Tishk International University
Mechatronics Engineering Department
Computer Architecture ME228 /A

Computer Architecture: Cache Memory
Chapter NO: 4

Characteristics

1. **Location** (internal or external)
   - Internal: cache memory
   - CPU register
   - External: peripheral storage device
2. **Capacity**: in terms of bytes or words

Cache memory size is often restricted to **between 8 KB and 64 KB**

WORD (16 bits/**2 bytes**)

3. **Unit of transfer**
   - For internal memory: number of electrical lines into/out the memory (64, 128, 256 bits)
4. **Access method**
   - Sequential access
   - Direct access
   - Random Access (through address)
   - Associative: make a comparison of desired bit locations within a word for a specified match, and to do this for all words simultaneously.

Characteristics

5. Performance:

- **Access Time**
  - RAM: time to perform a read or write operation, from the address is presented to the memory to the data have been stored or made available for use
  - Other memory: time to read–write mechanism at the desired location
- **Memory Cycle Time:** the access time + any additional time required before a second access (the additional time may be required for transients)
- **Transfer Time**: the rate at which data can be transferred into or out of a memory unit.

Characteristics

6. Physical type: such as semiconductor memory (RAM), magnetic surface memory used for disk and tape, and optical and magneto-optical (CD and DVD)

7. Physical characteristics: Volatility, erasable

- Volatile: such as semiconductor memory
- Non Volatile: such as Magnetic-surface memories and semiconductor memory.
- Non Erasable memory part of non-volatile such as Read-only-Memory (ROM)
- Power consumption

8. Organisation: physical arrangement of bits to form words

Location

- Internal (e.g., processor registers, cache, main memory)
- External (e.g., optical disks, magnetic disks, tapes)

Capacity

- Number of words
- Number of bytes

Unit of Transfer

- Word
- Block

Access Method

- Sequential
- Direct
- Random
- Associative

Performance

- Access time
- Cycle time
- Transfer rate

Physical Type

- Semiconductor
- Magnetic
- Optical
- Magneto-optical

Physical Characteristics

- Volatile/nonvolatile
- Erasable/nonerasable

Organization

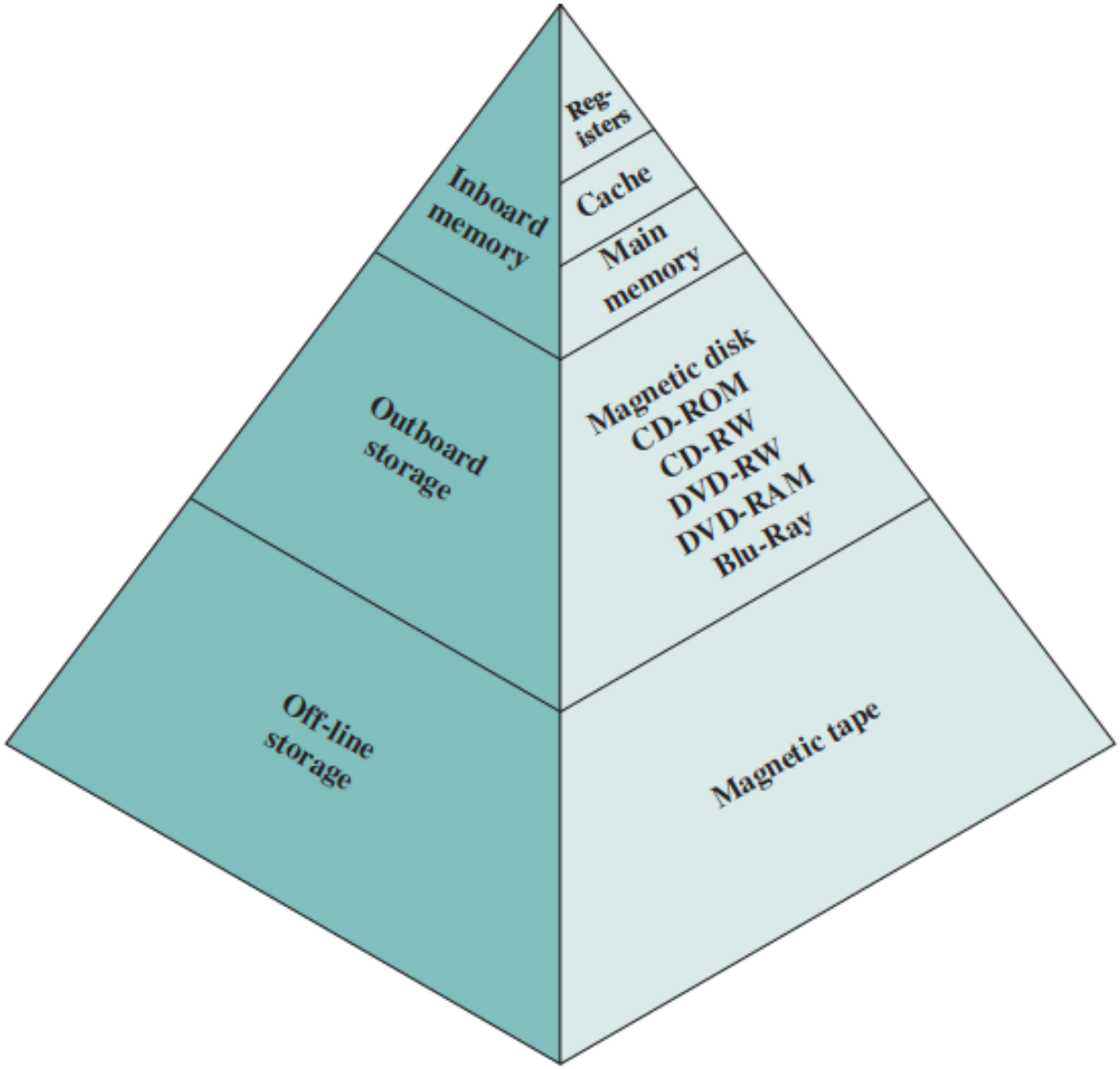
- Memory modules

Memory Hierarchy

- To design computer's memory, three key characteristics: **capacity, speed, and cost**.
  - Capacity: the memory storage must match the application requirements
  - Speed: memory speed must match processor speed.
  - Cost: must be reasonable compared to the other components

Memory Hierarchy - Diagram

Figure 1



Memory Hierarchy - Diagram

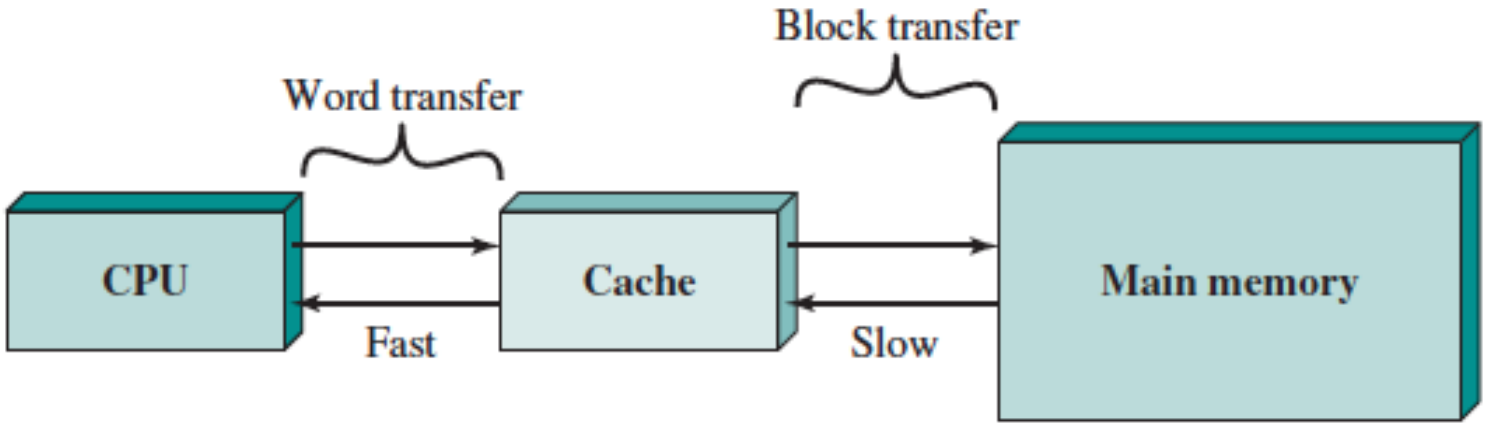
The way out of this dilemma is not to rely on a single memory component or technology, but to employ a memory hierarchy. A typical hierarchy is illustrated in Figure 1. As one goes down the hierarchy, the following occur:

1. Decreasing cost per bit
2. Increasing capacity
3. Increasing access time
4. Decreasing frequency of access of the memory by the processor.

Smaller, more expensive, faster memories are supplemented by larger, cheaper, slower memories.

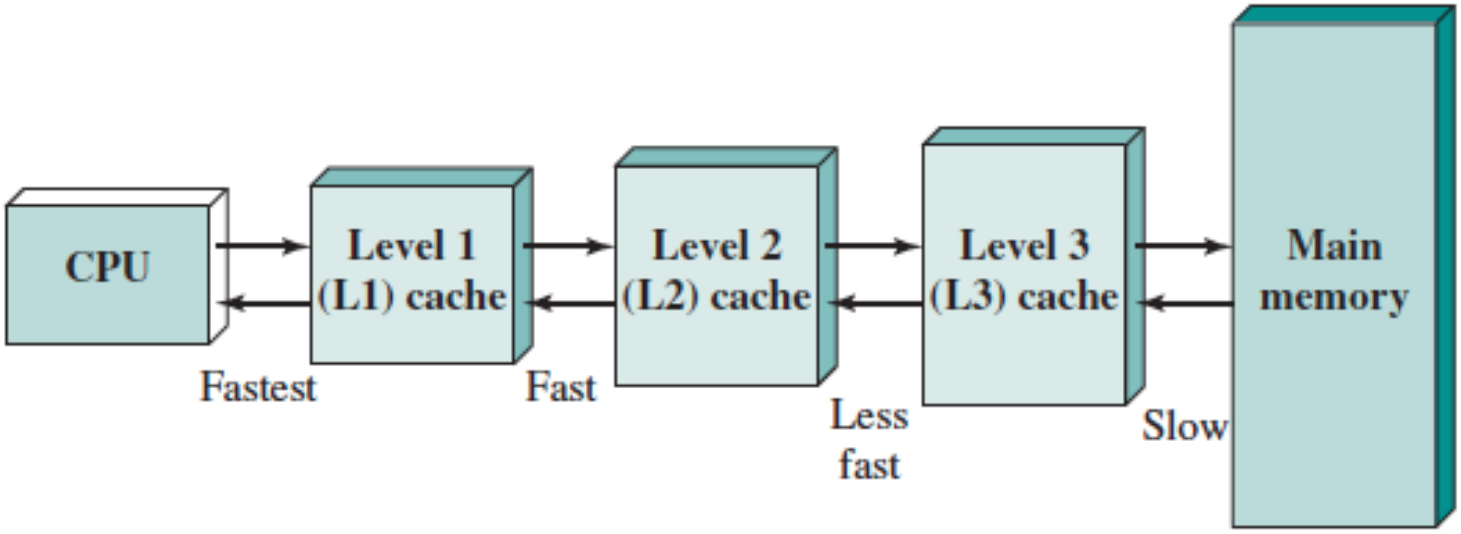
Cache Memory

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

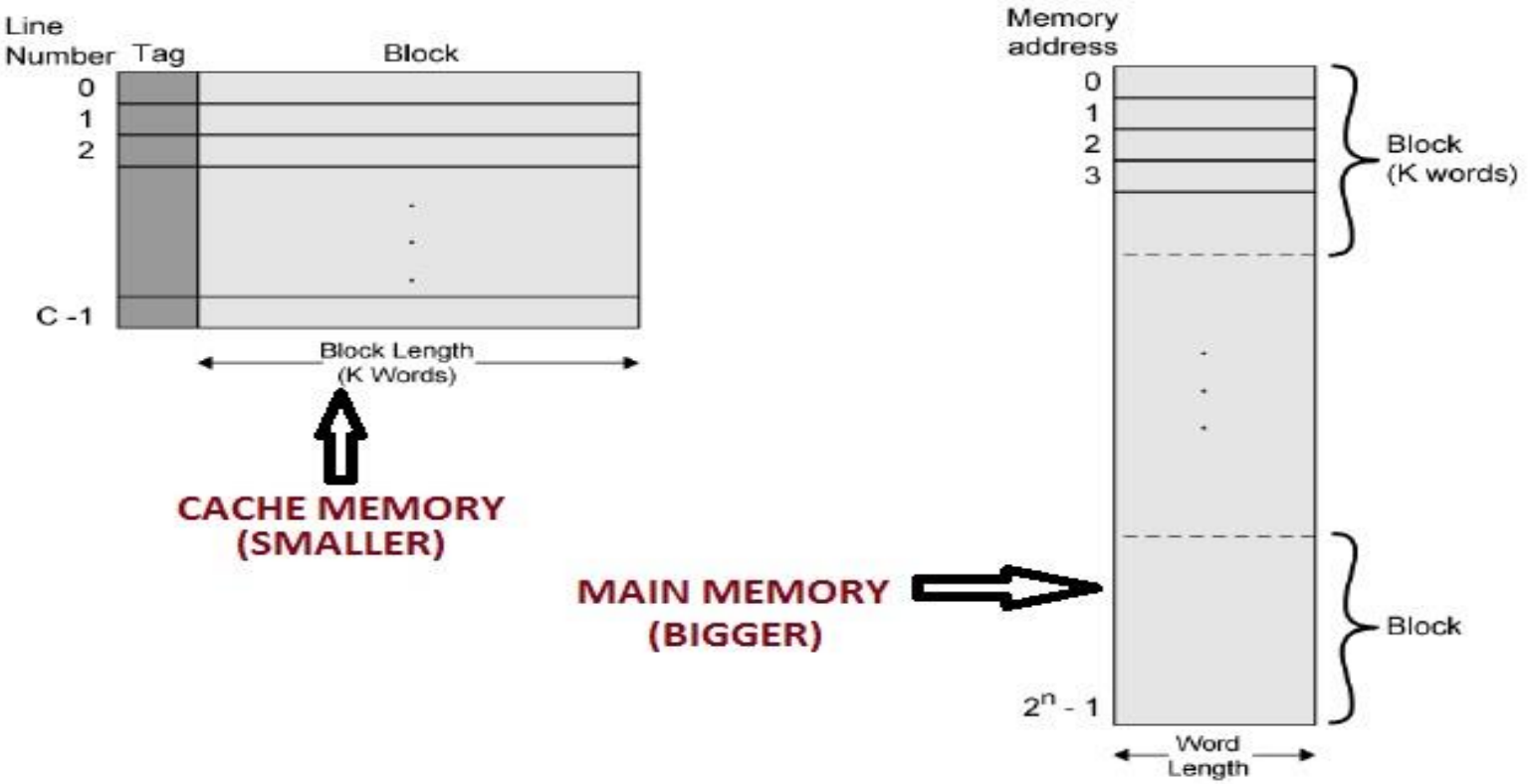


Cache Memory

Figure below depicts the use of multiple levels of cache. The L2 cache is slower and typically larger than the L1 cache, and the L3 cache is slower and typically larger than the L2 cache.



Cache/Main Memory Structure



i) Main Memory has n address bits - 2^n words of memory

ii) Cache has fixed length blocks (C blocks) of K words, where $2^n > K$

iii) From the cache viewpoint, memory has M blocks where $M = 2^n / K$

iv) One block if cache memory is called a line and K is the line size

v) No. of Main Memory Blocks = M

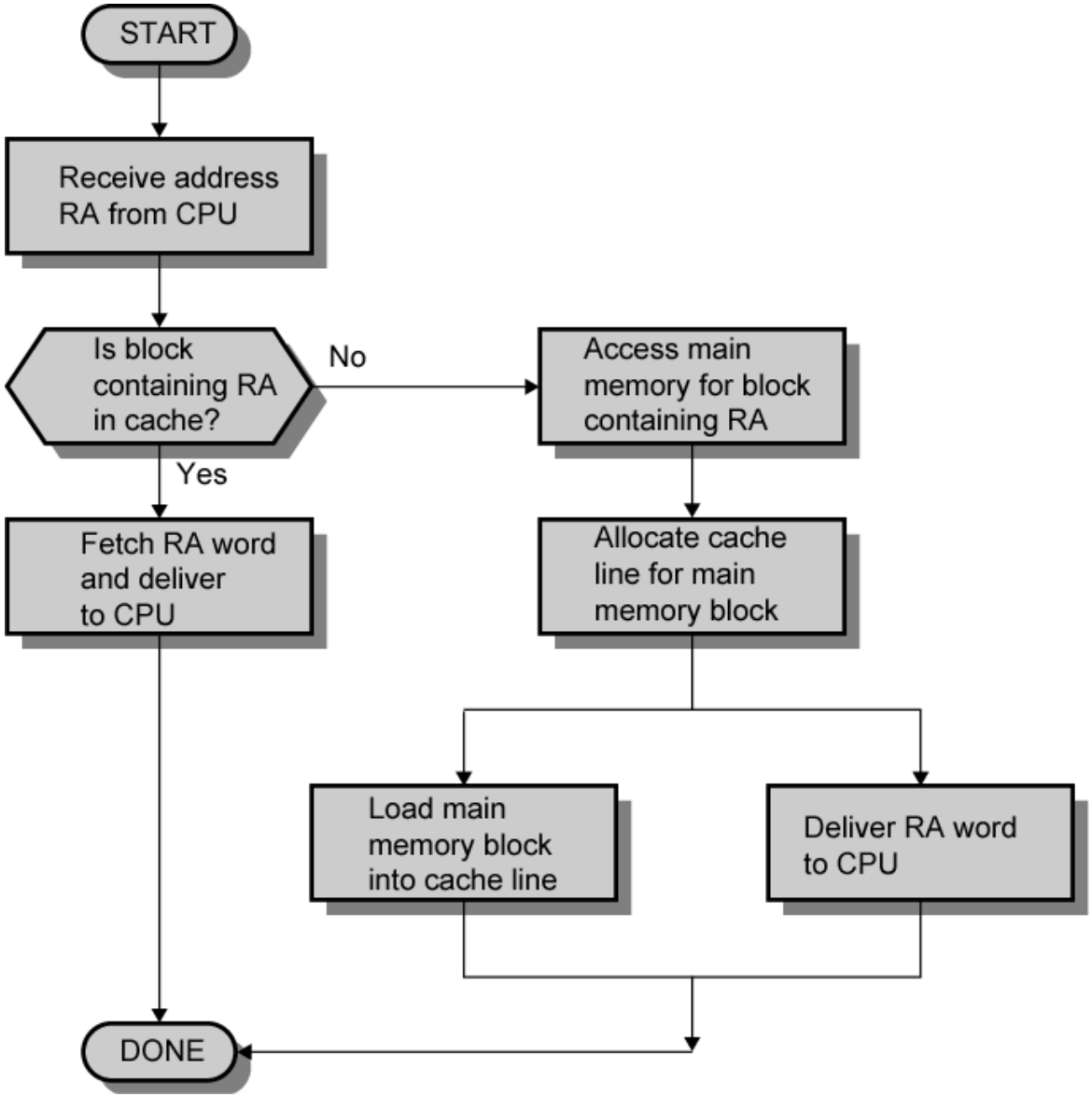
No. of Cache Memory Blocks = C

$C << M$

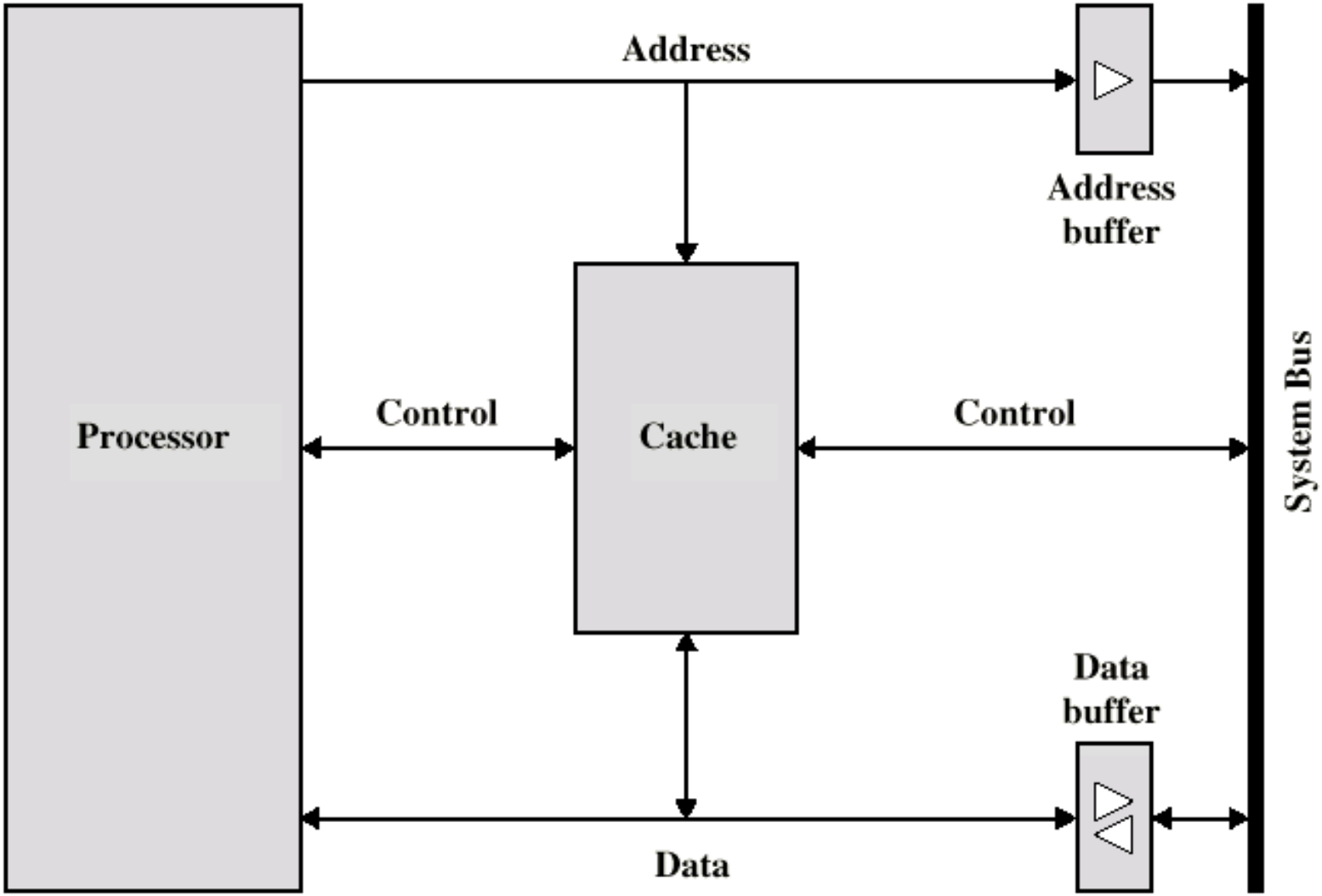
Cache operation – overview

- 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



Typical Cache Organization



Cache Design

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

Cache Addresses

- Logical
- Physical

Cache Size

Mapping Function

- Direct
- Associative
- Set associative

Replacement Algorithm

- Least recently used (LRU)
- First in first out (FIFO)
- Least frequently used (LFU)
- Random

Write Policy

- Write through
- Write back

Line Size

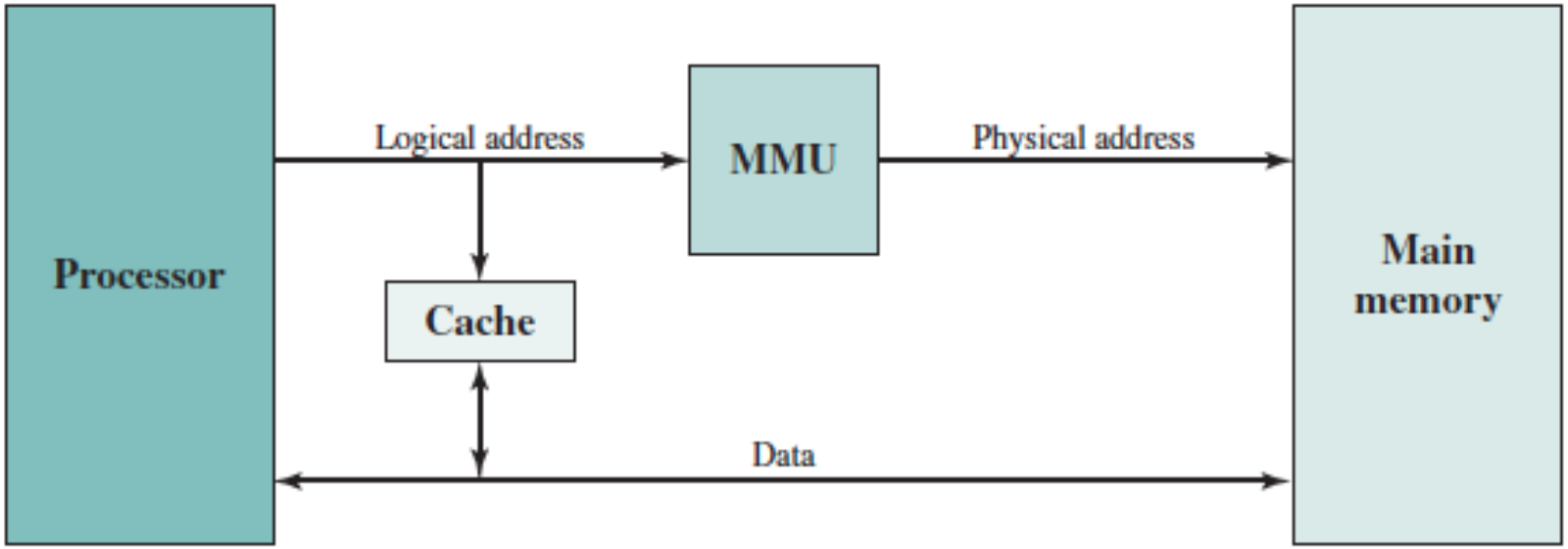
Number of Caches

- Single or two level
- Unified or split

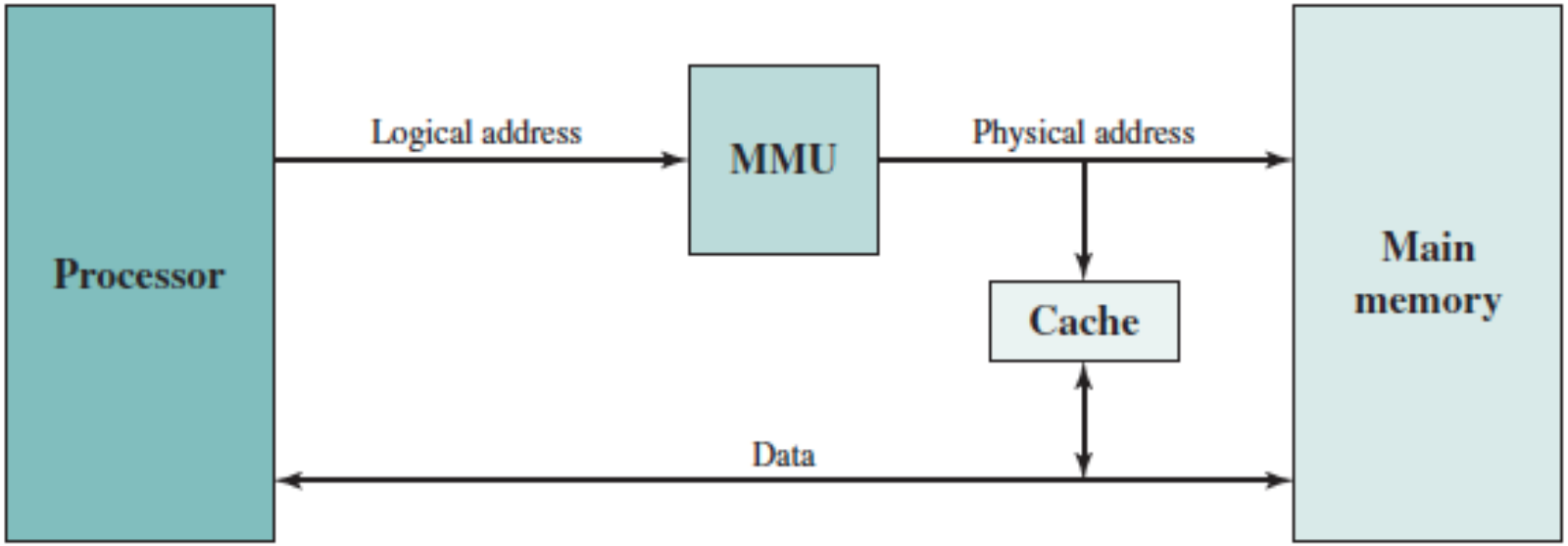
Cache Address

- For reads to and writes from main memory, a hardware memory management unit (MMU) translates each virtual address into a physical address in main memory.
- When virtual addresses are used, the system designer may choose to place the cache between the processor and the MMU or between the MMU and main memory.
- **A logical cache** (virtual cache) stores data using virtual addresses. The processor accesses the cache directly, without going through the MMU. A physical cache stores data using main memory physical addresses.
- **Advantage**:
  - Cache access speed is faster than for a physical cache, because the cache can respond before the MMU performs an address translation.
- **Disadvantage**:
  - Most virtual memory systems supply each application with the same virtual memory address space. Each application sees a virtual memory that starts at address 0. Thus, the same virtual address in two different applications refers to two different physical addresses. The cache memory must therefore be completely flushed with each application context switch, or extra bits must be added to each line of the cache to identify which virtual address space this address refers to.

Cache Address



(a) Logical cache



(b) Physical cache

Comparison of Cache Sizes

Processor	Type	Year of Introduction	L1 Cache[a]	L2 Cache	L3 Cache
IBM 360/85	Mainframe	1968	16–32 kB	—	—
PDP-11/70	Minicomputer	1975	1 kB	—	—
VAX 11/780	Minicomputer	1978	16 kB	—	—
IBM 3033	Mainframe	1978	64 kB	—	—
IBM 3090	Mainframe	1985	128–256 kB	—	—
Intel 80486	PC	1989	8 kB	—	—
Pentium	PC	1993	8 kB/8 kB	256–512 kB	—
PowerPC 601	PC	1993	32 kB	—	—
PowerPC 620	PC	1996	32 kB/32 kB	—	—
PowerPC G4	PC/server	1999	32 kB/32 kB	256 kB to 1 MB	2 MB
IBM S/390 G6	Mainframe	1999	256 kB	8 MB	—
Pentium 4	PC/server	2000	8 kB/8 kB	256 kB	—
IBM SP	High-end server/ supercomputer	2000	64 kB/32 kB	8 MB	—
CRAY MTA[b]	Supercomputer	2000	8 kB	2 MB	—
Itanium	PC/server	2001	16 kB/16 kB	96 kB	4 MB
Itanium 2	PC/server	2002	32 kB	256 kB	6 MB
IBM POWER5	High-end server	2003	64 kB	1.9 MB	36 MB
CRAY XD-1	Supercomputer	2004	64 kB/64 kB	1 MB	—
IBM POWER6	PC/server	2007	64 kB/64 kB	4 MB	32 MB
IBM z10	Mainframe	2008	64 kB/128 kB	3 MB	24–48 MB
Intel Core i7 EE 990	Workstation/ server	2011	6 × 32 kB/ 32 kB	1.5 MB	12 MB
IBM zEnterprise 196	Mainframe/ server	2011	24 × 64 kB/ 128 kB	24 × 1.5 MB	24 MB L3 192 MB L4

Size does matter

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

Mapping Function

- There are fewer cache lines than main memory blocks, an algorithm is needed for mapping main memory blocks into cache lines.
- Need to determine which main memory block currently occupies a cache line.
- The choice of the mapping function dictates how the cache is organized.
- Three techniques for mapping function:
  - Direct
  - Associative
  - Set associative.

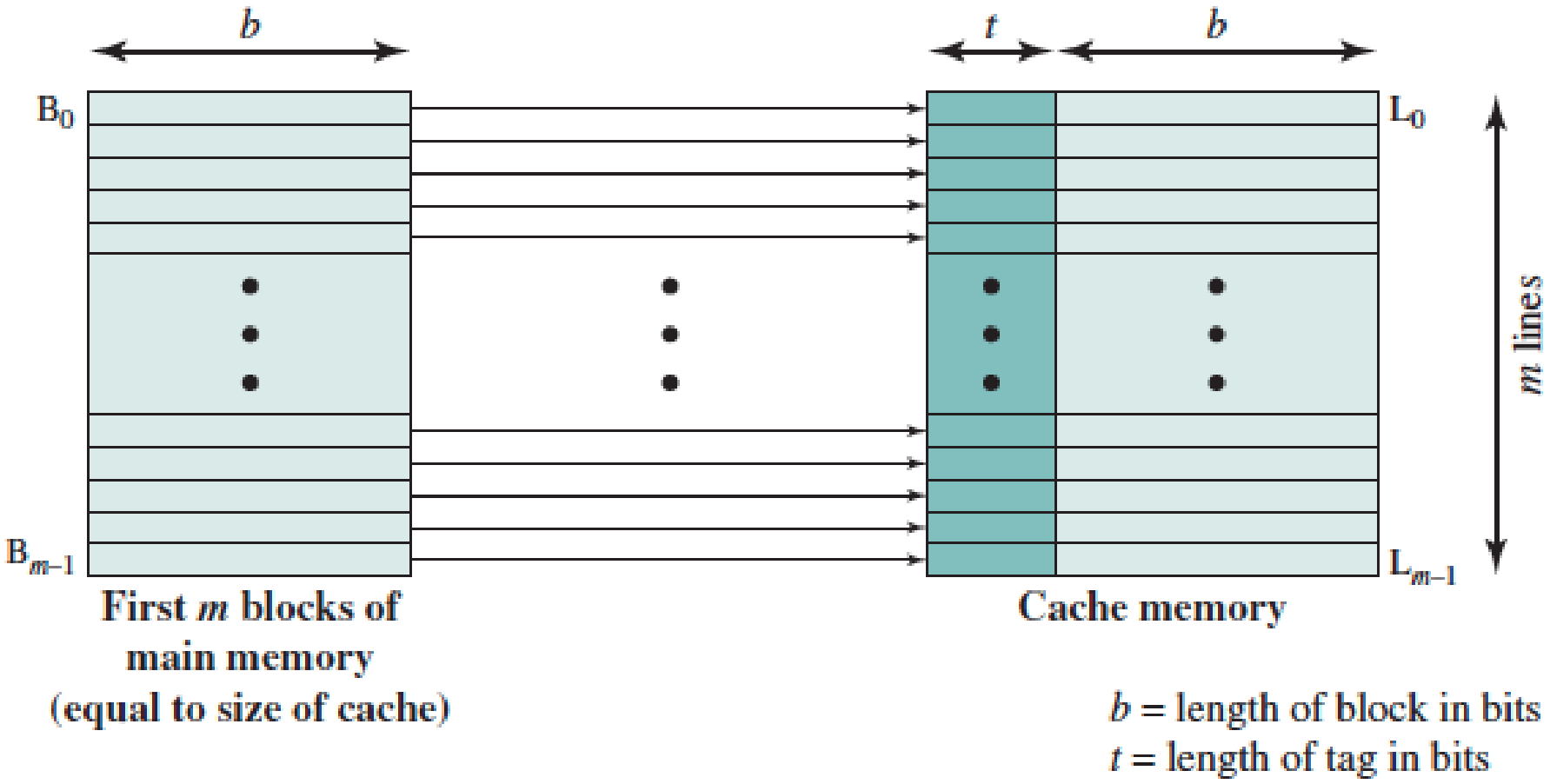
Mapping Function

- Cache of 64 kByte
- Cache block of 4 bytes
  - i.e. cache is 16k (2^{14}) lines of 4 bytes
- 16 MBytes 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)

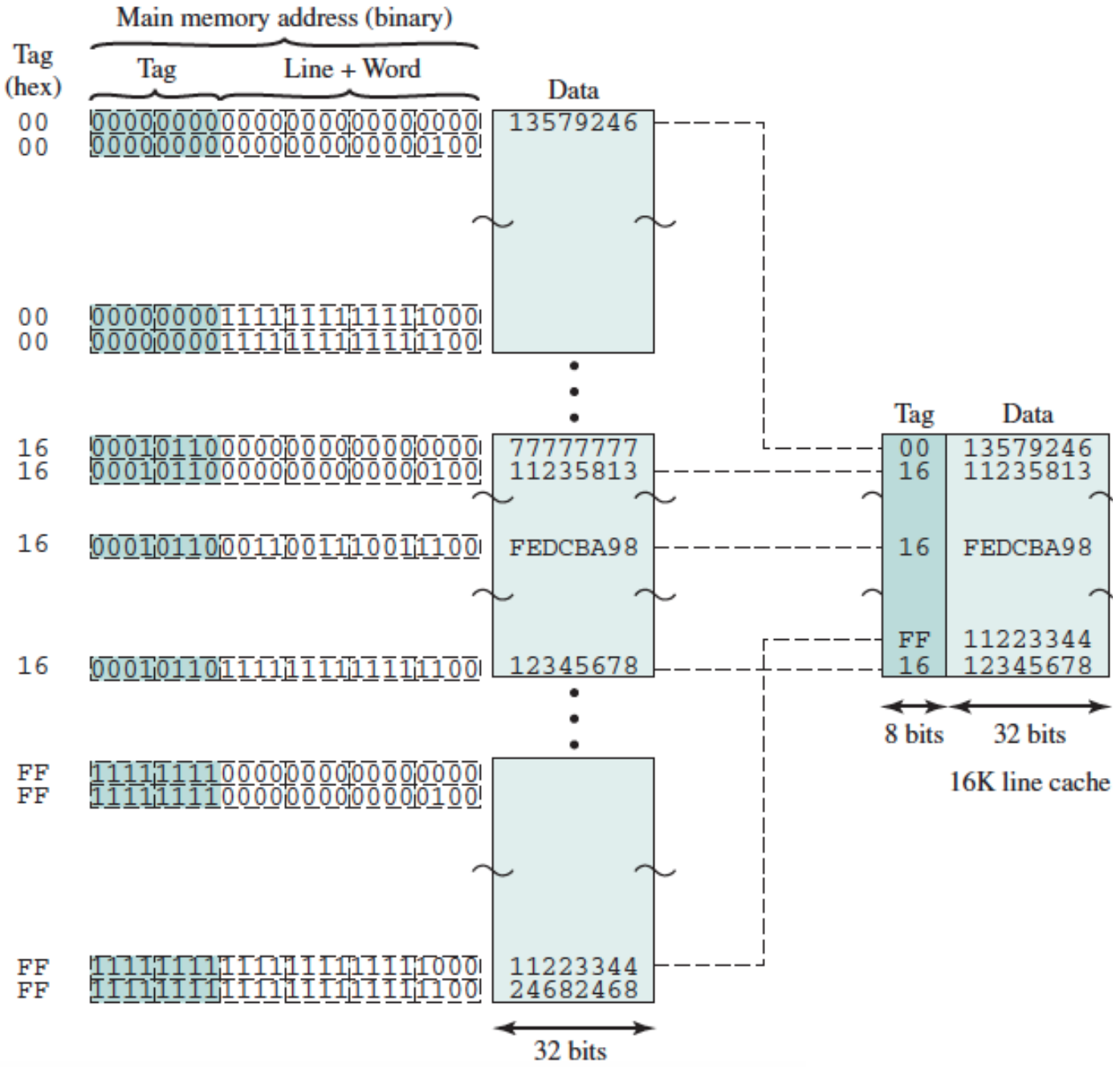
Direct Mapping



Direct Mapping: Address Structure

- 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

Example


Main memory address (binary)
Tag
(hex)
Tag
Line + Word
Data
00
00
00000000000000000000000000
00000000000000000000000100
13579246
00
00
00000000111111111111111000
00000000111111111111111100
16
16
00010110000000000000000000
00010110000000000000000100
77777777
11235813
16
00010110011001110011001100
FEDCBA98
16
00010110111111111111111100
12345678
FF
FF
11111111000000000000000000
11111111000000000000000100
FF
FF
11111111111111111111111000
11111111111111111111111100
11223344
24682468
32 bits
Tag
Data
00
16
13579246
11235813
16
FEDCBA98
FF
16
11223344
12345678
8 bits
32 bits
16K line cache

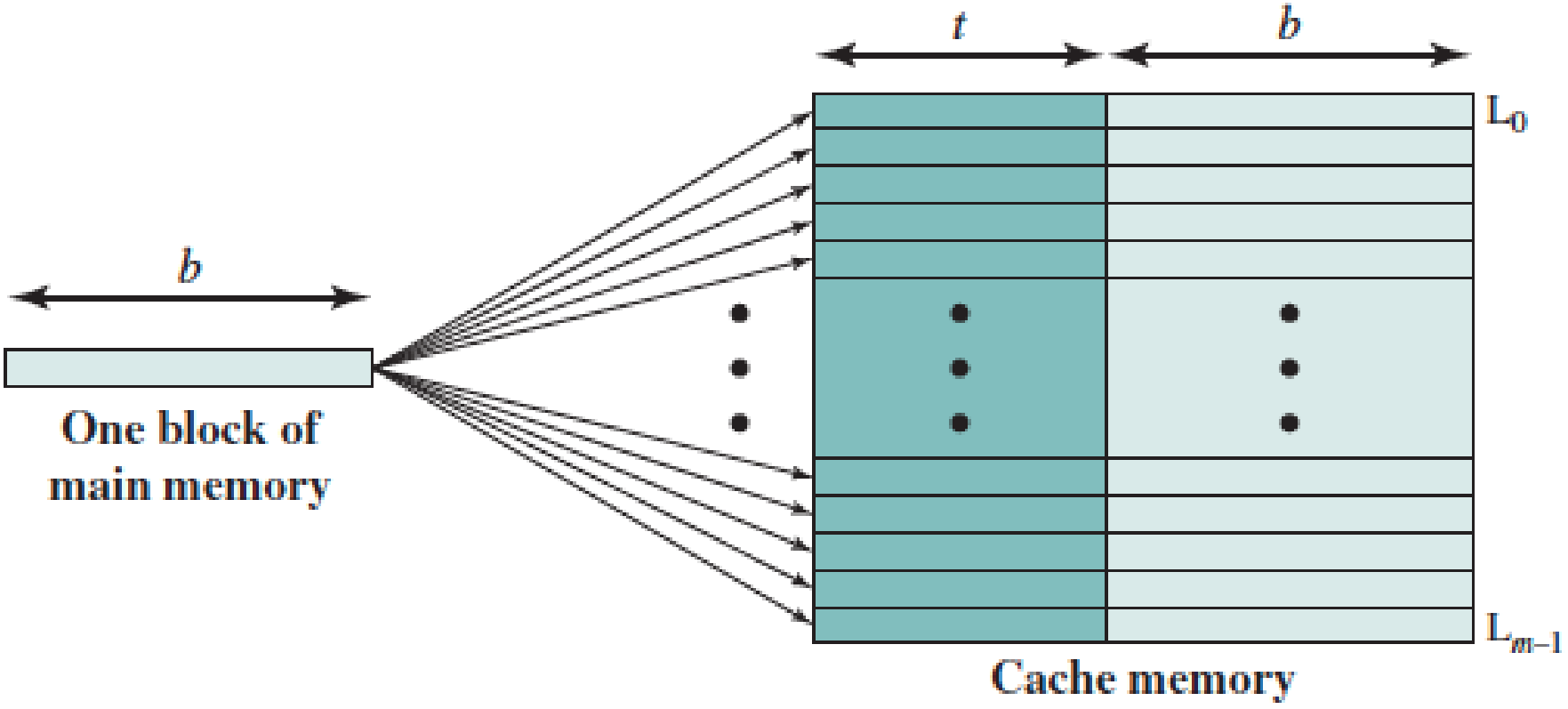
Direct Mapping Summary

- Address length = (s + w) bits
- Number of addressable units = 2s+w words or bytes
- Block size = line size = 2w words or bytes
- Number of blocks in main memory = 2s+ w/2w = 2s
- Number of lines in cache = m = 2r
- 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

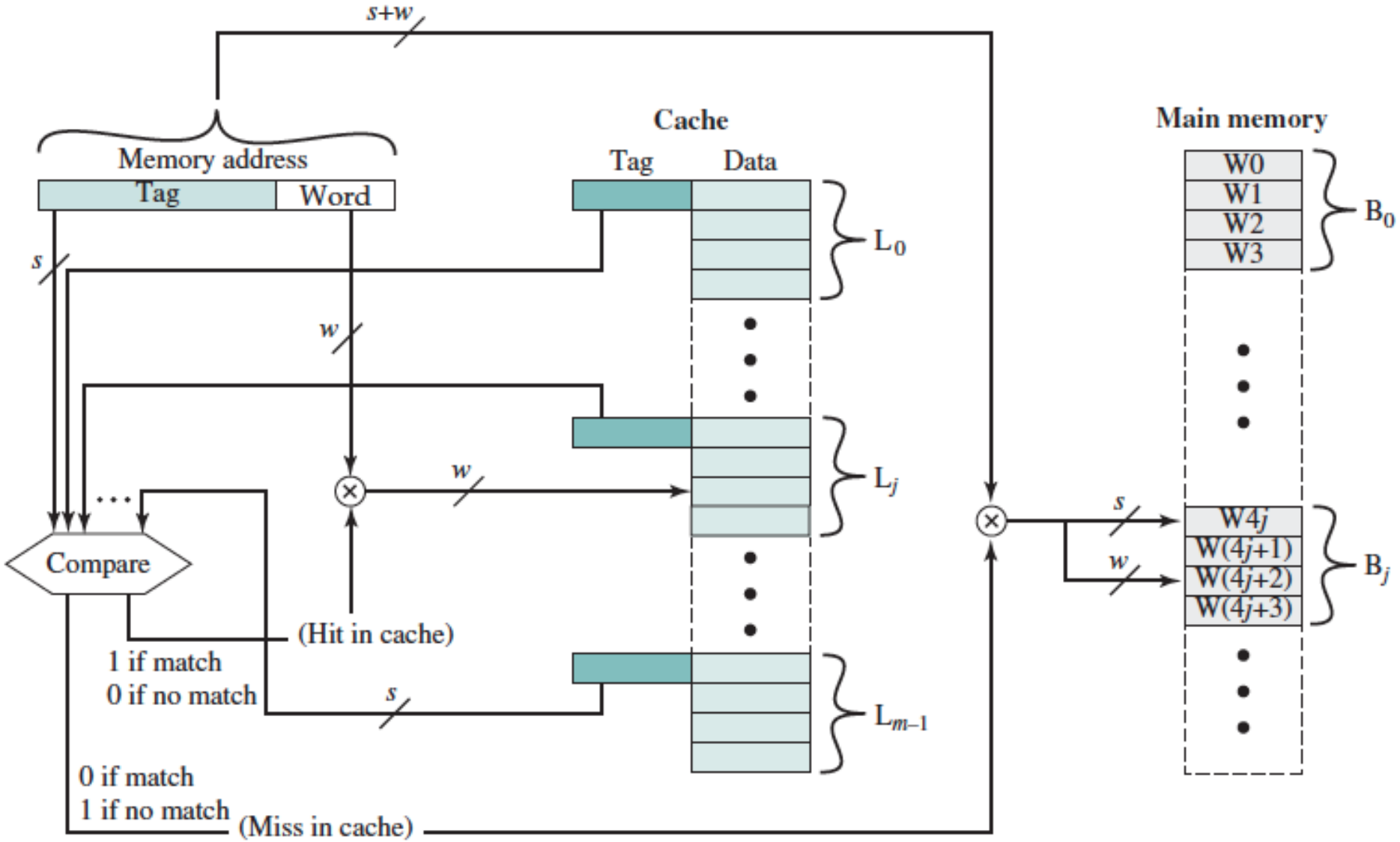
Associative Mapping



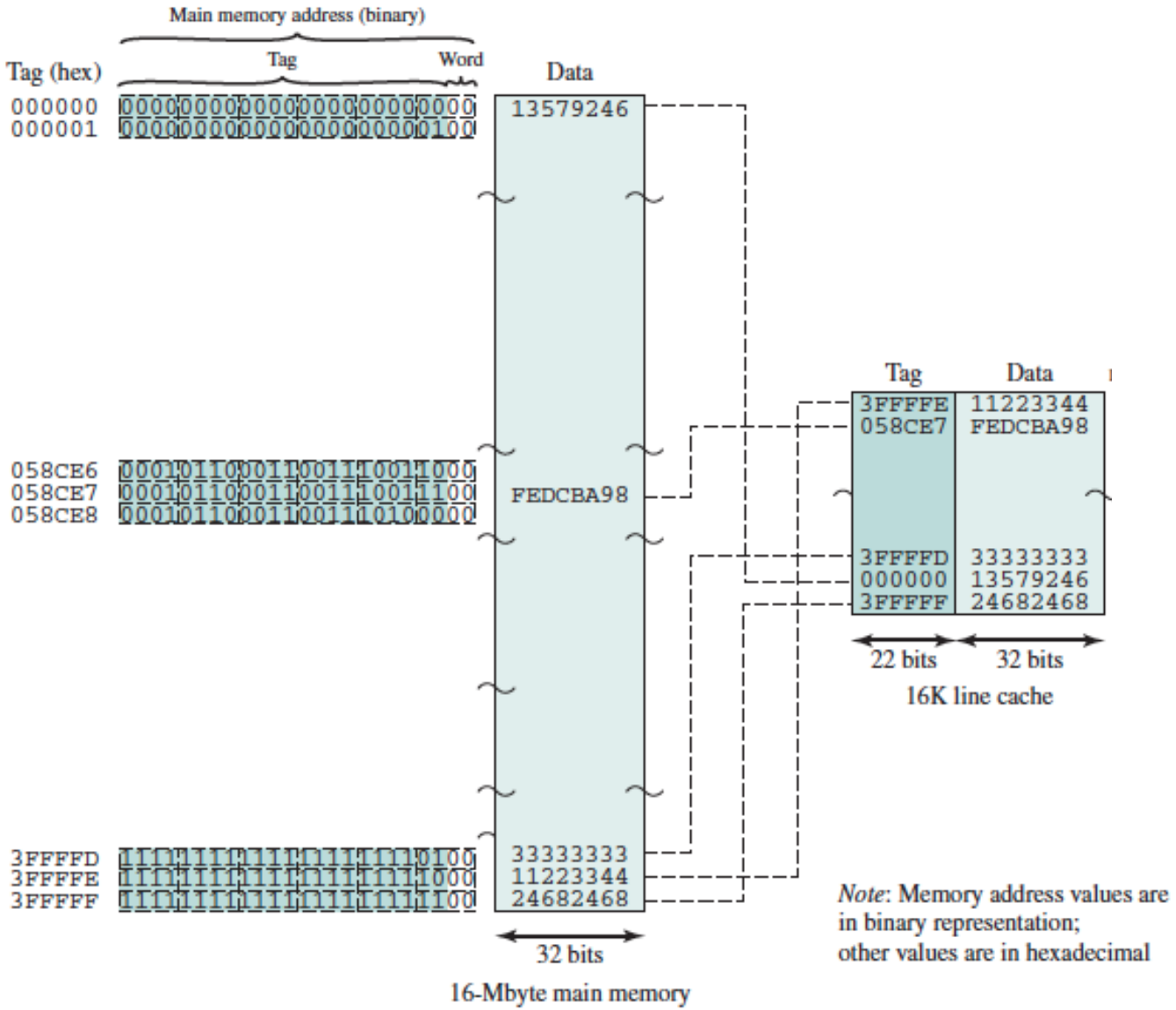
Associative Mapping

- 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



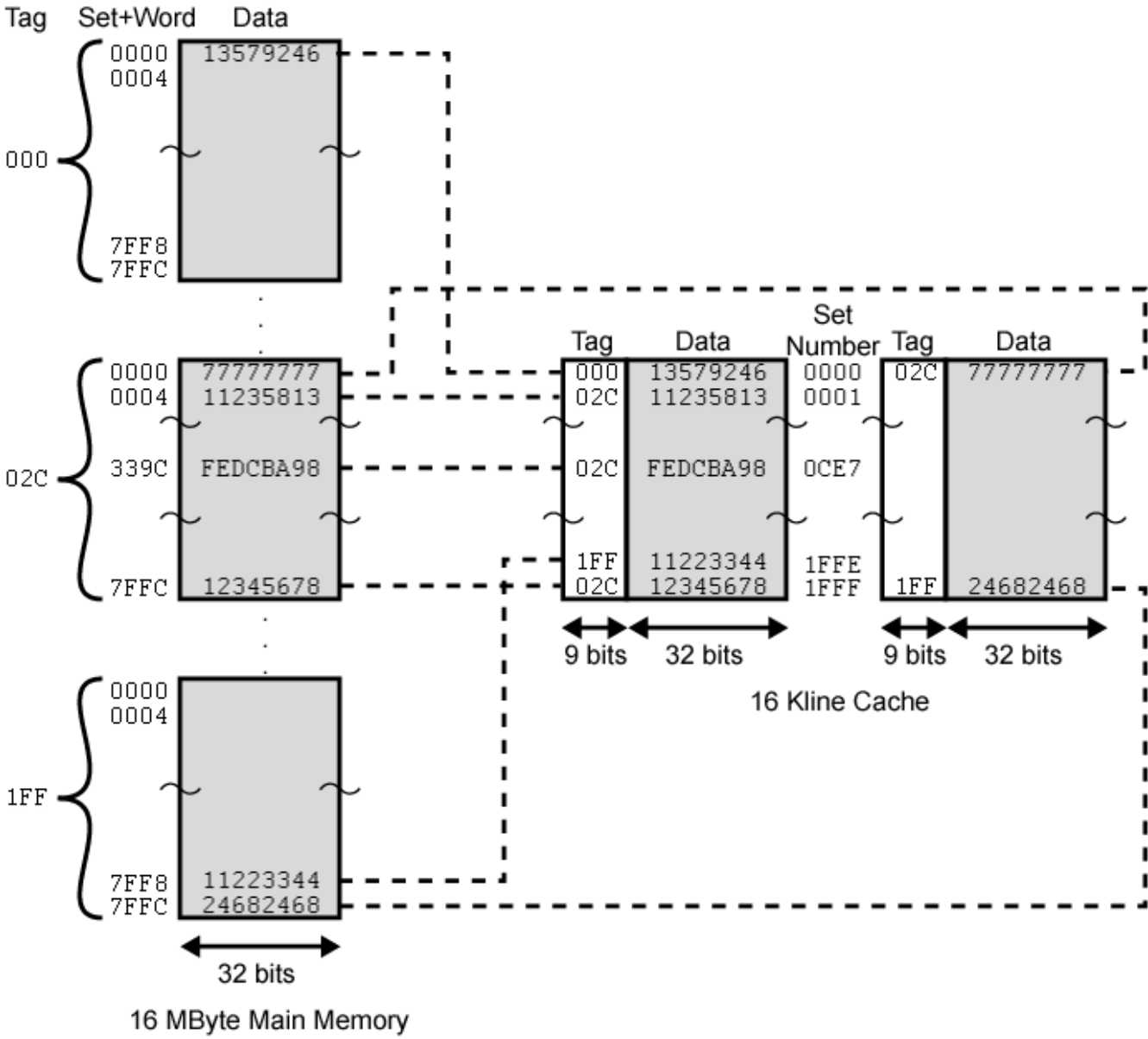
Associative Mapping Example

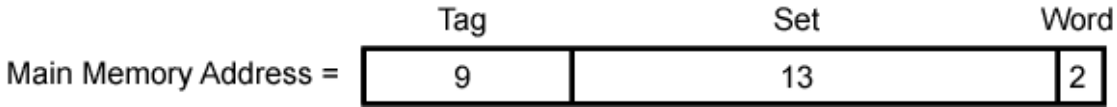


Set Associative Mapping

- Cache is divided into a number of sets
- Each set contains a number of lines
- A given block maps to any line in a given set
  - e.g. Block B can be in any line of set i
- e.g. 2 lines per set
  - 2 way associative mapping
  - A given block can be in one of 2 lines in only one set

Two Way Set Associative Mapping Example





Computer Architecture- TERM REPORT

- TOPIC: "**Designing of Cache Memory for specific application**"
- PERCENTAGE: **out of 10%**
- SUBMISSION DEADLINE: **2ND / JUNE. /2022**.

Computer Architecture-TERM REPORT

The report structure is:

- **Title**
- **Objective:** (The objective describes the goal of the reported work.)
- **Theory:** (The theory is a formal design comprising descriptions, etc.)
- **Design:** (The design part comprises flow charts, algorithms, tables, diagrams, derivations, etc.)
- **Implementation:** (The implementation is the description of functional modules, hierarchical relationships, etc.)
- **Results analysis (if any):** (The Analysis part should discuss other aspects, like the complexity of algorithms in terms of average and worst-case complexity for time and space, the robustness of the approach used, finer technical details, etc.)
- **Conclusion and Future Improvements:** (The conclusion and future aspect should summarize the report in brief, what improvements can be possible, limitations, various applications of this design, etc.)
- **Bibliography:** The bibliography section should provide a detailed list of references for books, journals, websites, conferences, and others in the standard accepted formats.