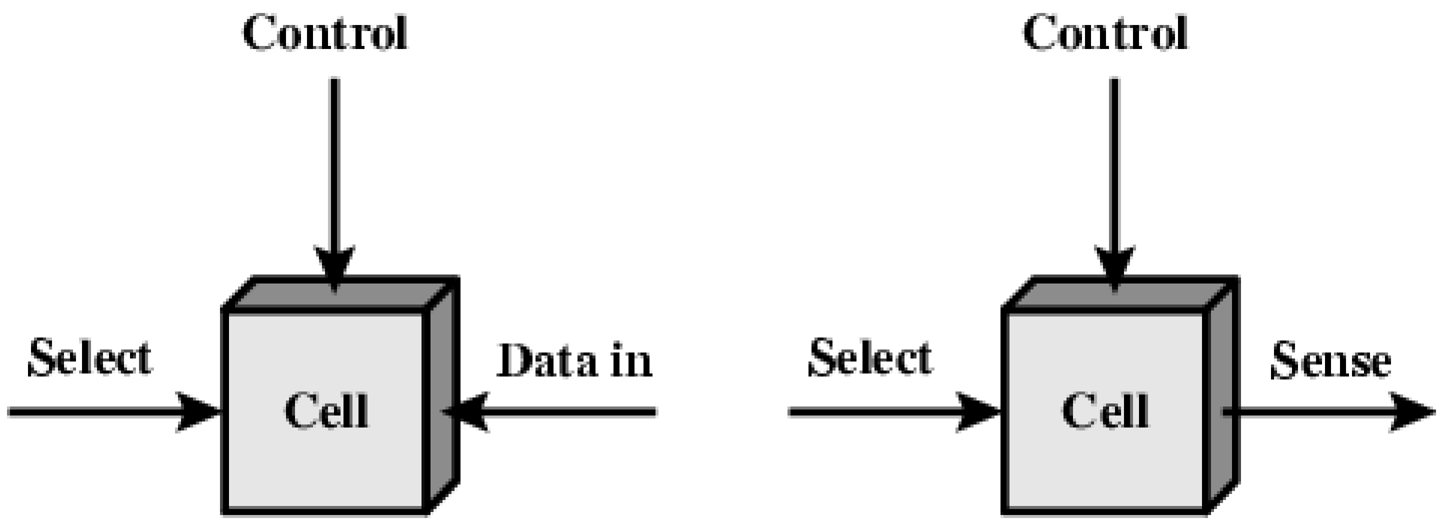
Semiconductor Memory

- RAM
    - Read/Write
    - Volatile
    - Temporary storage
    - Static or dynamic (will discuss littler later)



Memory Cell Operation

- Select terminal select the cell for read and write
- The control terminal indicated read or write



(a) Write (b) Read

Static RAM

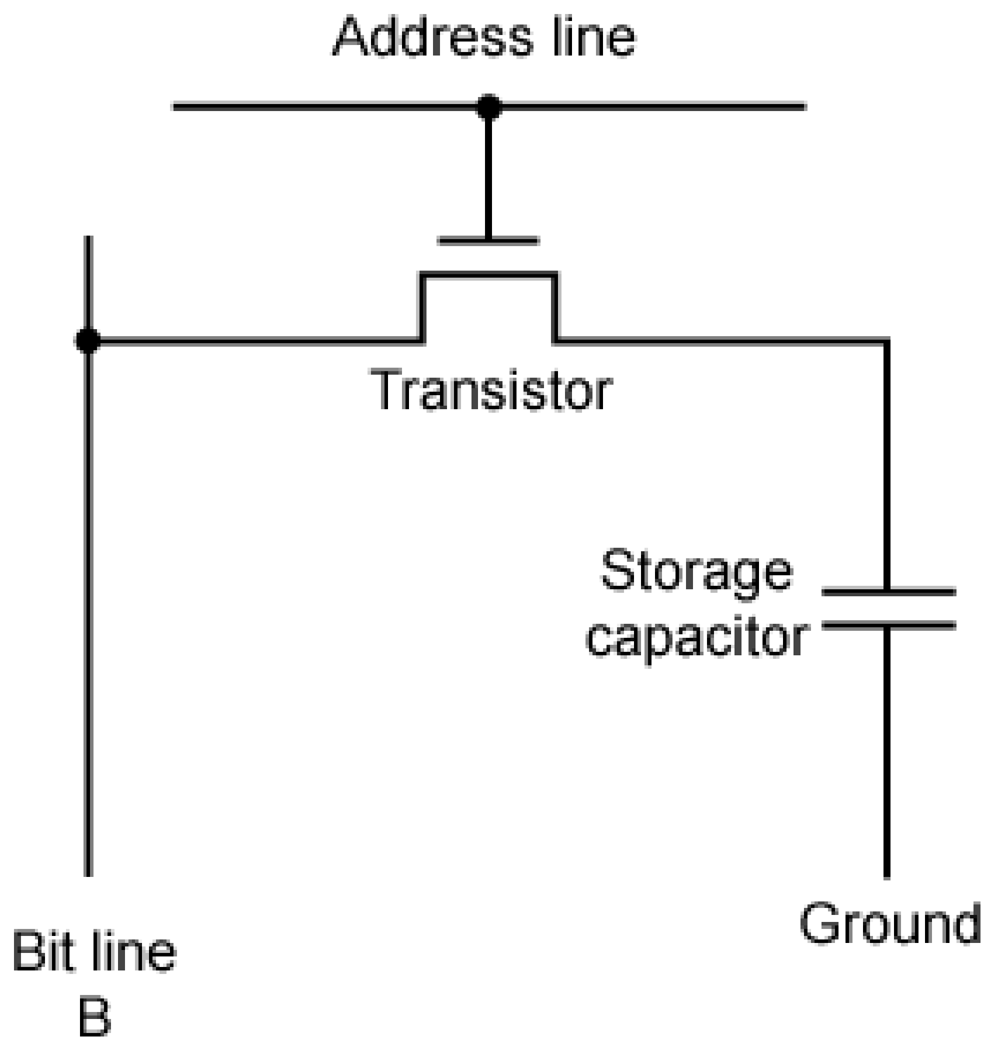
- Bits stored as on/off switches (e.g. flip flops)
- No charges to leak
- No refreshing needed when powered
- More complex construction
- More expensive
- Does not need refresh circuits
- Faster
- e.g. Cache

Dynamic RAM

- Bits stored as charge in capacitors
- Charges leak because of the nature of capacitors
- Need refreshing even when powered
- Simpler construction
- Less expensive
- Slower
- Ex: Main memory
- Essentially analogue
  - Level of charge determines value

Dynamic RAM Operation

- Address line
    - Active when bit read or written
    - Transistor switch closed (current flows)
- Write
    - Voltage to bit line
        - High for 1 low for 0
    - Then signal address line
        - Transfers charge to capacitor
- Read
    - Address line selected
        - transistor turns on
    - Charge from capacitor fed via bit line to sense amplifier



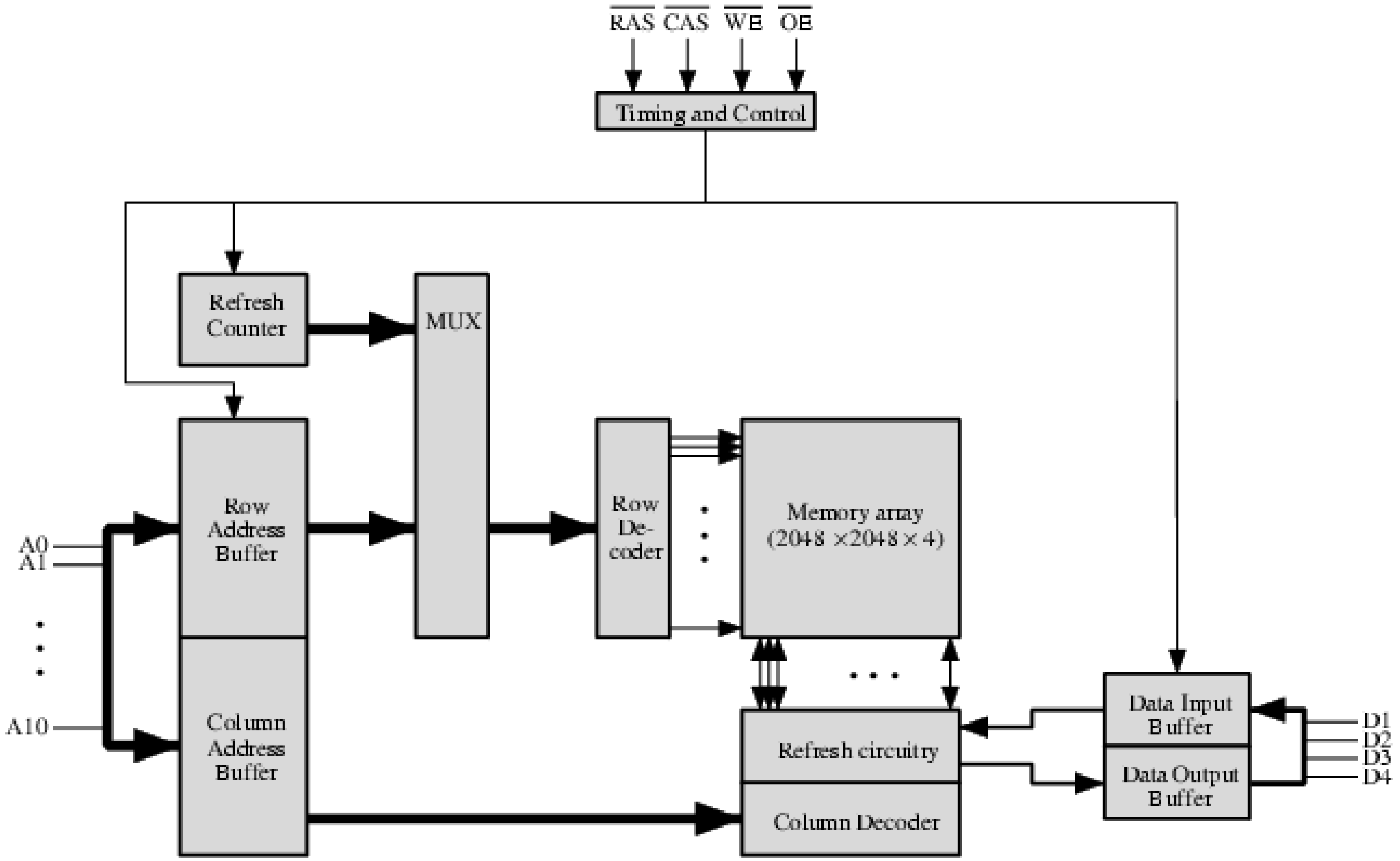
Refreshing

- Refresh circuit included on chip
- Disable chip
- Count through rows
- Read & Write back
- Takes time
- Slows down apparent performance

Organisation in detail

- A 16Mbit chip can be organised as 1M of 16 bit words
- A bit per chip system has 16 lots of 1Mbit chip with bit 1 of each word in chip 1 and so on
- A 16Mbit chip can be organised as a 2048 x 2048 x 4bit array
  - (16M = 224 = 211 * 211 * 4)
  - Reduces number of address pins
    - Multiplex row address and column address
    - 11 pins to address (211=2048)
    - Adding one more pin doubles range of values, so x 4 capacity

Typical 16 Mb DRAM (4M x 4)



SRAM v DRAM

- Both volatile
  - Power needed to preserve data
- Dynamic cell
  - Simpler to build, smaller
  - Less expensive
  - Needs refresh
  - Larger memory units
- Static
  - Faster
  - Cache

Read Only Memory (ROM)

- Permanent storage
  - Nonvolatile
- Microprogramming
- Library subroutines
- Systems programs (BIOS)

Types of ROM

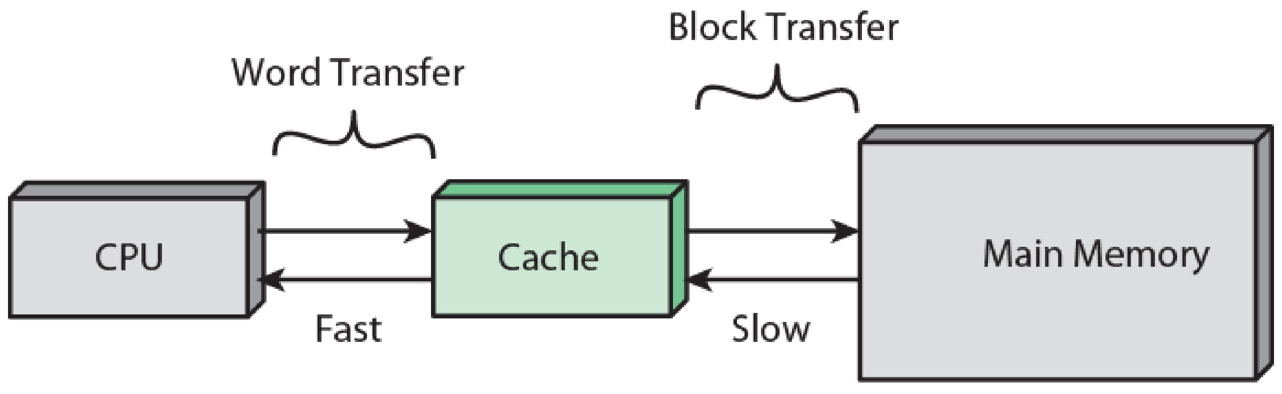
- Written during manufacture
  - Very expensive
- Programmable (once)
  - PROM
  - Needs special equipment to program
- Read “mostly”
  - Erasable Programmable (EPROM)
    - Erased by UV
  - Electrically Erasable (EEPROM)
    - Takes much longer to write than read
  - Flash memory
    - Erase whole memory electrically

External Memory Types

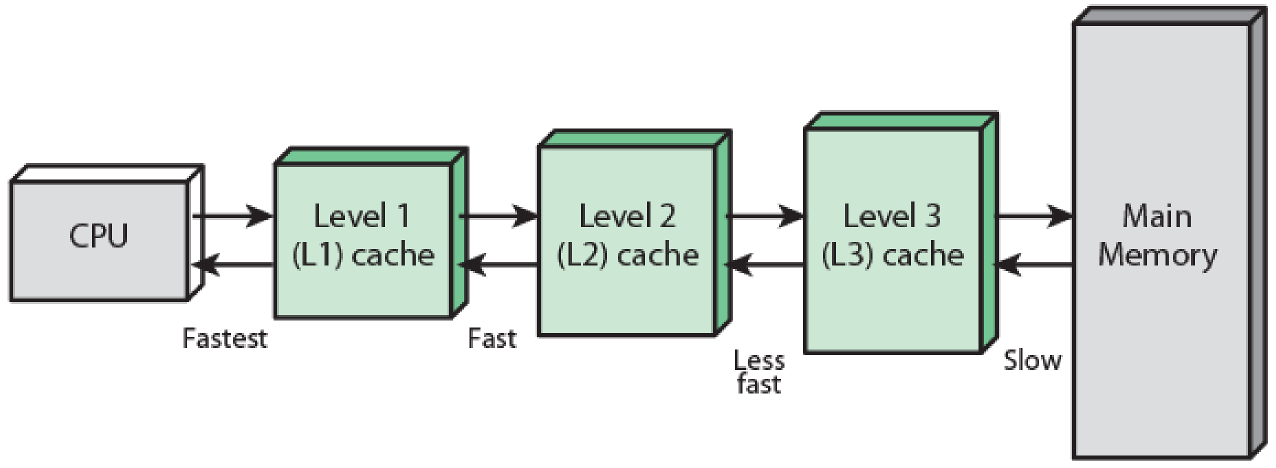
- HDD
  - Magnetic Disk(s)
  - SDD (Solid State Disk(s))
- Optical
  - CD-ROM
  - CD-Recordable (CD-R)
  - CD-R/W
  - DVD
- Magnetic Tape

Cache Memory

Cache Organization



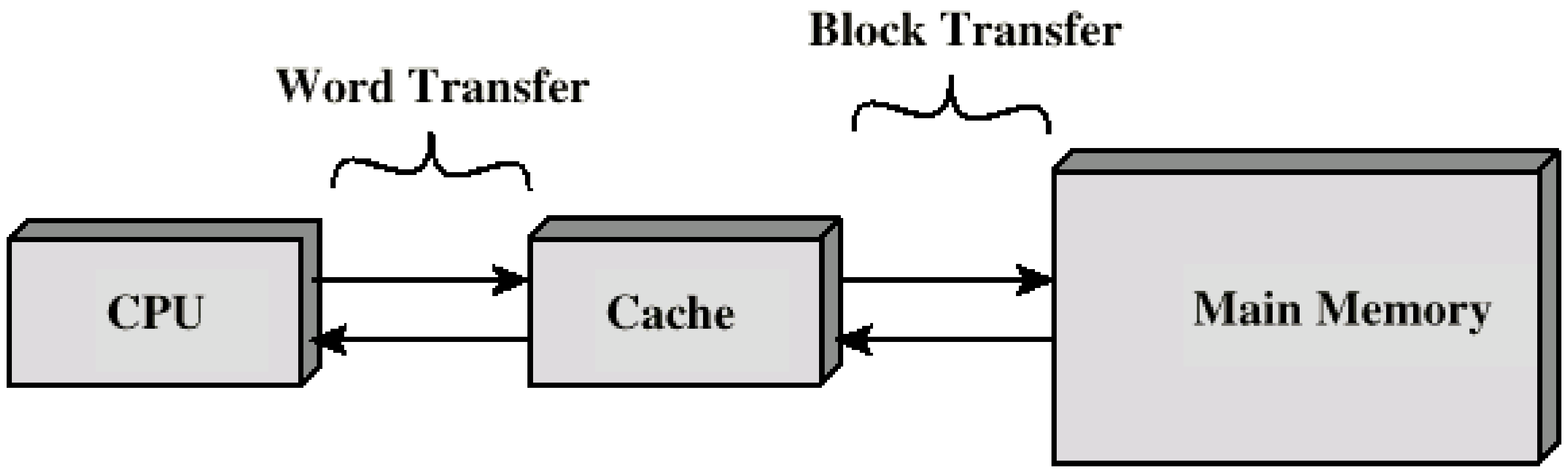
(a) Single cache



(b) Three-level cache organization

Cache

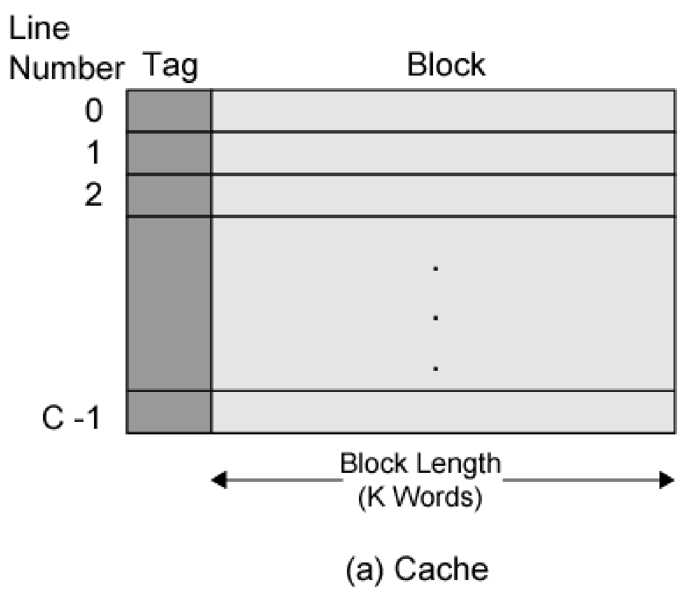
- Small amount of fast memory
- Sits between normal main memory and CPU
- May be located on CPU chip or module



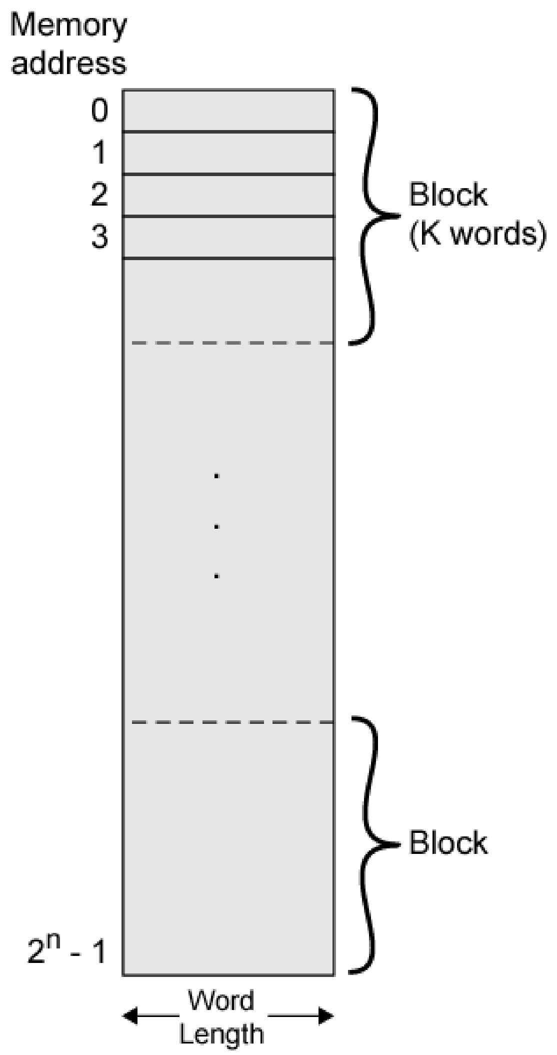
Cache

- Every address reference goes first to the cache;
  - if the desired address is not here, then we have a **cache miss**;
    - The contents are fetched from main memory into the indicated CPU register and the content is also saved into the cache memory
  - If the desired data is in the cache, then we have a **cache hit**
    - The desired data is brought from the cache, at very high speed (low access time)
- Most software exhibits **temporal locality of access**, meaning that it is likely that same address will be used again soon, and if so, the address will be found in the cache
- Transfers between main memory and cache occur at granularity of cache lines or cache blocks, around 32 or 64 bytes (rather than bytes or processor words).

Cache/Main Memory Structure



(a) Cache



(b) Main memory

M* = *2*^{n} */ K blocks in RAM

C << M blocks in cache

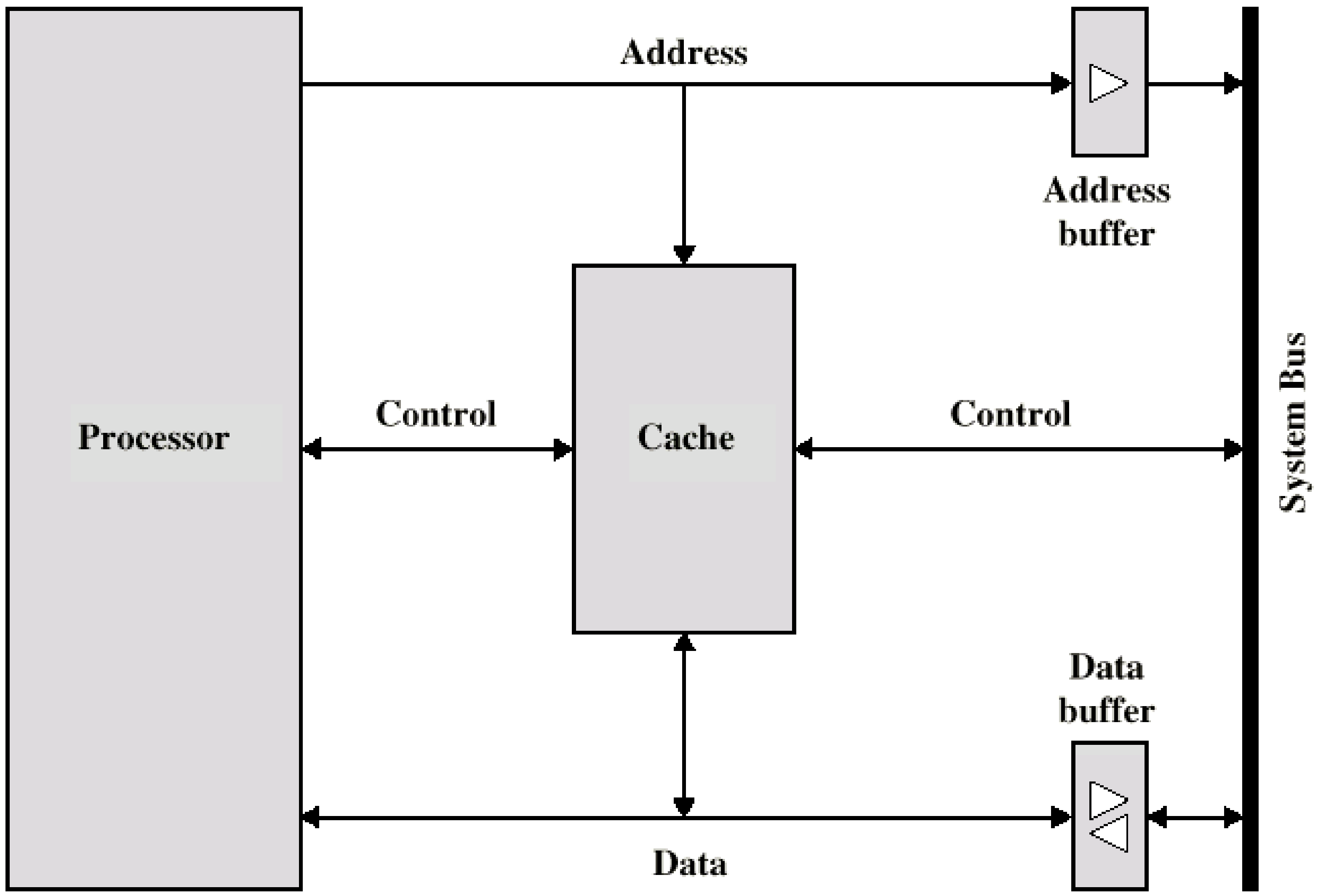
M, numober of blocks in ram

C, number of lines in cache

Cache operation – overview

- CPU requests contents of memory location
- Check cache for this data
- If present, get from cache (fast) *(cache hit)*
- If not present, read required block from main memory to cache *(cache miss)*
- Then deliver from cache to CPU
- Cache includes tags to identify which block of main memory is in each cache slot

Typical Cache Organization



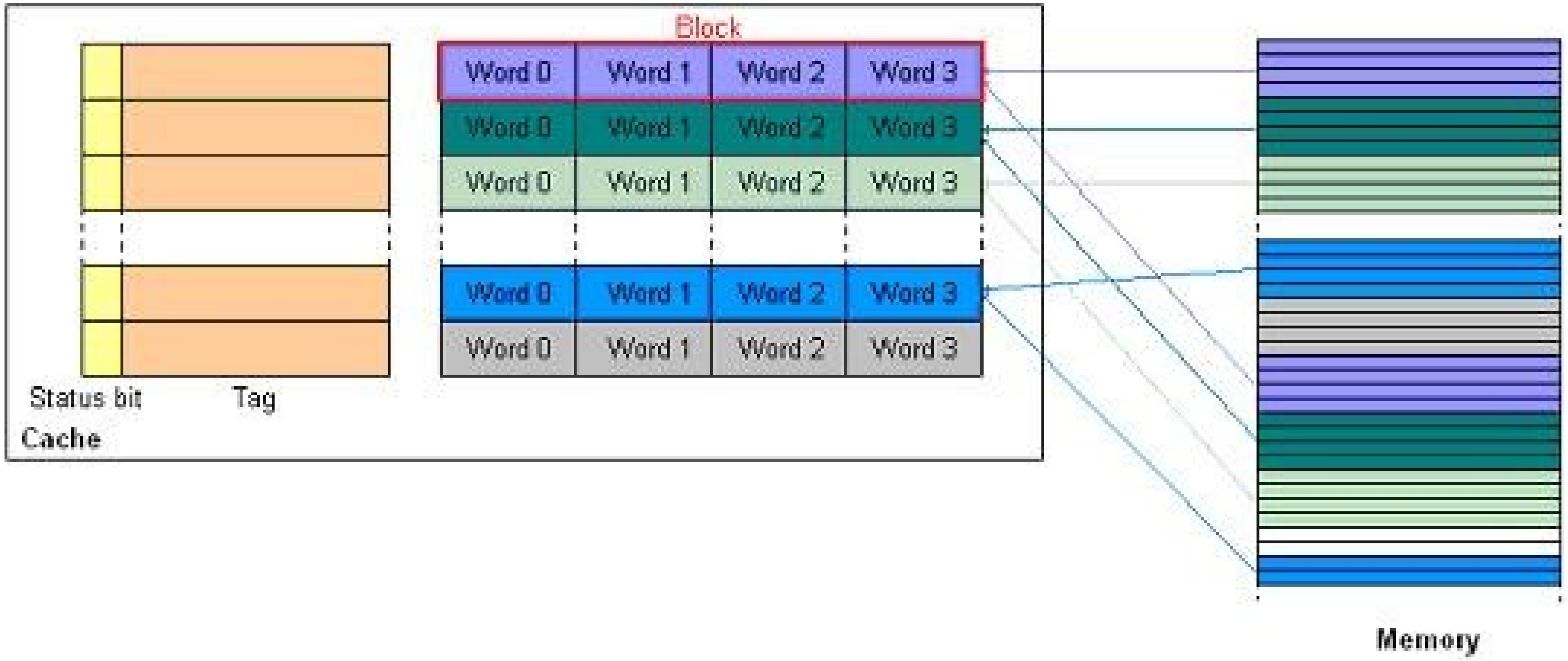
Where can a block be placed in Cache?

- Direct mapped Cache
  - Each block has only one place where it can appear in the cache
  - (Block Address) MOD (Number of blocks in cache)
- Fully associative Cache
  - A block can be placed anywhere in the cache
- Set associative Cache
  - A block can be placed in a restricted set of places into the cache
  - A set is a group of blocks in the cache
  - (Block Address) MOD (Number of sets in the cache)

Direct Mapping

- Each block of main memory maps to only one cache line
- Address is in two parts
- Least Significant *w* bits identify unique word
- Most Significant *s* bits specify one memory block
- The MSBs are split into a cache line field r and a tag of s-r (most significant)
- Mapping Function: ***i = j mod m***
    - Where
        - i = cache line, j = block number
        - m =number of lines in cache

Direct Mapping



Next:

Peripherals