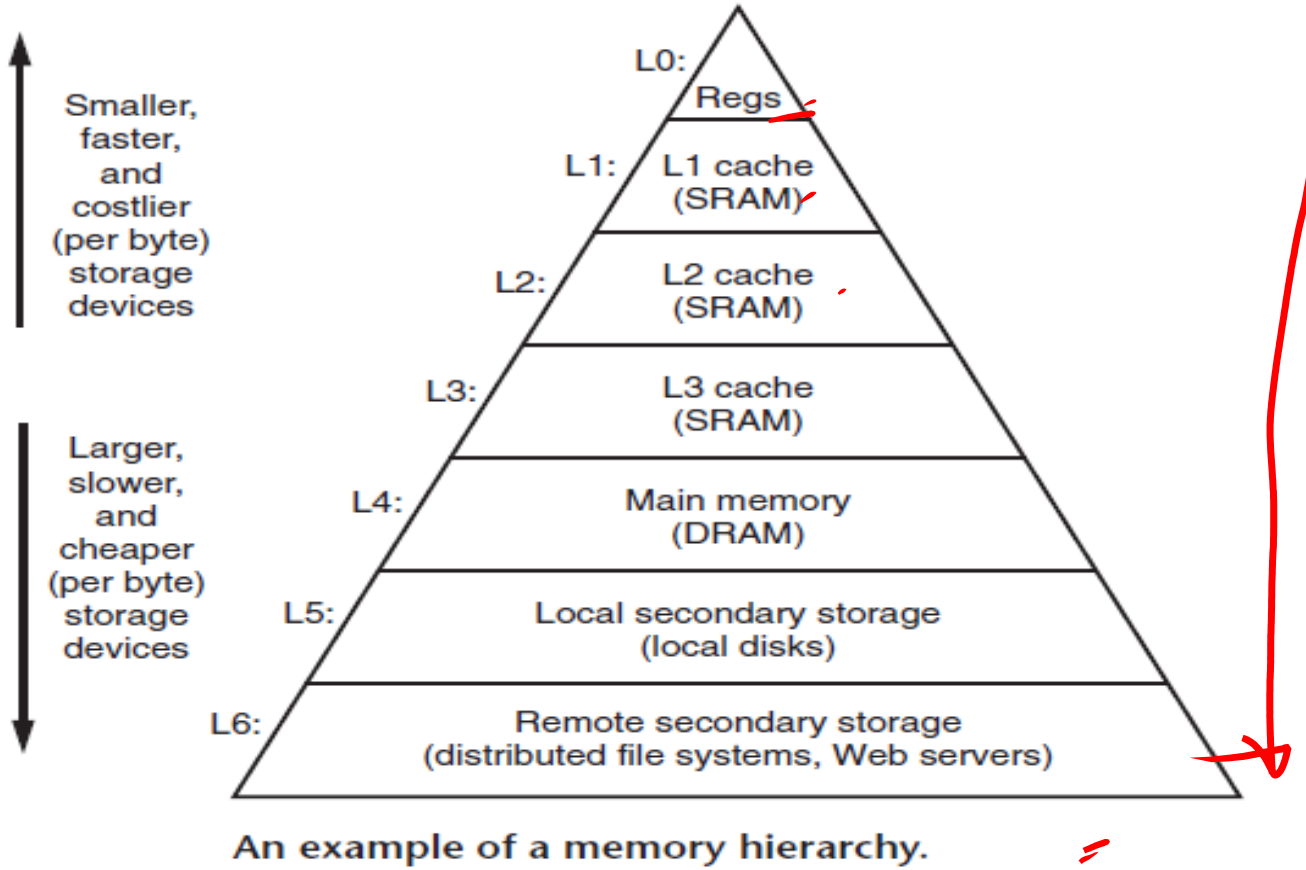
Computer Organization and Software Systems

Contact Session 4

Dr. Lucy J. Gudino

The Bottom Line

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



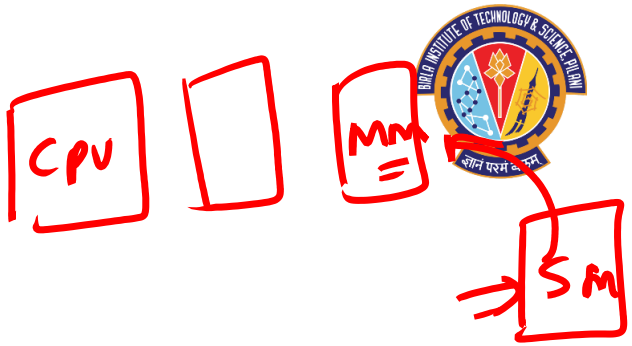
An example of a memory hierarchy.

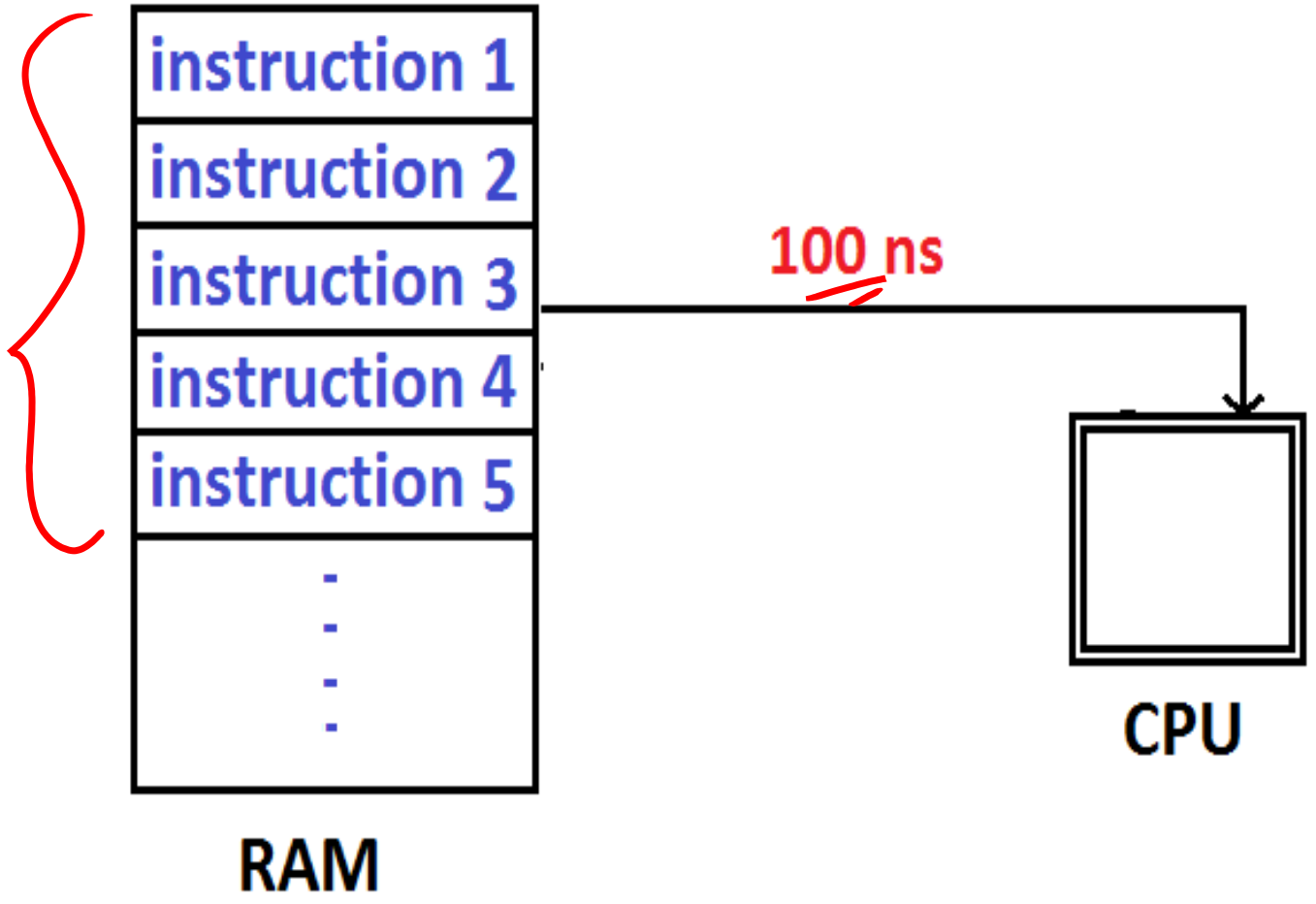
- Faster access time , ---------- cost per bit
- Greater capacity, ---------- cost per bit
- Greater capacity, ----------access time

Memory Hierarchy

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

Performance enhancement -Motivation



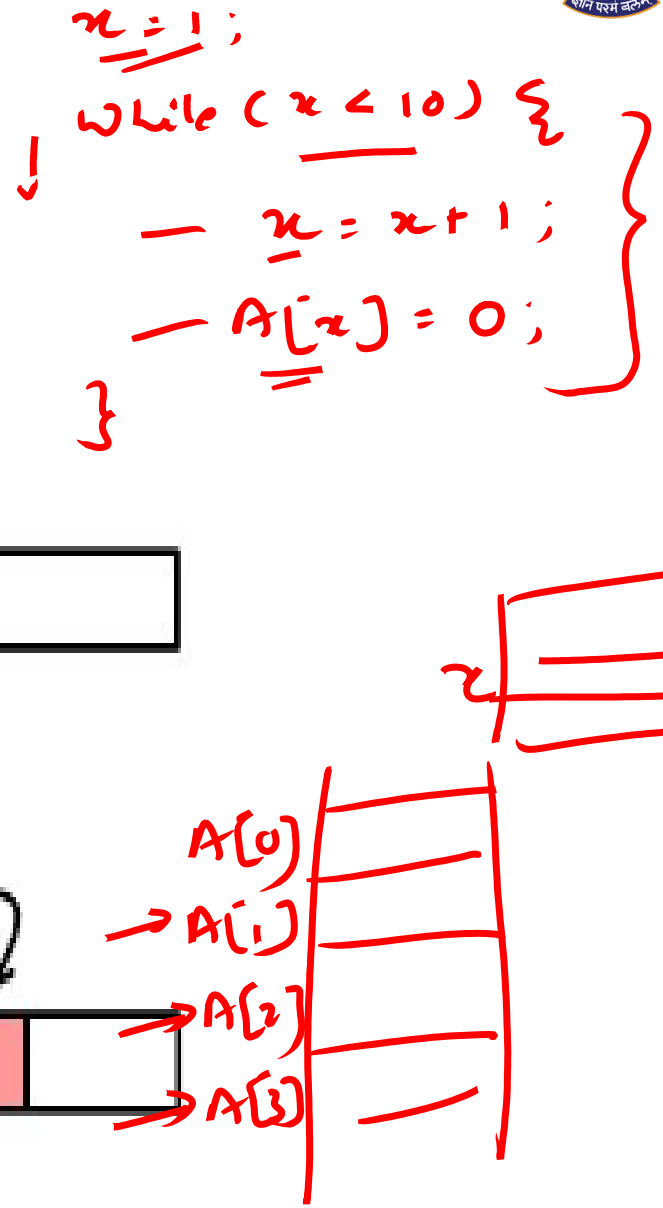


Performance enhancement -Motivation

Locality of Reference

During the course of the execution of a program, memory references tend to cluster

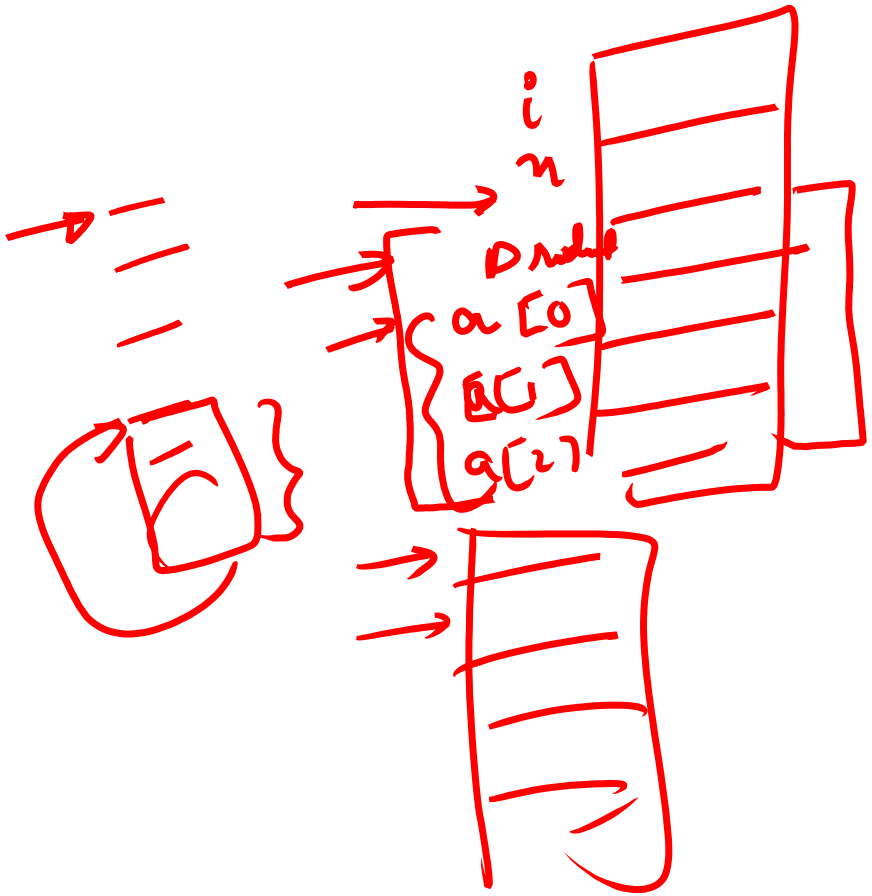
- **Temporal locality:** Locality in time
  - If an item is referenced, it will tend to be referenced again soon
- **Spatial locality:** Locality in space
  - If an item is referenced, items whose addresses are close by will tend to be referenced soon.



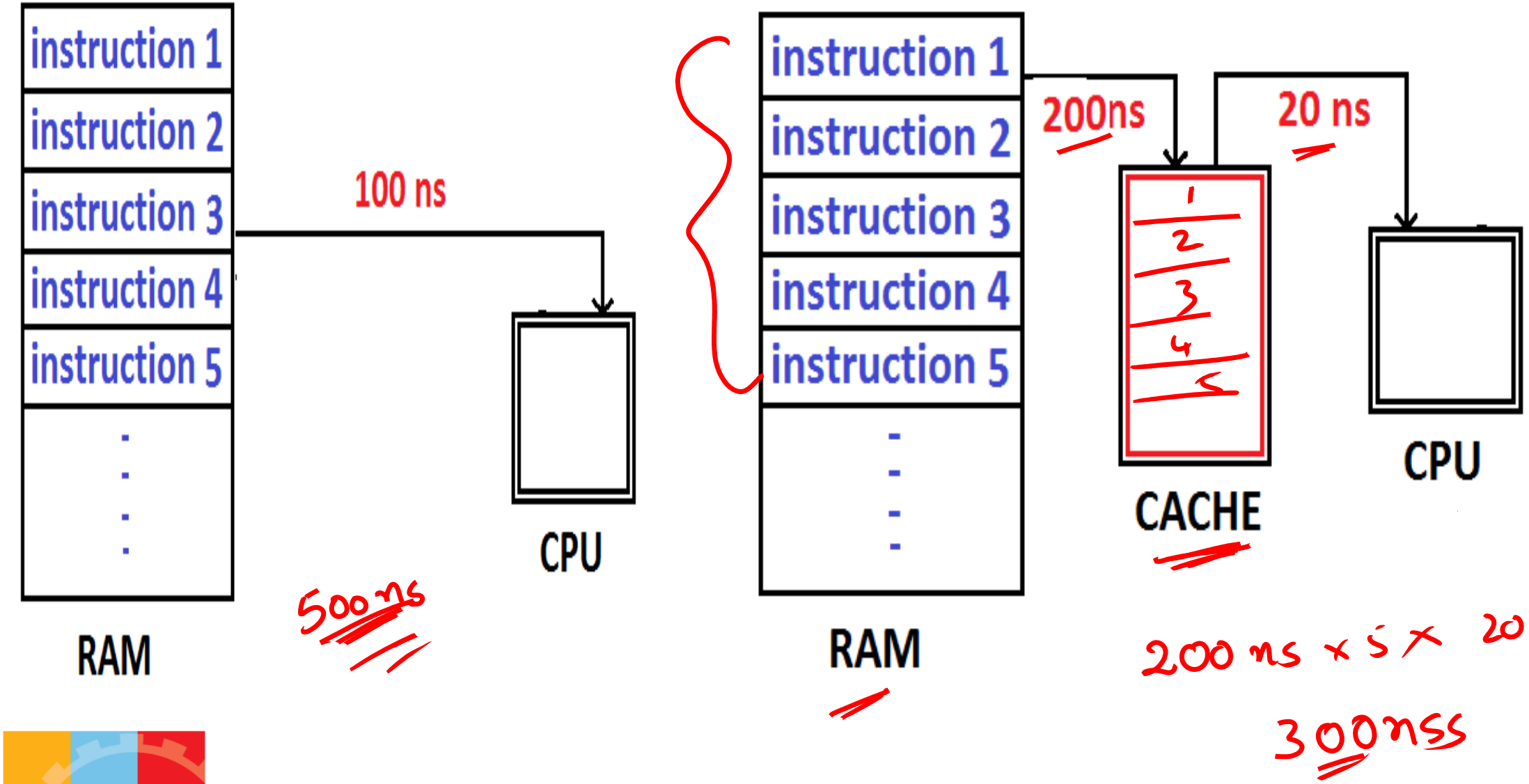
Example

```
product  = 1;
for ( i = 0; i < n-1; i++)
    product = product *  a[i] ;
```

- Data :
  - Access array elements in succession – spatial locality
  - Reference to "product" "i" and "n" in each iteration – Temporal locality
- Instructions :
  - Reference instructions in sequence : Spatial locality
  - Looping through : Temporal locality



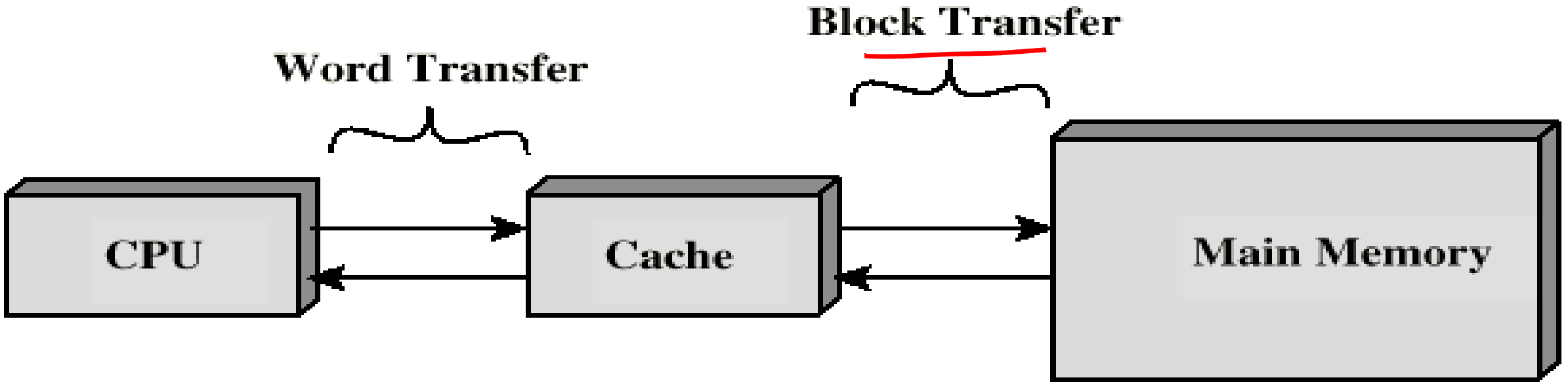
Performance enhancement -Motivation



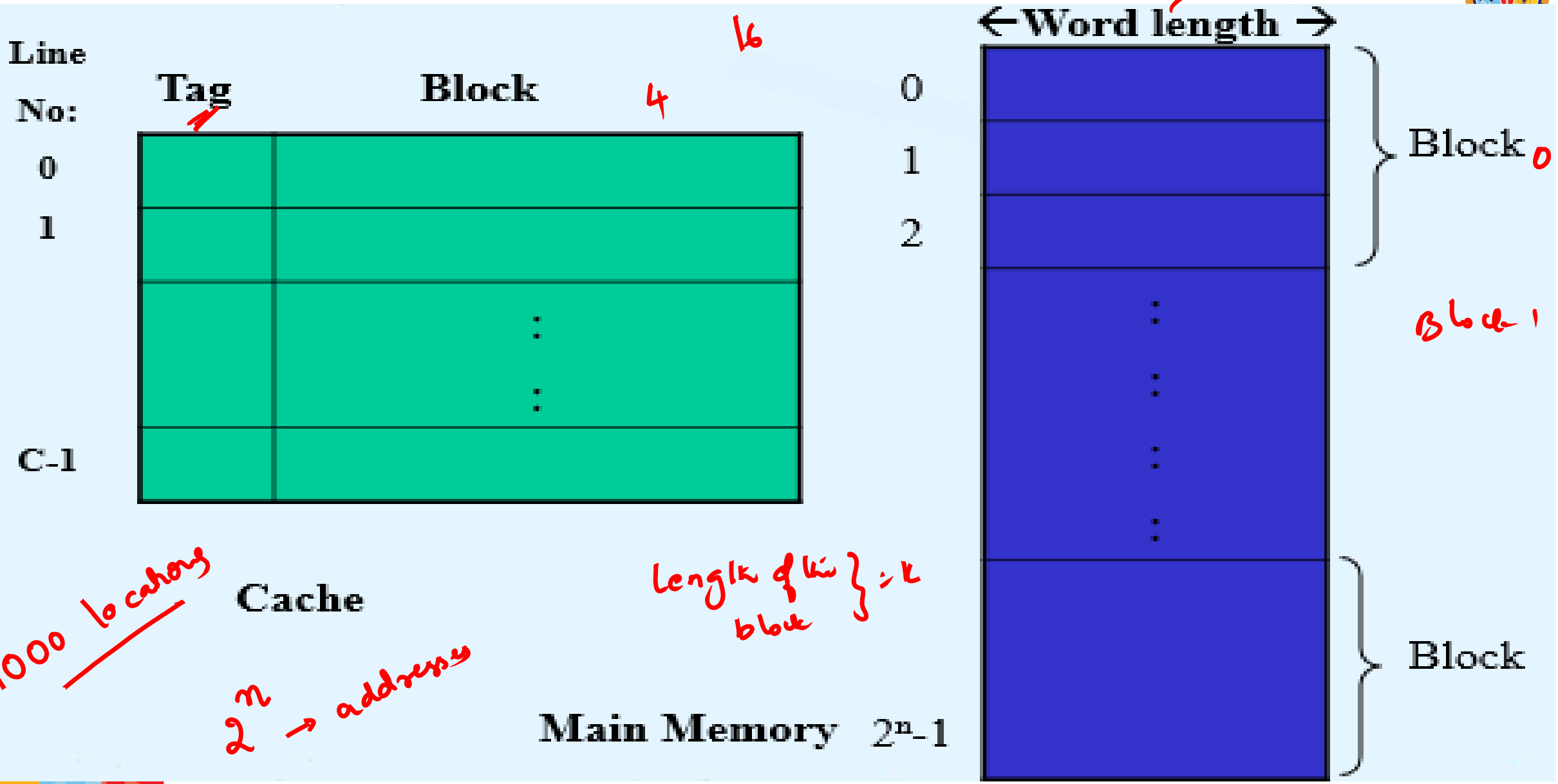
Cache

Cache

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



Cache and Main Memory Structure



$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

$2^4 = 16$

$2^5 = 32$

$2^6 = 64$

$2^7 = 128$

$2^8 = 256$

$2^9 = 512$

$2^{10} = 1024 = 1K$

$2^{11} = 2^1 \cdot 2^{10} = 2K$

$2^{12} = 2^2 \cdot 2^{10} = 4K$

$2^{13} = 8K$

$2^{14} = 16K$

$2^{15} = 32K$

$2^{16} = 64K$

$2^{17} = 128K$

$2^{18} = 256K$

$2^{19} = 512K$

$2^{20} = 1024K$

$2K \cdot 1K$

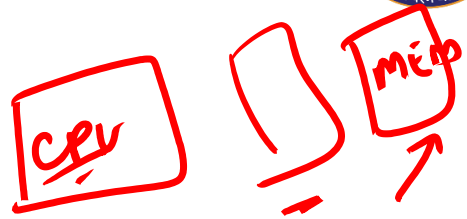
$= 1MB$

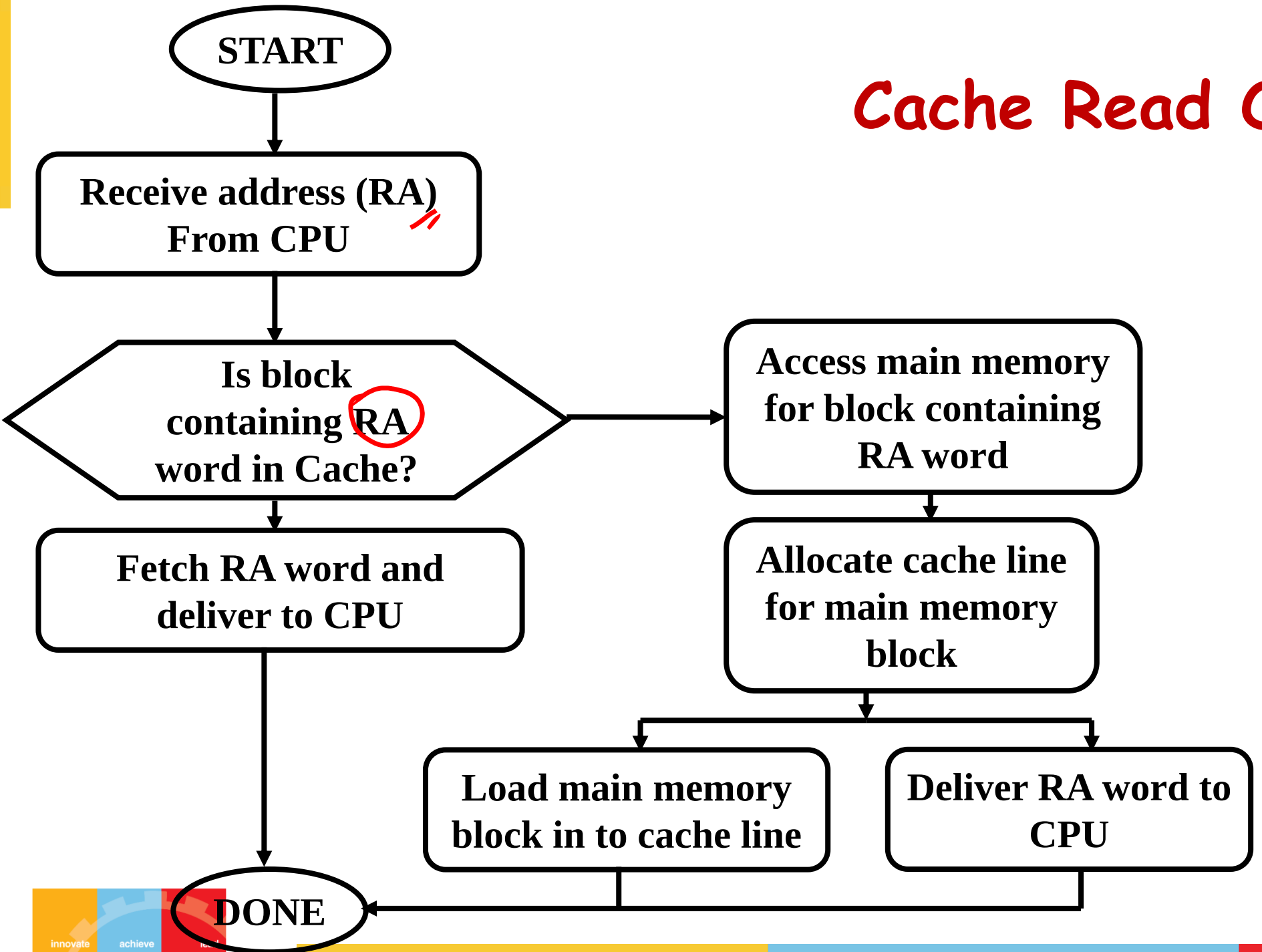
$2^{30} = 1GB$ $2^{40} = 1TB$

$=$

1000

Cache Read Operation





Performance of cache

- Hit ratio : Number of Hits / total references to memory
- Hit
- Miss

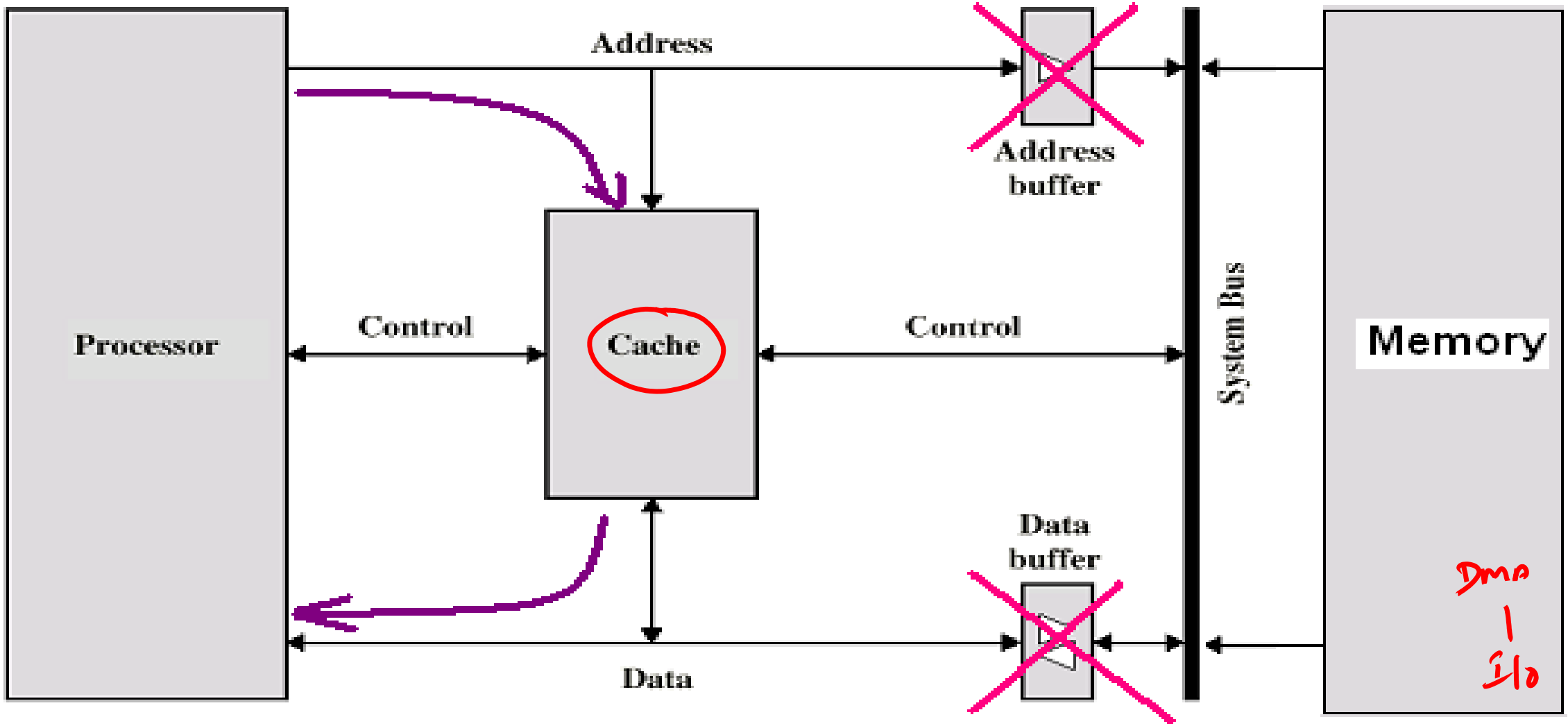
100/100 = 1

0/100 = 0

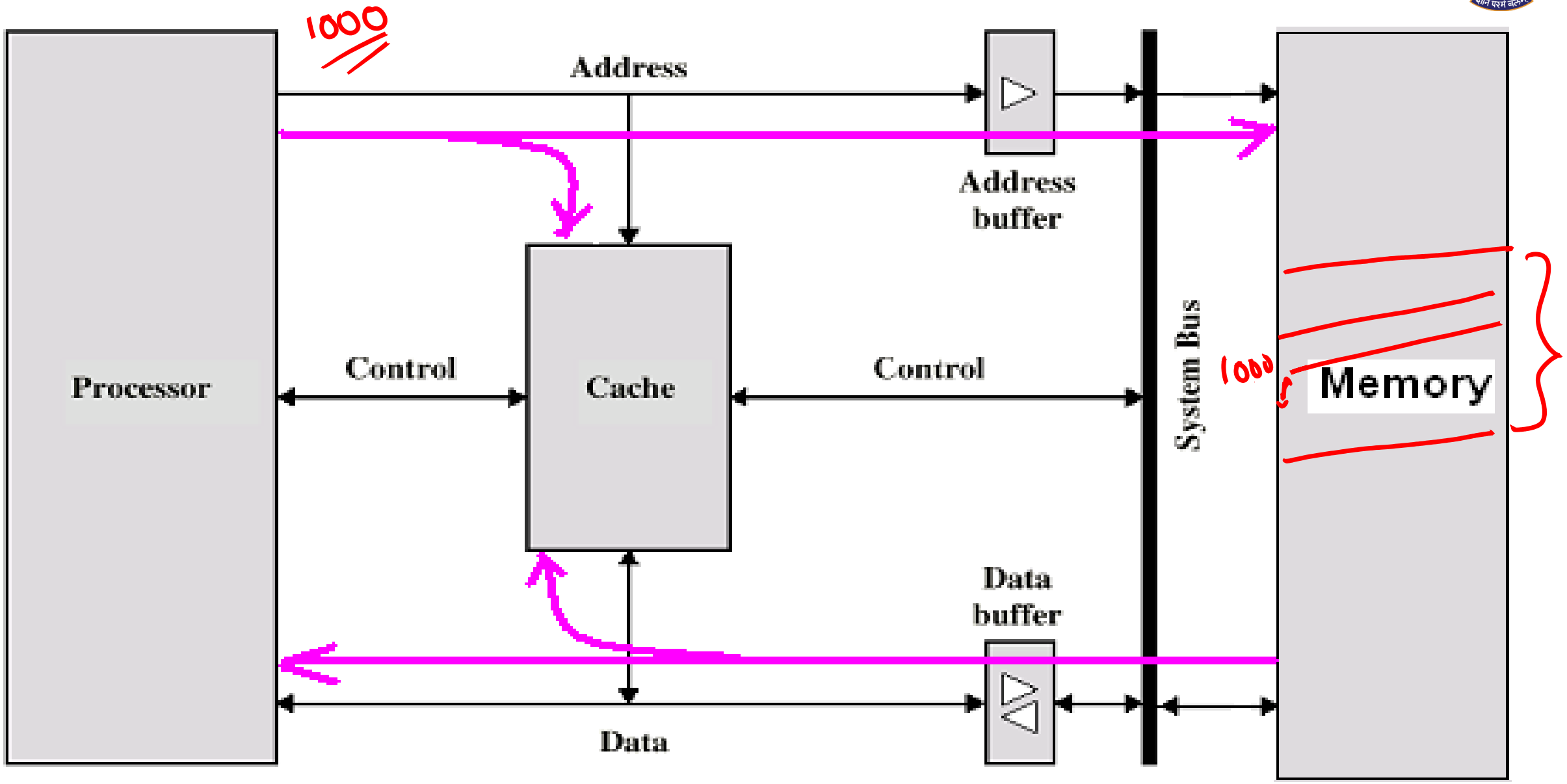
0 - 100 +

Read Hit

Read < hit / miss
Write < hit / miss

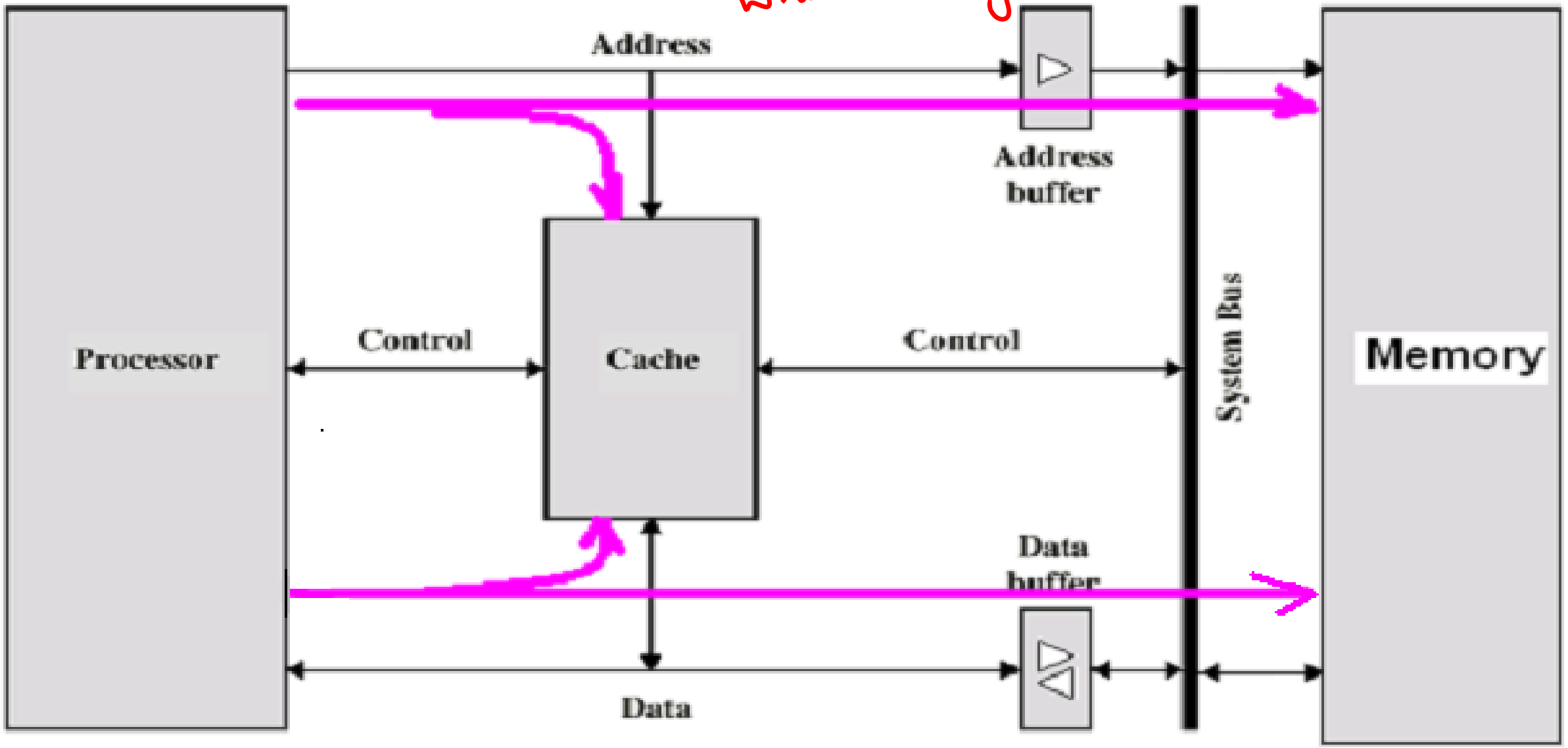


Read Miss

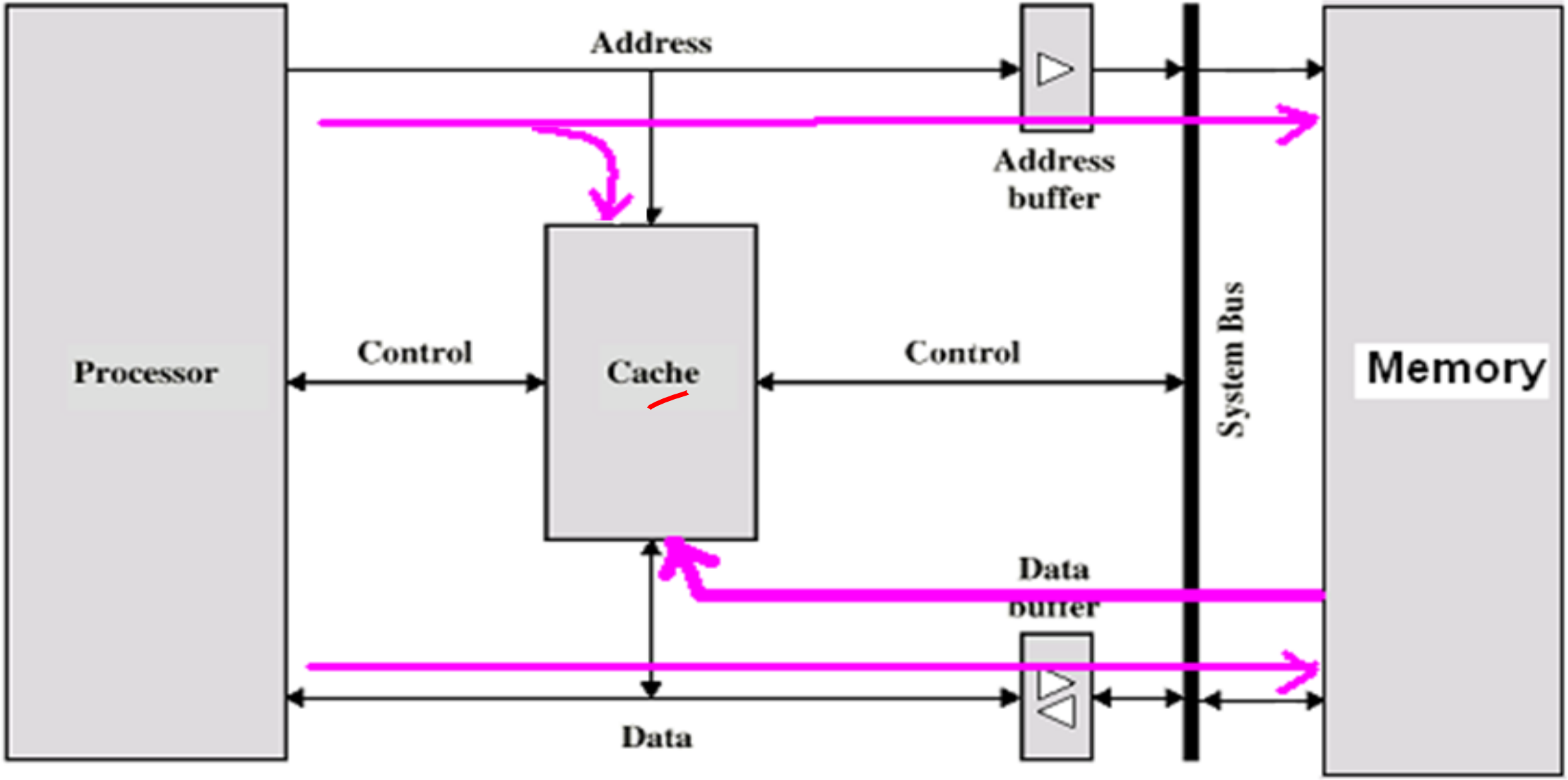

1000
Address
Address buffer
Processor
Control
Cache
Control
System Bus
1000
Memory
Data buffer
Data

Write hit

Write back policy
Write through policy } ⇒



Write miss

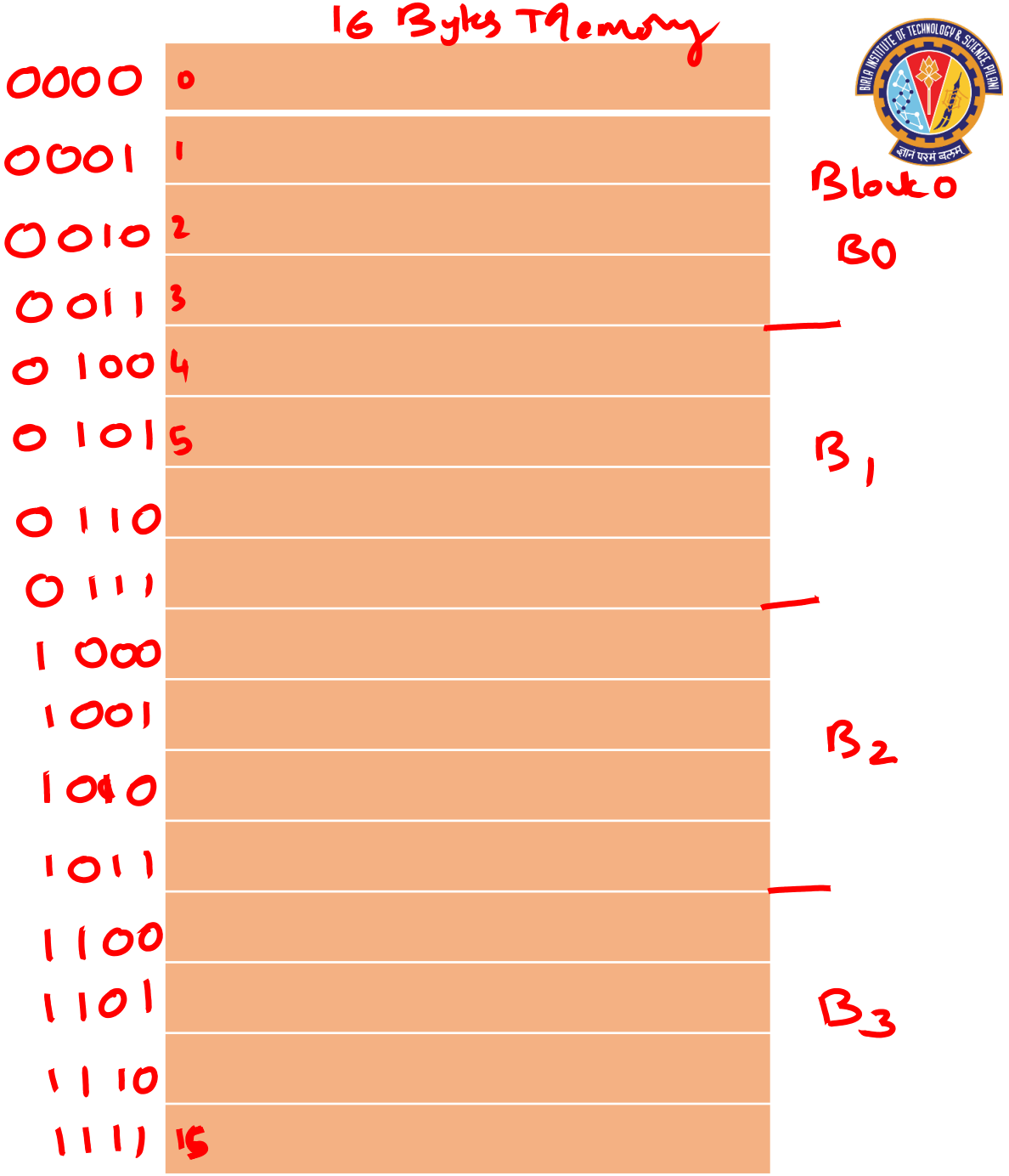

Address
Address buffer
Processor
Control
Cache
Control
System Bus
Memory
Data buffer
Data

Mapping Function

- How memory blocks are mapped to cache lines
- Three types
  - -Direct mapping
  - -Associative mapping
  - -Set Associative mapping

Direct Mapped Cache

- 16 Bytes main memory ⇒ # address bits = 4 bits
  - How many address bits are required?
- Memory block size is 4 bytes
- Cache of 8 Byte
  - How many cache lines? ⇒ 8/4 = 2
  - cache contains 2 lines (4 bytes per Line)



Direct Mapped Cache

- 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**
  - **i = j modulo m**

where i = cache line number

j = main memory block no.

m = no.of lines in the cache

Direct Mapped Cache

i = j modulo m

For b_0 : $0 \div 2 \Rightarrow 0 \Rightarrow B_0 \Rightarrow b_0$

b_1 $1 \div 2 \Rightarrow 1 \Rightarrow B_1 \Rightarrow l_1$

b_2 $2 \div 2 \Rightarrow 0 \Rightarrow B_2 \Rightarrow l_0$

b_3 $3 \div 2 \Rightarrow 1 \Rightarrow B_3 \Rightarrow l_1$

$l_0 \Rightarrow B_0$ & B_2

$l_1 \Rightarrow B_1$ & B_3

- Address is split in three parts:
  - Tag
  - Line
  - Word

1 2 bits

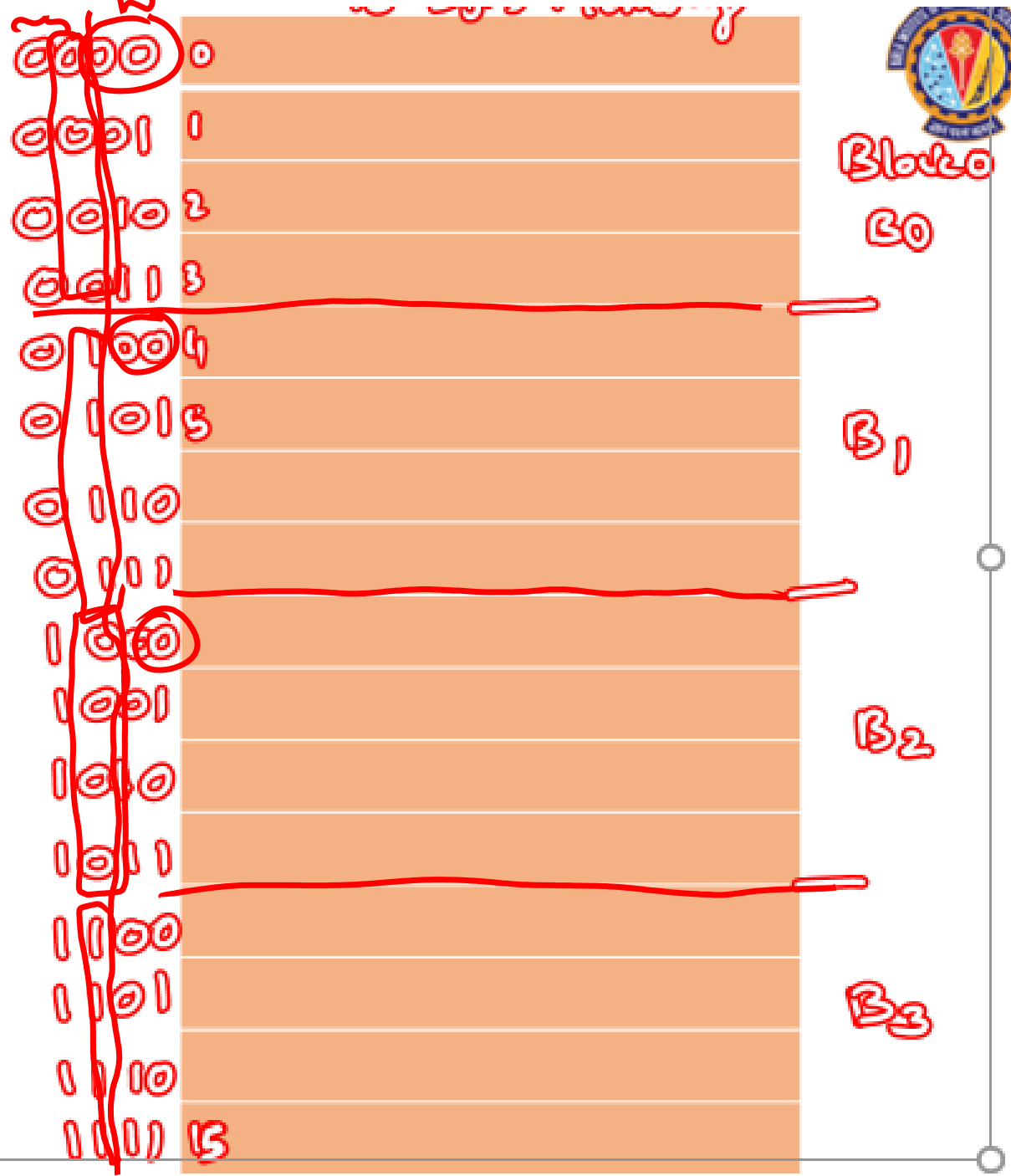
Tag	line	word

1

2 lines

$2^n = 4$

2^2



address of MM = 4 bits
block = 4 blocks
Cache lines = 2
of words stored in a block = 4 words

$l_0 \rightarrow B_0, B_2$

$l_1 \rightarrow B_1, B_3$

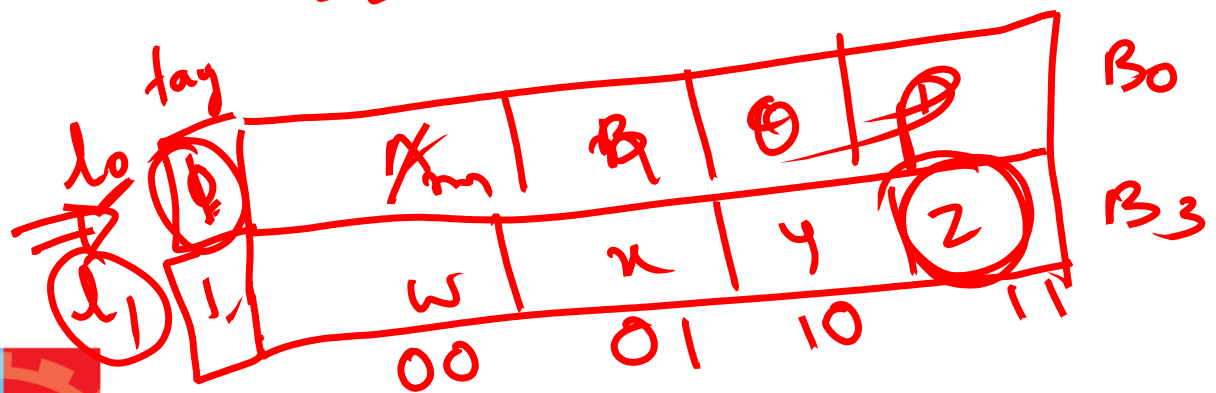
0 - l_0

1 - l_1

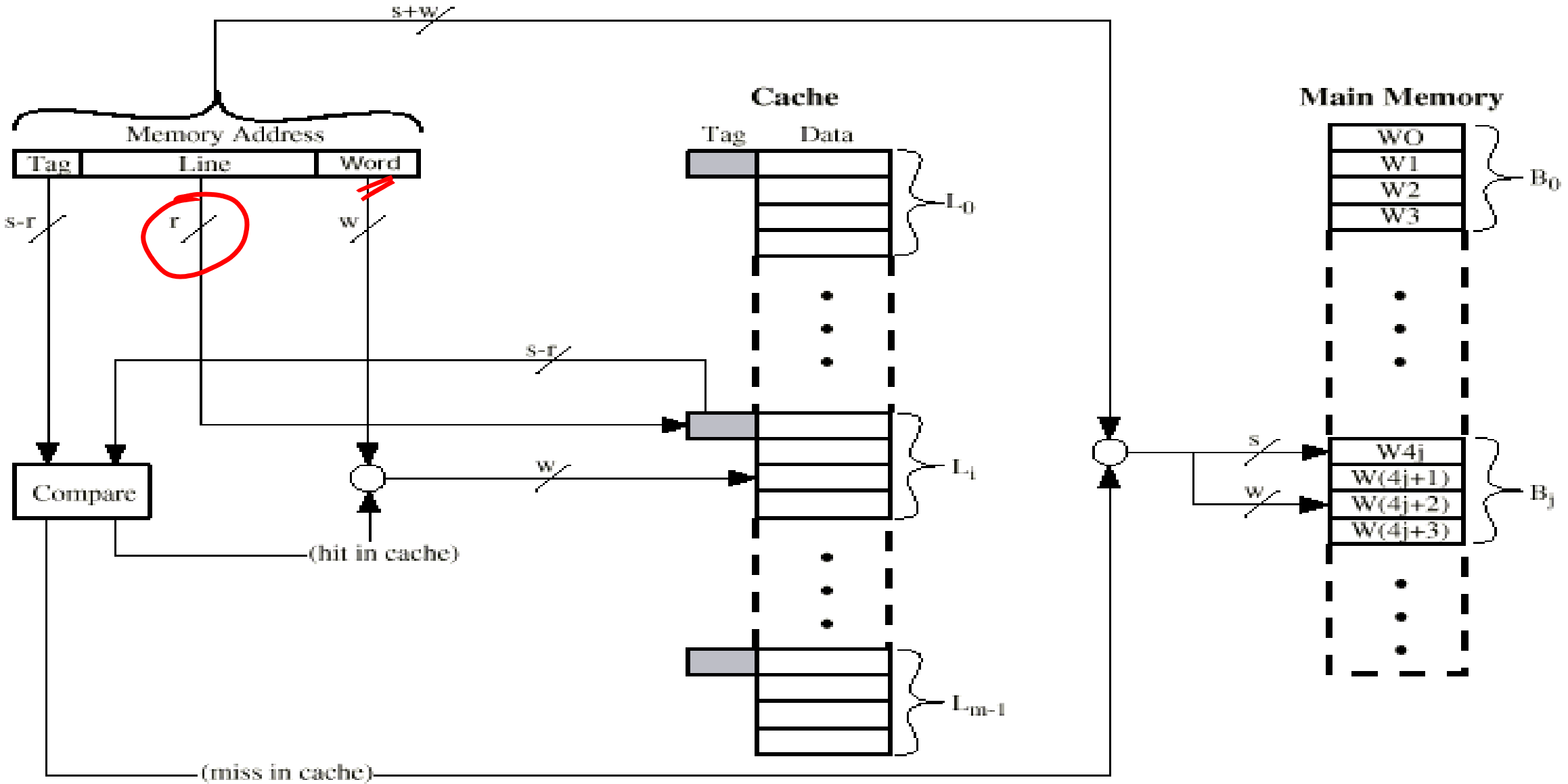
4 words / block

2 bits to address a word in a block

How many bits are needed to identify a Cache line



Direct Mapped Cache Organization



Direct mapped cache- Summary

$2^n =$ —

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

Direct mapped cache- Summary

Direct mapped cache-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

Problem 1 : Direct Mapped Cache

•Given :

- Cache of 64KByte, Cache block of 4 bytes
- 16MBytes main memory

•Find out

a) Number of bits required to address the main memory

b) Number of blocks in main memory

c) Number of cache lines

d) Number of bits required to identify a word (byte) in a block

e) Number of bits to identify a block

f) Tag, Line, Word

Solution 1

Cache block size = MM block size

slot

line

$2^{10} = 1K$

$2^{20} = M$

$2^{30} = G$

$2^{40} = T$

•**Given :**

- **Cache of 64kByte, Cache block of 4 bytes**
- **16MBytes main memory**

•Find out

a) Number of bits required to address the main memory = 24 bits

Capacity of the main memory = 16 MB

$2^n = 16M$

$= 2^4 \cdot 2^{20} = 2^{24}$

•b) Number of blocks in main memory

Capacity of the mm / block size = $\frac{16M}{4}$ = 4 M blocks

$= 2^{22}$ blocks

•c) Number of cache lines

= Cache size / block size = $\frac{64K}{4}$ = 16 K lines

Solution 1

block size = 4 byte

$2^n = 4$

$n = 2$

- **Given :**
  - **Cache of 64kByte, Cache block of 4 bytes**
  - **16MBytes main memory**
- Find out
- d) Number of bits required to identify a word (byte) in a block? = 2 bit
- e) Number of bits required to identify a block = 24 bit - 2 bit = 22 bit
- f) Tag, Line, Word

cache lines = 16 K lines = 2^{14}

Tag s-r	Line r	Word w
8	14	2

Problem 2

Consider a machine with a byte addressable main memory of 2^{16} bytes and block size of 8 bytes. Assume that a direct mapped cache consisting of 32 lines is used with this machine.

a. How is a 16-bit memory address divided into tag, line number, and byte number?

b. Into what line would bytes with each of the following addresses be stored?

0001	0001	0001	1011
1100	0011	0011	0100
1101	0000	0001	1101
1010	1010	1010	1010

c. Suppose the byte with address 0001 1010 0001 1010 is stored in the cache. What are the addresses of the other bytes stored along with it?

d. How many total bytes of memory can be stored in the cache?

e. Why is the tag also stored in the cache?

Solution 2

Consider a machine with a byte addressable main memory of 2^{16} bytes and block size of 8 bytes. Assume that a direct mapped cache consisting of 32 lines is used with this machine.

a. How is a 16-bit memory address divided into tag, line number, and byte number?

b. Into what line would bytes with each of the following addresses be stored?

	Tag	Line	word
①	0001 0001	0001	1011
	1100 0011	0011	0100
	1101 0000	0001	1101
	1010 1010	1010	1010

16 bit

16 8 4 2 1

0 0011 ⇒ l_3

0 0110 ⇒ l_6

0 0011 ⇒ l_3

1 0101 ⇒ l_{21}

l_3

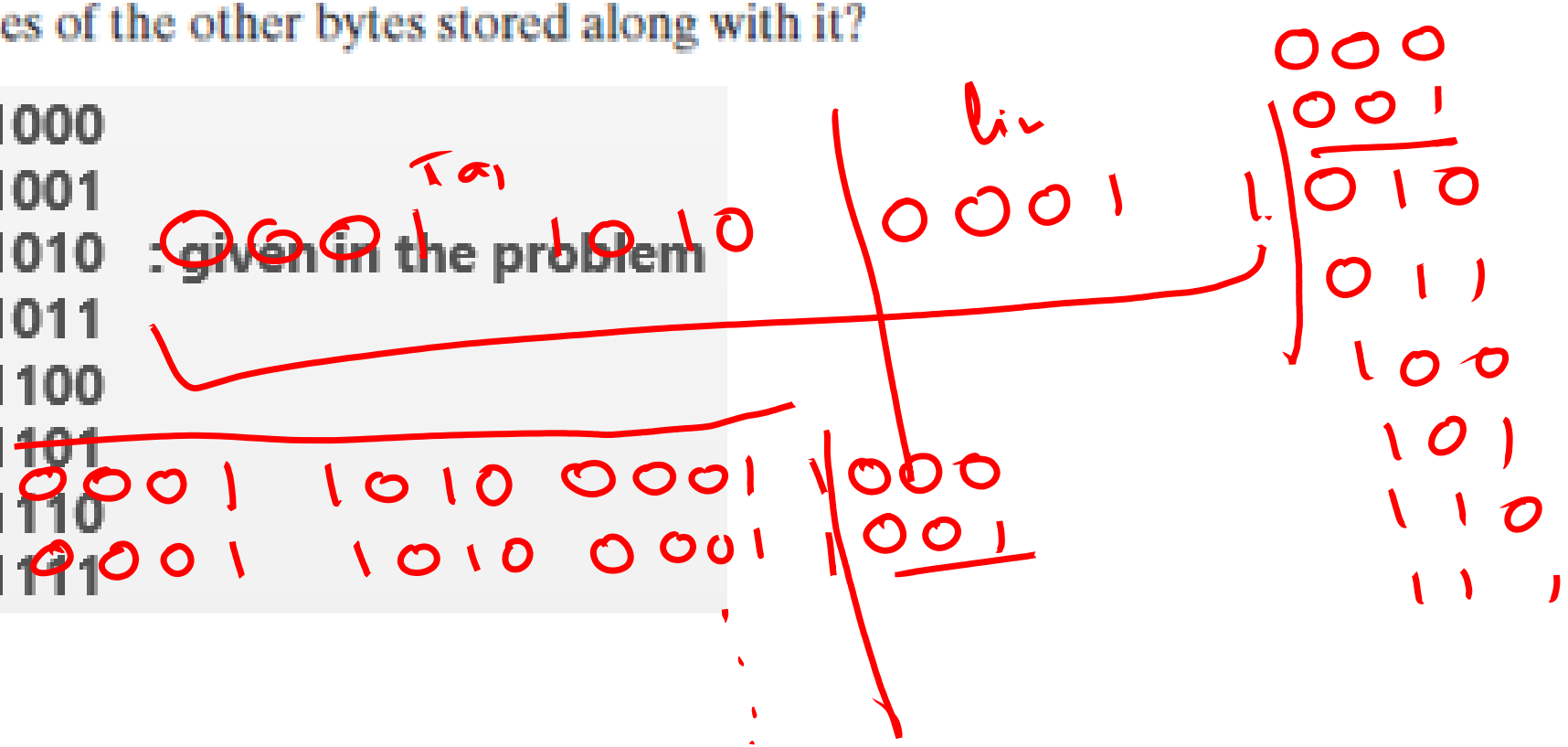
0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1

Solution 2

Consider a machine with a byte addressable main memory of 2^{16} bytes and block size of 8 bytes. Assume that a direct mapped cache consisting of 32 lines is used with this machine.

c. Suppose the byte with address 0001 1010 0001 1010 is stored in the cache. What are the addresses of the other bytes stored along with it?

0001 1010 0001 1**000**
0001 1010 0001 1**001**
0001 1010 0001 1**010** : given in the problem
0001 1010 0001 1**011**
0001 1010 0001 1**100**
0001 1010 0001 1**101**
0001 1010 0001 1**110**
0001 1010 0001 1**111**



Solution 2

Consider a machine with a byte addressable main memory of 2^{16} bytes and block size of 8 bytes. Assume that a direct mapped cache consisting of 32 lines is used with this machine.

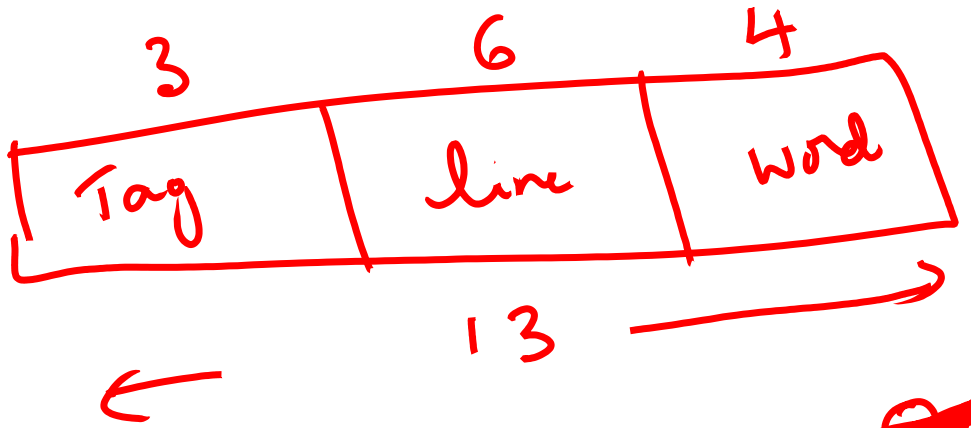
d. How many total bytes of memory can be stored in the cache?

$32 \times 8 = 256$ bytes

e. Why is the tag also stored in the cache?

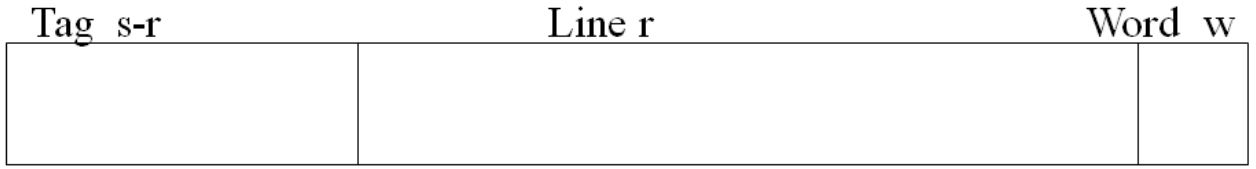
Problem 3

- Consider a direct-mapped cache with 64 cache lines and a block size of 16 bytes and main memory of 8K (Byte addressable memory) . To what line number does byte address 1200H map?



Hexadecimal	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Problem 4



- The system uses a L1 cache with direct mapping and 32-bit address format is as follows:
- bits 0 - 3 = offset (word)
- bits 4 - 14 = index bits (Line)
- bits15 - 31 = tag

a) What is the size of cache line?

b) How many Cache lines are there?

c) How much space is required to store the tags in the L1 cache?

d) What is the total Capacity of cache including tag storage?

Problem 5

- 16 Bytes main memory, Memory block size is 4 bytes, Cache of 8 Byte (cache is 2 lines of 4 bytes each)
- Block access sequence :

•0 2 0 2 2 0 0 2 0 0 0 2 1

- Find out hit ratio.

Problem 5 – Direct Mapped Cache

- 16 Bytes main memory, Memory block size is 4 bytes, Cache of 8 Byte (cache is 2 lines of 4 bytes each)

  Block access sequence :

  0 2 0 2 2 0 0 2 0 0 0 2 1

Find out hit ratio.

0 2 0 2 2 0 0 2 0 0 0 2 1

Problem 6

- Suppose a 1024-byte cache has an access time of 0.1 microseconds and the main memory stores 1 Mbytes with an access time of 1 microsecond. A referenced memory block that is not in cache must be loaded into cache .
- Answer the following questions:

a) What is the number of bits needed to address the main memory?

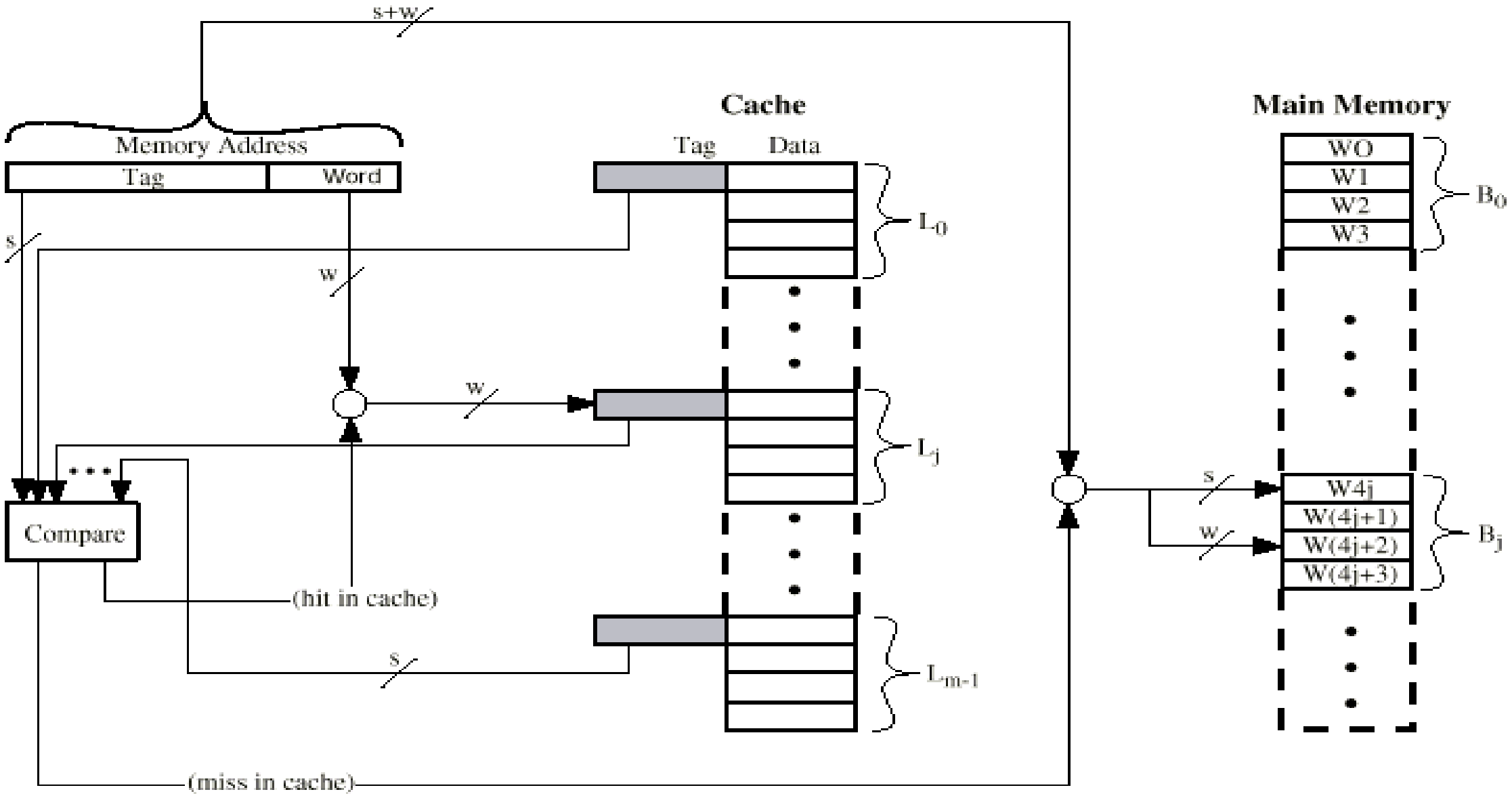
a) If the cache hit ratio is 95%, what is the average access time for a memory reference?

Avg access time = hit ratio * cache access + (1- hit ratio) * (cache access + memory access)

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

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 = 2^{w} words or bytes
- Number of blocks in main memory = $2^{s+ w}/2^{w} = 2^{s}$
- Number of lines in cache = undetermined
- Size of tag = s bits

Problem 7

- Given :
  - Cache of 128KByte, Cache block of 8 bytes
  - 32 MBytes main memory
- Find out

a) Number of bits required to address the memory

b) Number of blocks in main memory

c) Number of cache lines

d) Number of bits required to identify a word (byte) in a block?

e) Tag, Word

Problem 8

•Cache of 64KByte, Cache block of 4 bytes , 16 M Bytes main memory and associative mapping.

Fill in the blanks:

Number of bits in main memory address = _____

Number of lines in the cache memory = _____

Word bits = _____

Tag bits = _____

Problem 9

- 16 Bytes main memory, Memory block size is 4 bytes, Cache of 8 Byte (cache is 2 lines of 4 bytes each) and associative mapping. Block access sequence :

  0 2 0 2 2 0 0 2 0 0 0 2 1

  Find out hit ratio.

0 2 0 2 2 0 0 2 0 0 0 2 1