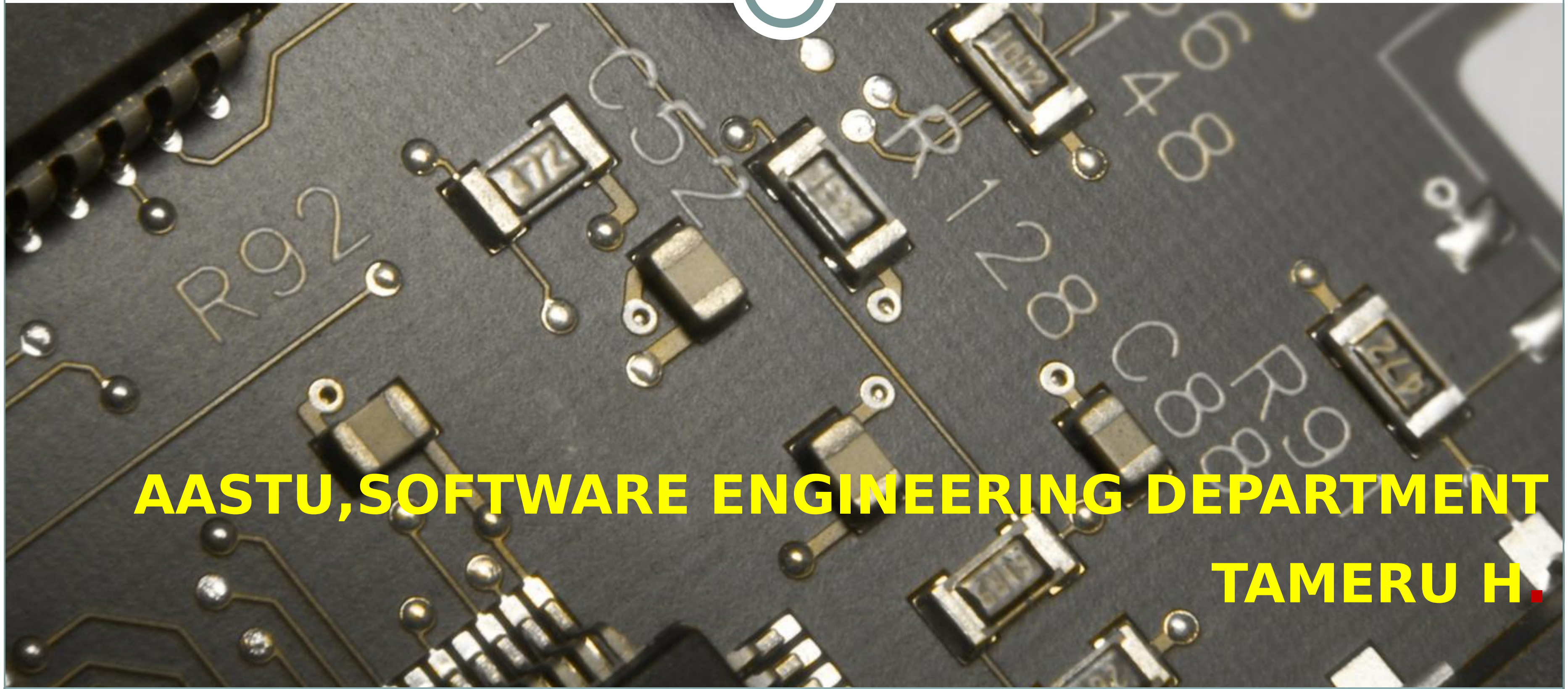
Chapter 4

Cache Memory



AASTU,SOFTWARE ENGINEERING DEPARTMENT

TAMERU H.

Memory Characteristics

- Location
- Capacity
- Unit of transfer
- Access method
- Performance
- Physical type
- Physical characteristics
- Organisation

Location

- CPU
- Internal
- External

Capacity

- Word size
  - The natural unit of organisation
- Number of words
  - or Bytes

Unit of Transfer

- Internal
  - Usually governed by data bus width
- External
  - Usually a block which is much larger than a word
- Addressable unit
  - Smallest location which can be uniquely addressed
  - Word internally

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 **vicinity** plus sequential search
  - Access time depends on location and previous location
  - **e.g. disk**

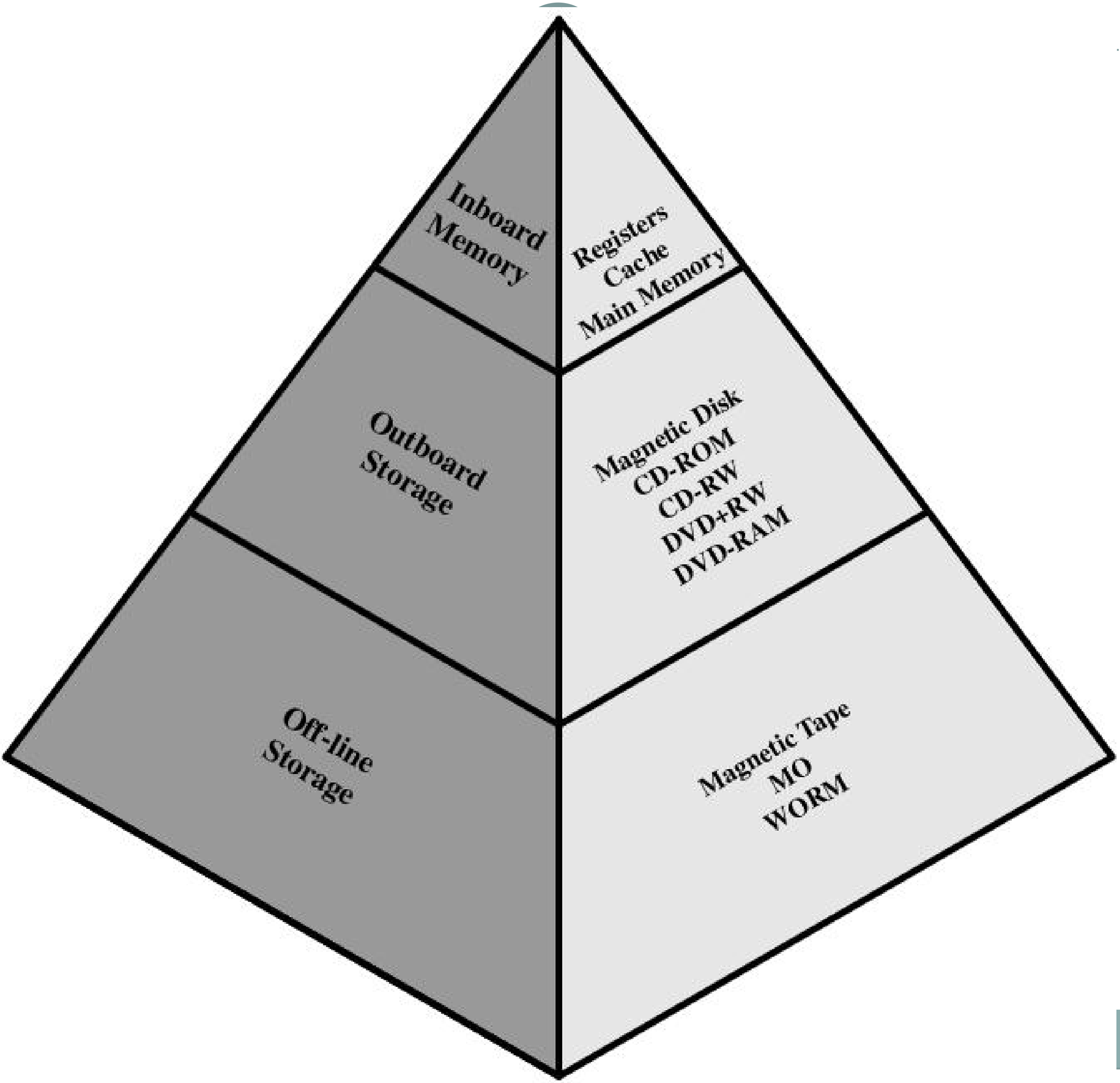
Cont'd…

- Random
  - Individual addresses identify locations exactly
  - Access time is independent of location or previous access
  - **e.g. RAM**
- Associative
  - Data is located by a comparison with contents of a portion of the store
  - Access time is independent of location or previous access
  - **e.g. cache**

Memory Hierarchy

- Registers
  - In CPU
- Internal or Main memory
  - May include one or more levels of cache
  - “RAM”
- External memory
  - Backing store

Memory Hierarchy - Diagram



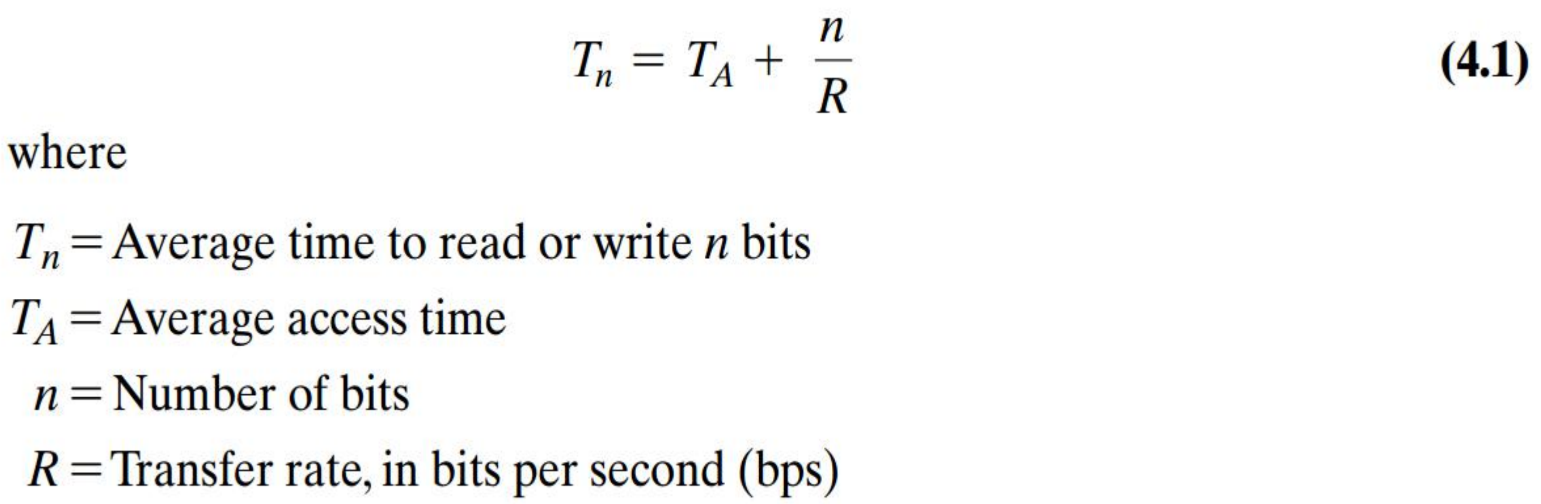
Performance

- Access time
  - Time between presenting the address and getting the valid data
- Memory Cycle time
  - Time may be required for the memory to “recover” before next access
  - Cycle time is access + recovery
- Transfer Rate
  - Rate at which data can be moved

Cont’d…

- For random-access memory, it is equal to **1/(cycle time).**

For non-random-access memory, the following relationship holds:



$$T_n = T_A + \frac{n}{R} \qquad \textbf{(4.1)}$$

where

T_n = Average time to read or write n bits

T_A = Average access time

n = Number of bits

R = Transfer rate, in bits per second (bps)

Physical Types

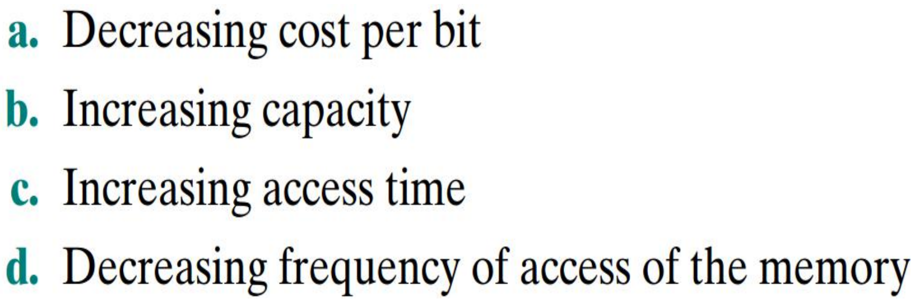
- Semiconductor
  - RAM
- Magnetic
  - Disk & Tape
- Optical
  - CD & DVD
- Others
  - Bubble
  - Hologram

The Bottom Line

- How much?
  - Capacity
- How fast?
  - Time is money
- How expensive?

Hierarchy List

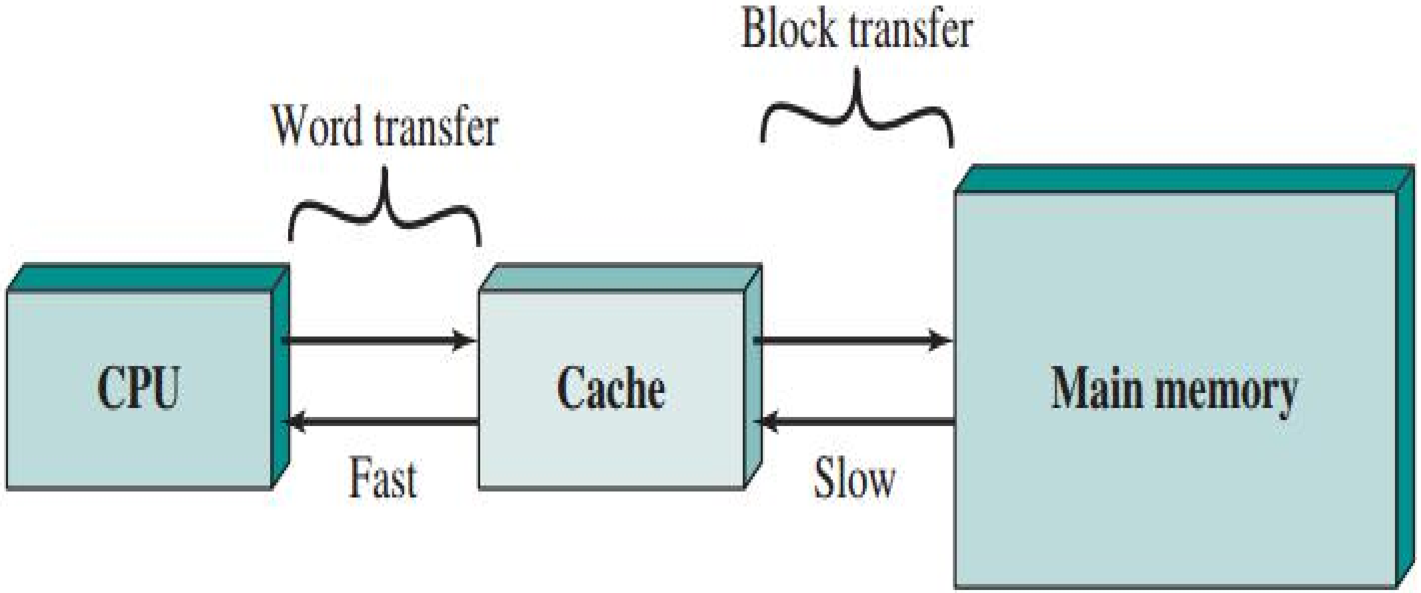
- Registers
- L1 Cache
- L2 Cache
- Main memory
- Disk cache
- Disk
- Optical
- Tape



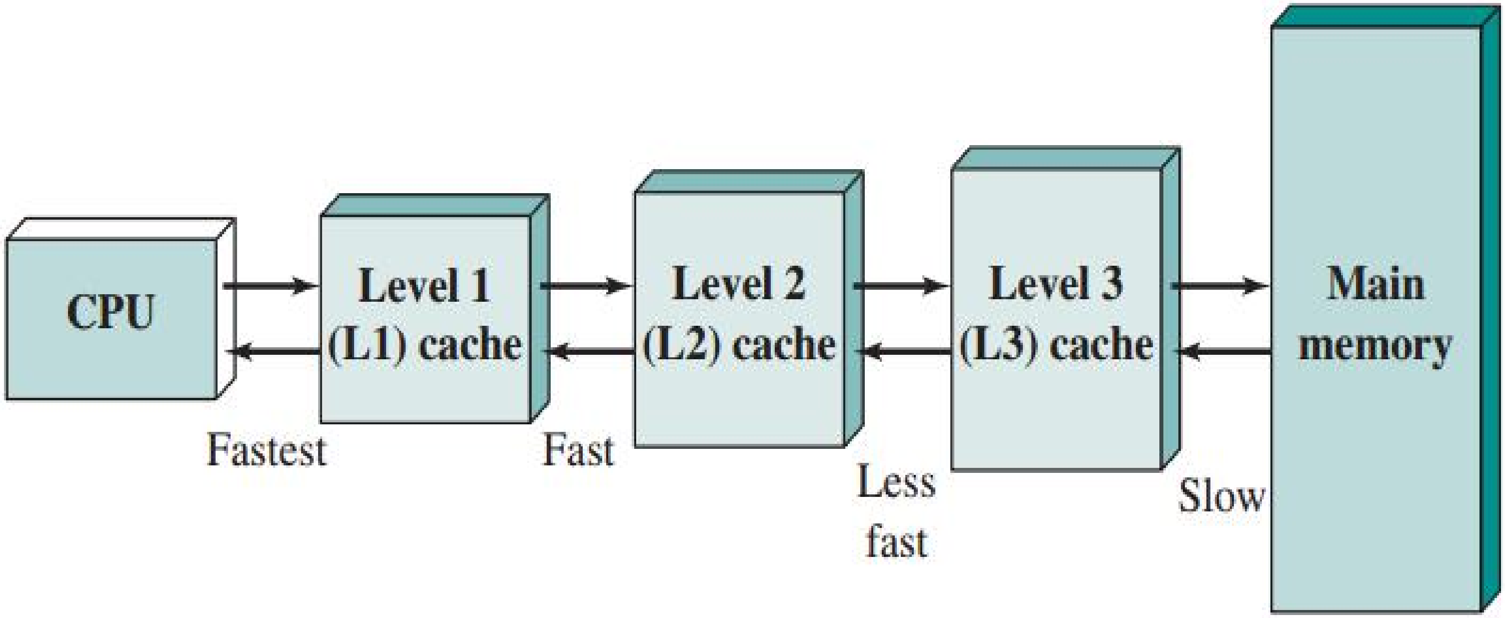
a. Decreasing cost per bit
b. Increasing capacity
c. Increasing access time
d. Decreasing frequency of access of the memory

Cache

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

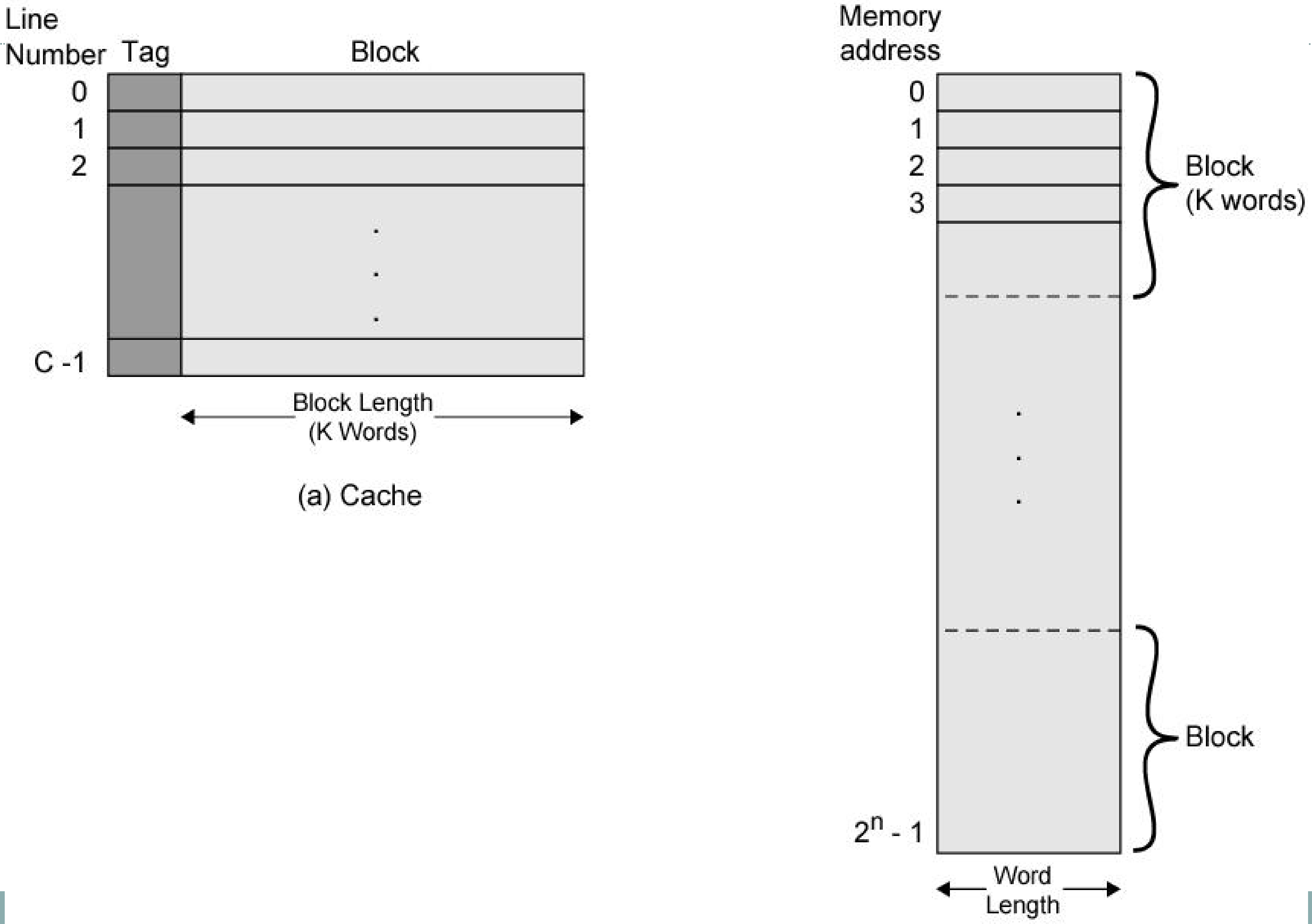


Cont'd…



(b) Three-level cache organization

Cache/Main Memory Structure



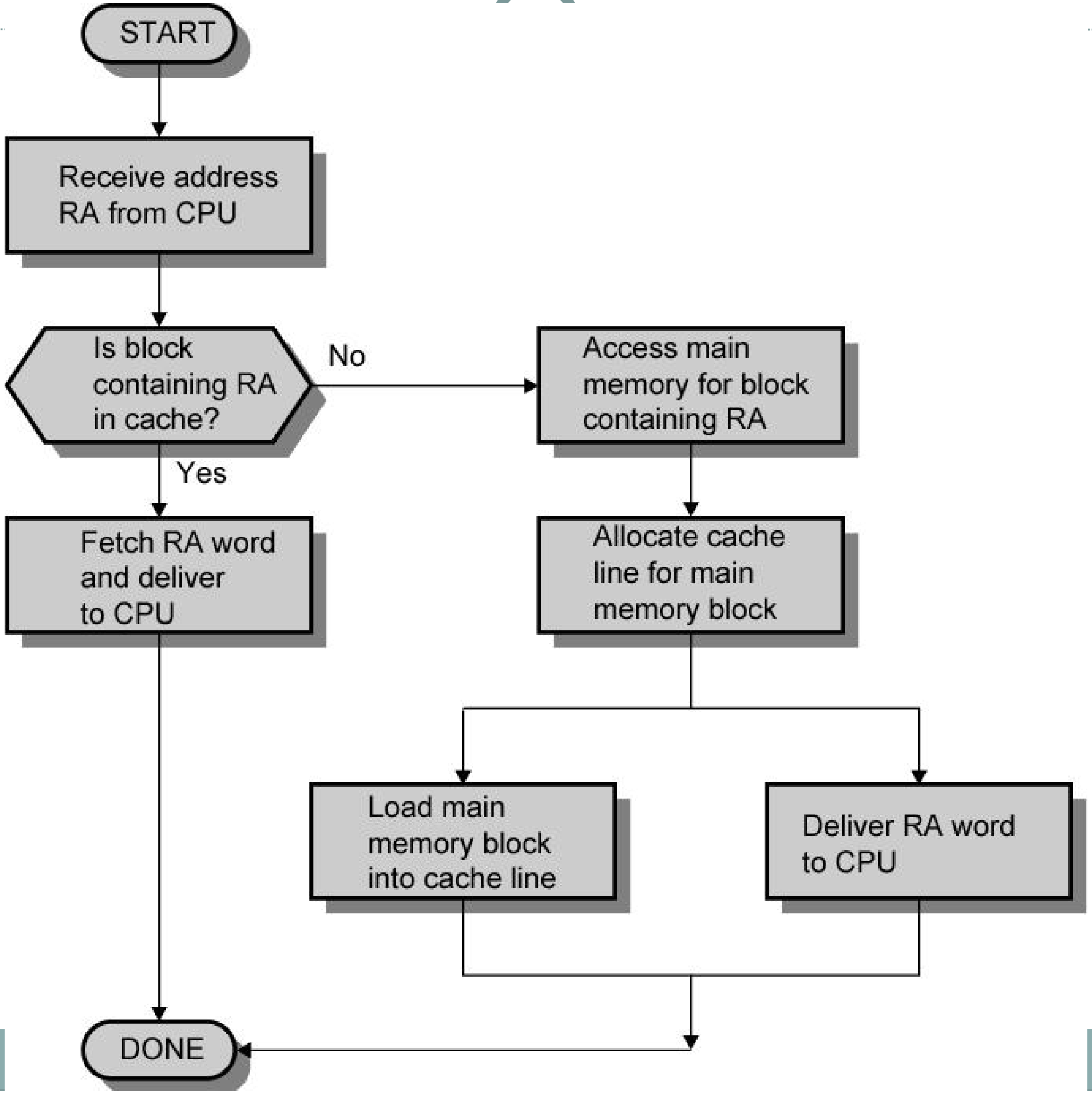
(a) Cache

(b) Main memory

Cache operation

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

Cache Read Operation - Flowchart


START
Receive address RA from CPU
Is block containing RA in cache?
No
Yes
Access main memory for block containing RA
Fetch RA word and deliver to CPU
Allocate cache line for main memory block
Load main memory block into cache line
Deliver RA word to CPU
DONE

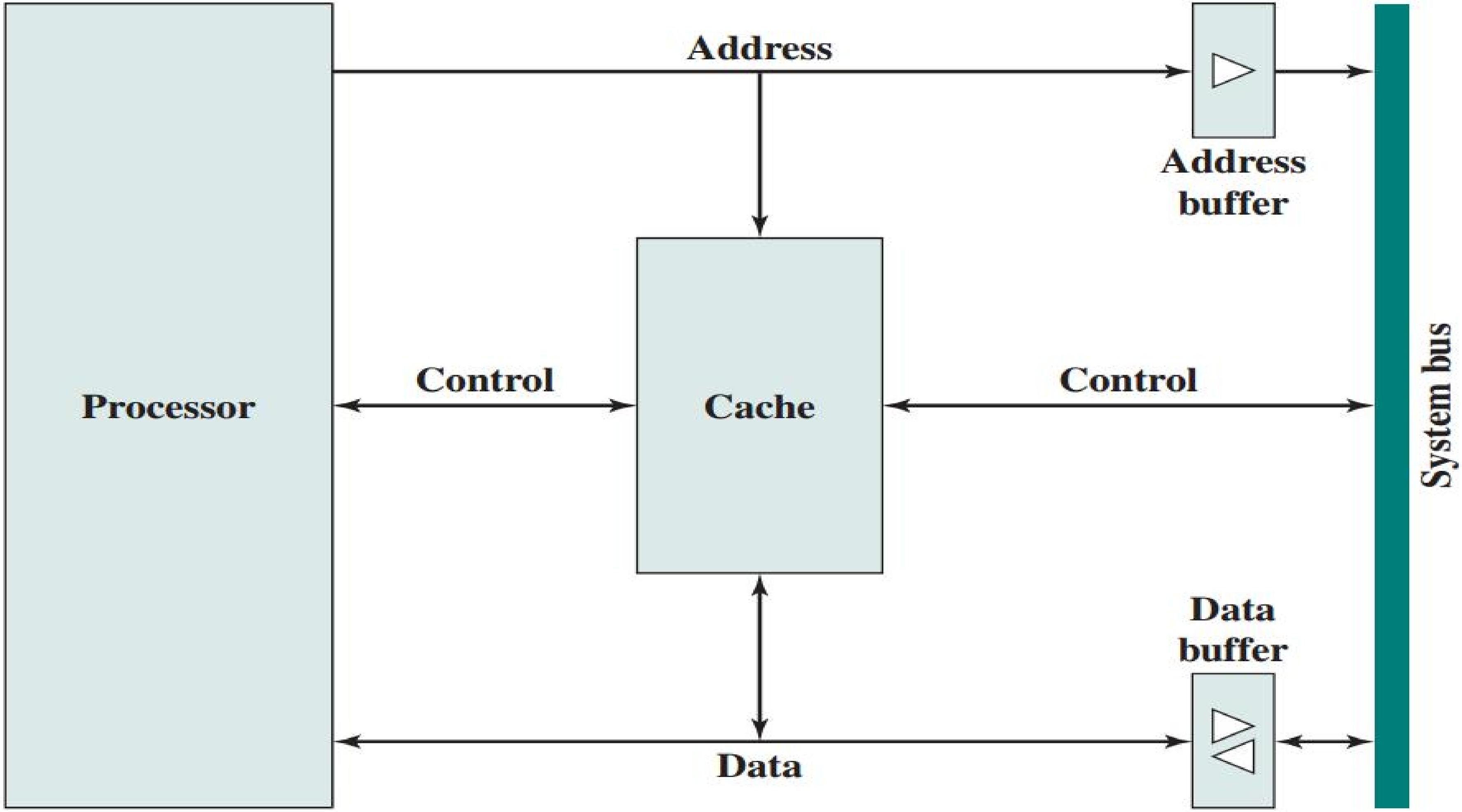
Cache Design

- Size
- Mapping Function
- Replacement Algorithm
- Write Policy
- Block Size
- Number of Caches

Size does matter

- Cost
  - More cache is expensive
- Speed
  - More cache is faster (up to a point)
  - Checking cache for data takes time

Typical Cache Organization



Mapping Function

- Cache of 64kByte
- Cache block of 4 bytes
  - i.e. cache is 16k (2^{14}) lines of 4 bytes
- 16MBytes main memory
- 24 bit address
  - (2^{24}=16M)

Direct Mapping

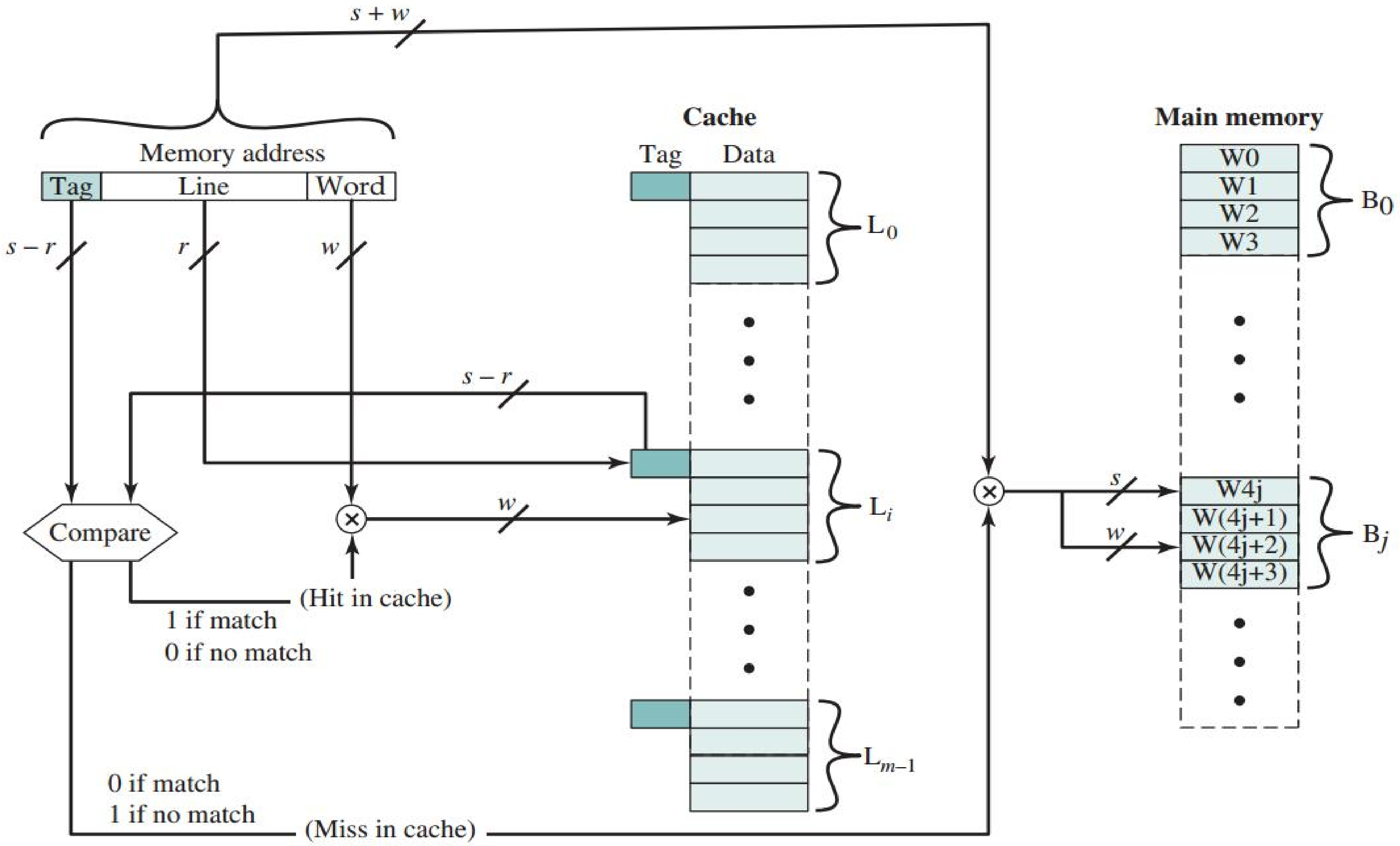
- Each block of main memory maps to only one cache line
  - i.e. if a block is in cache, it must be in one specific place
- Address is in two parts
- Least Significant **w** bits identify **unique word**
- Most Significant **s** bits specify one memory block
- The MSBs are split into a cache line field **r** and a tag of **s-r** (most significant)

Direct Mapping
Address Structure

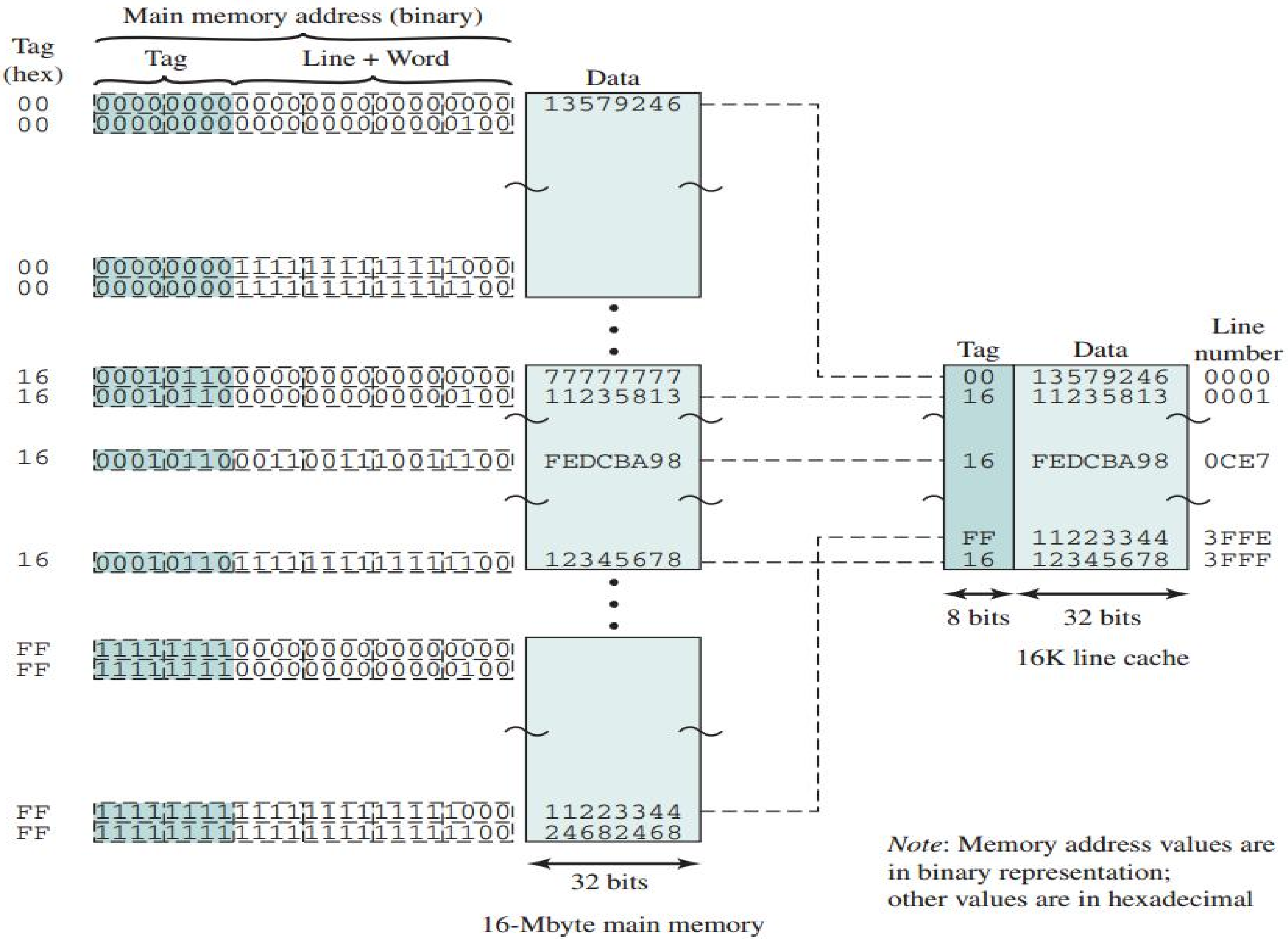
Tag s-r	Line or Slot r	Word w
8	14	2

- 24 bit address
- 2 bit word identifier (4 byte block)
- 22 bit block identifier
  - 8 bit tag (=22-14)
  - 14 bit slot or line
- No two blocks in the same line have the same Tag field
- Check contents of cache by finding line and checking Tag

Direct Mapping Cache Organization

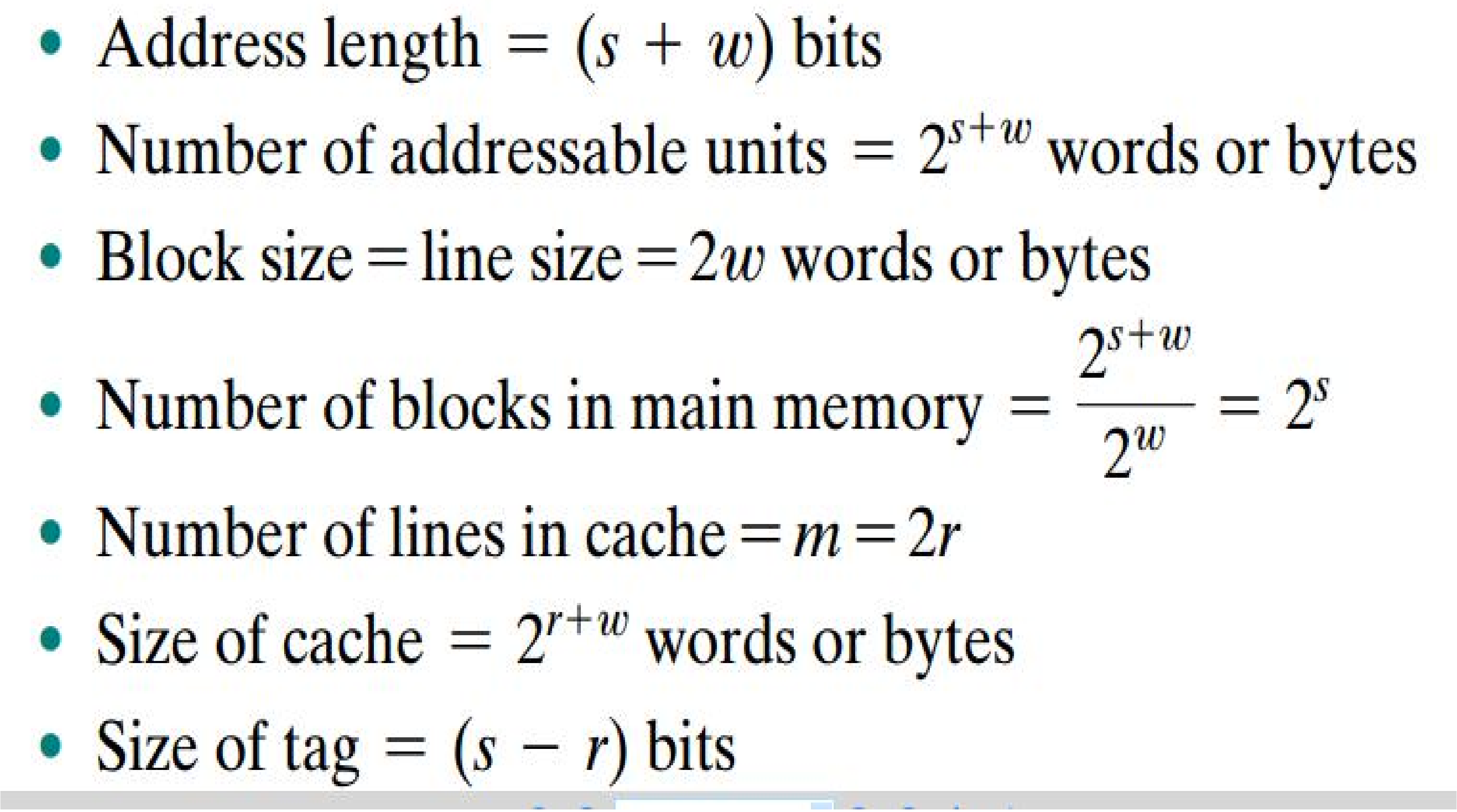


Direct Mapping Example



Note: Memory address values are in binary representation; other values are in hexadecimal

Direct Mapping Summary



- Address length $= (s + w)$ bits
- Number of addressable units $= 2^{s+w}$ words or bytes
- Block size = line size = $2w$ words or bytes
- Number of blocks in main memory $= \dfrac{2^{s+w}}{2^w} = 2^s$
- Number of lines in cache $= m = 2r$
- Size of cache $= 2^{r+w}$ words or bytes
- Size of tag $= (s - r)$ bits

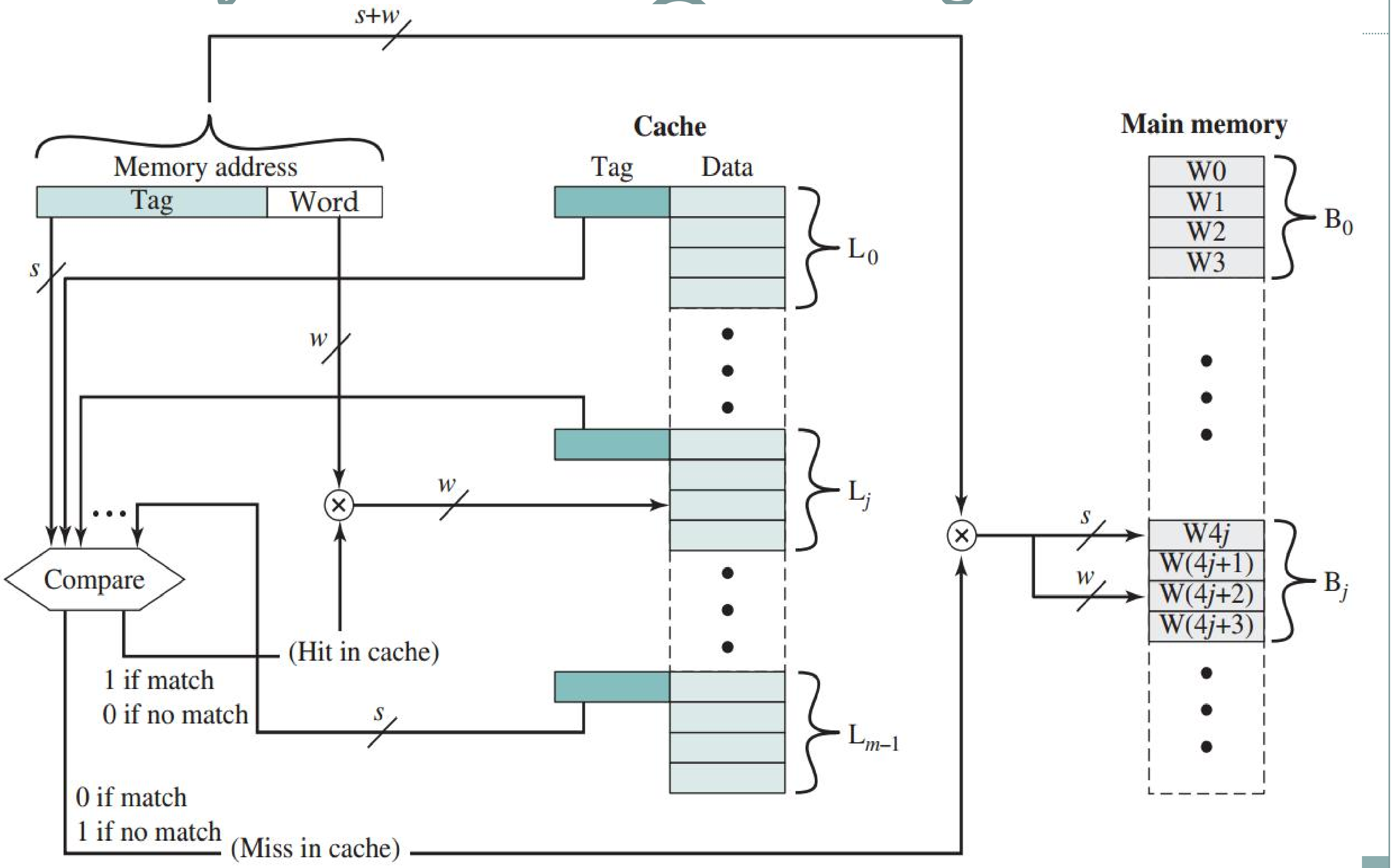
Direct Mapping pros & cons

- Simple
- Inexpensive
- Fixed location for given block
  - If a program accesses 2 blocks that map to the same line repeatedly, cache misses are very high

Associative Mapping

- A main memory block can load into **any line** of cache
- Memory address is interpreted **as tag and word**
- Tag uniquely identifies block of memory
- Every line’s tag is examined for a match
- Cache searching gets **expensive**

Fully Associative Cache Organization



Associative Mapping Summary

- Address length = $(s + w)$ bits
- Number of addressable units = 2^{s+w} words or bytes
- Block size = line size = $2w$ words or bytes
- Number of blocks in main memory = $\dfrac{2^{s+w}}{2^{w}} = 2^{s}$
- Number of lines in cache = undetermined
- Size of tag = s bits

Reading Assignment

- Set Associative mapping

Replacement Algorithms

Direct mapping

- No choice
- Each block only maps to one line
- Replace that line

Replacement Algorithms

Associative & Set Associative

- Hardware implemented algorithm **(speed)**
- Least Recently used (**LRU**)
- First in first out (**FIFO**)
  - replace block that has been in cache longest
- Least frequently used
  - replace block which has had **fewest hits**
- Random

Write Policy

- Must not overwrite a cache block **unless main memory is up to date**
- Multiple CPUs may have individual caches
- I/O may address main memory directly

Write through

- All writes go to **main memory as well as cache**
- Multiple CPUs can monitor main memory traffic to keep local (to CPU) cache up to date
- **Lots of traffic**
- Slows down writes

Write back

- Updates initially made in cache only
- **Update bit** for cache slot is set when update occurs
- If block is to be replaced, write to main memory only if update bit is set
- Other caches get out of sync
- I/O must access main memory through cache