Unit 5
Hierarchical Memory System

Characteristics of Memory Systems

Study of an effective memory organization that support the processing power of processor.

Classification of memory system according to their characteristics:

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

Characteristics of Memory Systems

1. Location:

A. Internal Memory:

1. Internal memory is often equated with main memory.
2. The processor requires its own local memory, in the form of registers.
3. Example: Processor register, Cache Memory and Main Memory.

B. External Memory:

1. External memory consists of peripheral storage devices
2. that are accessible to the processor via I/O controllers.
3. Example: Magnetic Disk, Tape

2. Capacity:

The amount of data device can hold.

1. For internal memory, this is typically expressed in terms of bytes (1 byte = 8 bits) or words. Common word lengths are 8, 16, and 32 bits.
2. Example , if memory device is given as 64K x 16. This means the device has a word size of 16 bit and total of 4096 (4K) words in main memory.
3. External memory capacity is typically expressed in terms of bytes.

3. Unit of Transfer:

1. Internal memory, the unit of transfer is equal to the number of electrical lines into and out of the memory module. This may be equal to the **word length.**
2. External memory, data are often transferred in much larger units than a word, and these are referred to as **blocks.**

4. Methods of Accessing units of Data

1. **Sequential Access:**

a. **Memory is organised into units of data, called Record.**

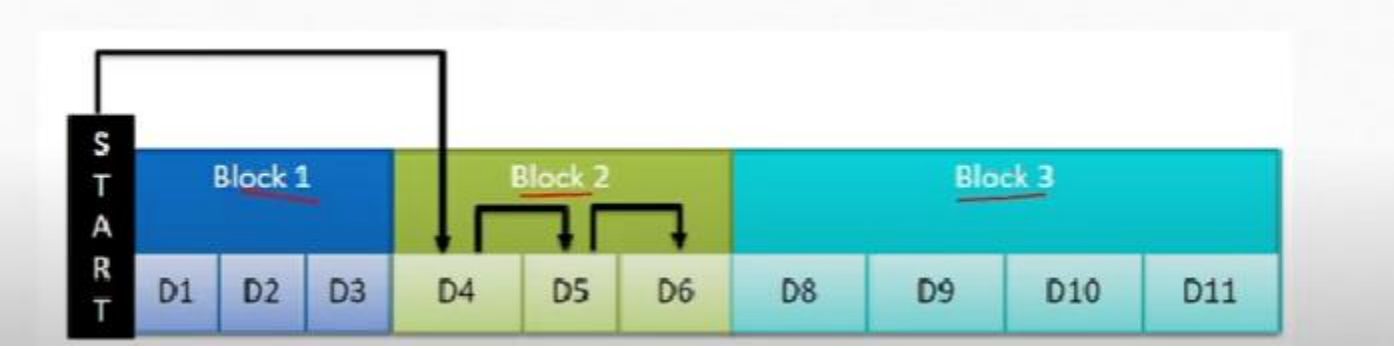
b. **Access must be made in linear sequence.**

c. **Access time depends on location of data and previous location.**

d. **Example: Magnetic Tape**

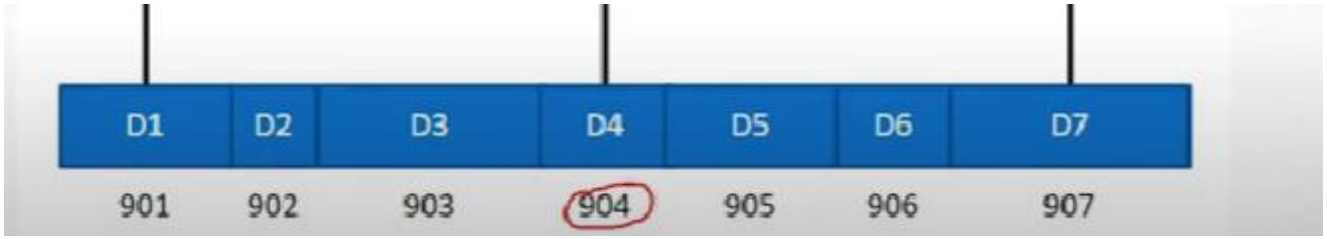
2. Direct Access:

a. Individual blocks have unique address.
b. Access is accomplish by direct access.
c. Example: Magnetic Disk



3.Random Access

- Individual address referred to memory location.
- **Access time independent of location of data or previous access.**
- **Example: Ram**



4. Associative

- Accessed by the content of the data rather than address.
- **Access time independent of location of data or previous access.**
- Example : Cache

5. Performance:

- **Access time (latency):** *It is a time that read and write operation takes.*
- **Memory cycle time:** Cycle time is access time + recovery time. This time related to system bus.
- **Transfer rate:** This is the rate at which data can be transferred into or out of a memory unit.

$$T_n = T_A + \frac{n}{R}$$

where

Tn = Average time to read or write *n* bits

TA = Average access time

n = Number of bits

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

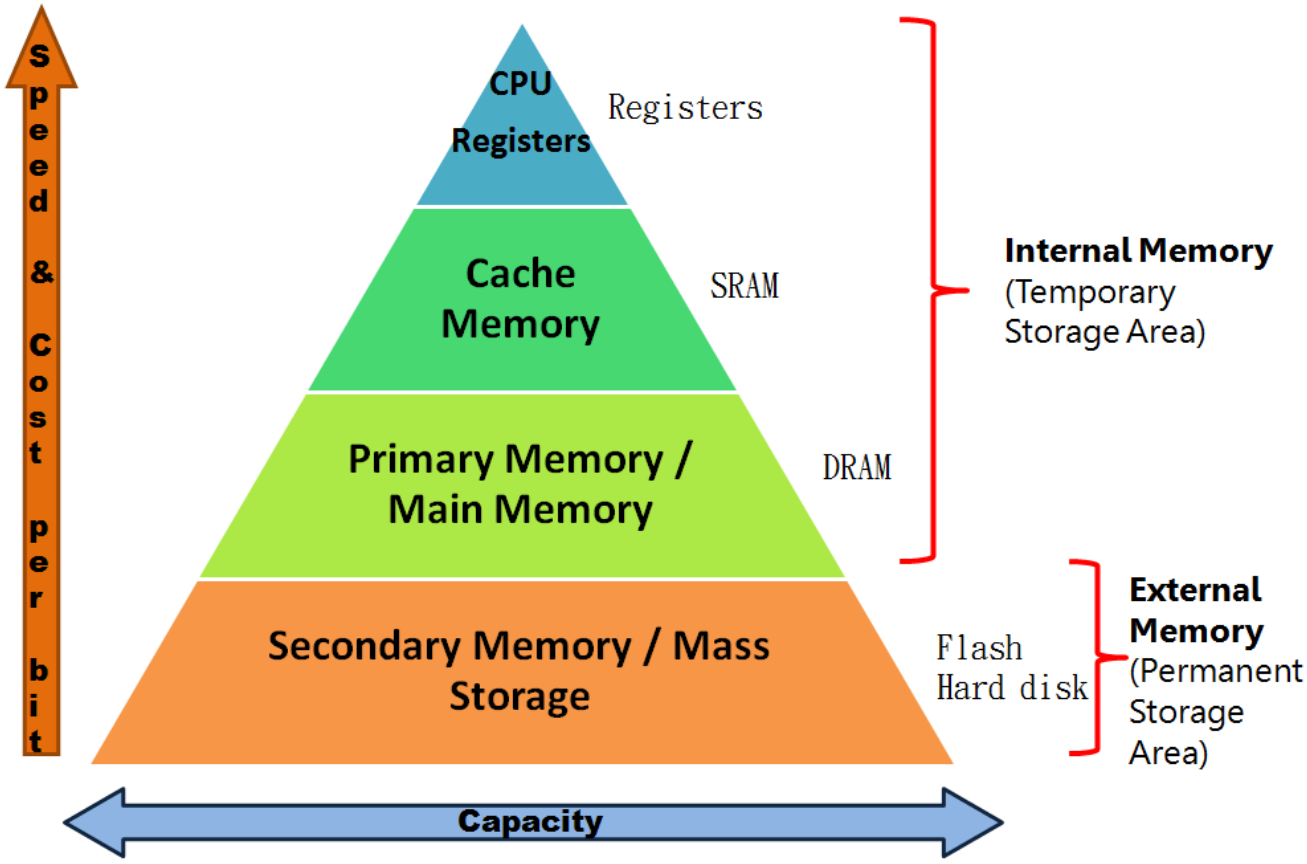
6. Physical types of memory have been employed. The most common today are semiconductor memory, magnetic surface memory, used for disk and tape, and optical and magneto-optical.

5. Physical characteristics of data storage:

1. In a volatile memory, information decays naturally or is lost when electrical power is switched off.
2. In a nonvolatile memory, information once recorded remains without deterioration until deliberately changed; no electrical power is needed to retain information. Magnetic-Surface memories are non-volatile.

8. Organization:*organization* refers to the physical arrangement of bits to form words.

Memory Hierarchy



Locality of Reference

- Locality of reference refers to a phenomenon in which a computer program tends to access same set of memory locations for a particular time period.
- In other words, Locality of Reference refers to the tendency of the computer program to access instructions whose addresses are near one another.

The property of locality of reference is mainly shown by:

1. Loops in program cause the CPU to repeatedly execute a set of instructions that constitute the loop.
2. Subroutine calls, cause the set of instructions are fetched from memory each time the subroutine gets called.
3. References to data items also get localized, meaning the same data item is referenced again and again.

Cache Operation:

There are two ways with which data or instruction is fetched from main memory and get stored in cache memory. These two ways are the following:

1. **Temporal Locality –**
   Temporal locality means current data or instruction that is being fetched may be needed soon. So we should store that data or instruction in the cache memory so that we can avoid again searching in main memory for the same data.
2. **Spatial Locality –**
   Spatial locality means instruction or data near to the current memory location that is being fetched, may be needed soon in the near future. This is slightly different from the temporal locality. Here we are talking about nearly located memory locations while in temporal locality we were talking about the actual memory location that was being fetched.

Main Memory Organization

- ROM
- As the name suggests, a **read-only memory (ROM)** contains a permanent pattern of data that cannot be changed.
- A ROM is nonvolatile; that is, no power source is required to maintain the bit values in memory.
- While it is possible to read a ROM, it is not possible to write new data into it.

- PROM
- PROM stands for **Programmable Read Only Memory**.
- It is a computer memory chip, and it is possible to program it once after creation.
- After programming the PROM, the information we write to it becomes permanent. Therefore, we cannot erase or delete that written data.

- EPROM:
- EPROM stands for **Erasable Programmable Read Only Memory**.
- We can erase and reprogram an EPROM without replacing it.
- It is possible to erase and write to it by exposing the memory chip to ultraviolet light.

- EEPROM:
- EEPROM stands for **Electrically Erasable Programmable Read-Only Memory**.
- It is a memory chip that we can erase and reprogram using electrical charge.
- It consists of a collection of floating gate transistors.
- The flash memory is a type of EEPROM which has a higher density and lower number of write cycles.

RAM

- The full form of RAM is random access memory.
- It is a memory device that is located on the motherboard of a computer and is used as the area of memory where the computer temporarily stores its work.
- RAM is volatile, which means that the contents of memory can be erased when electricity is removed from it. You can both read and write to RAM.

Features of RAM

1. RAM is the internal memory of a computer. It functions as the primary memory of a computer system. Various computations can be stored here, and items from the hard drive are loaded into RAM memory for easy access by the CPU of the computer system.
2. RAM is volatile. Even though non-volatile versions of RAM have been developed mostly commonly, the volatile version of the RAM is used. Volatile means the memory loses the data stored on it once the electricity is turned off.
3. This type of memory can be accessed directly without having to go through a sequence of memory locations. This random access makes RAM an expensive type of memory.
4. RAM is the fastest type of computer memory, hence it is the memory of choice for a computer's internal memory.
5. The functionality and speed of the computer dramatically depend on the RAM. If there is not sufficient RAM, then a computer will not be able to load and run the operating system quickly.

Types of RAM

1. **Static RAM**
2. **Dynamic RAM**

1. Static RAM:

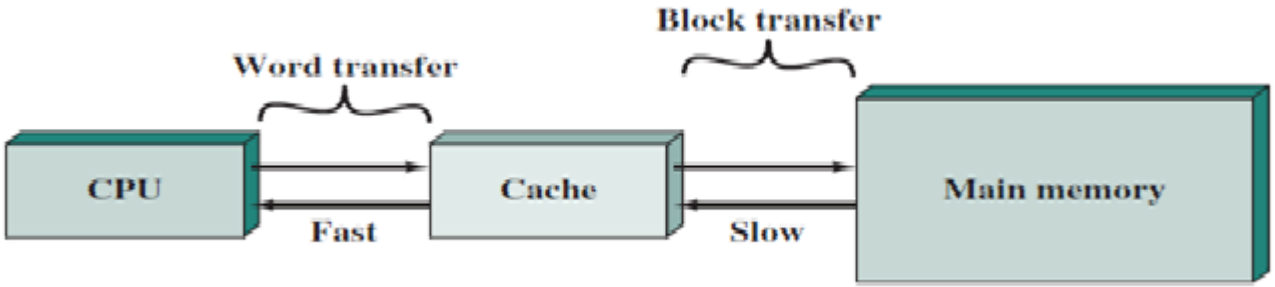
- Static RAM needs not to be refreshed continuously to retain a bit of information that is stored in it.
- It does not require any extra power to stop the leakage of power, so that makes it quicker than DRAM.
- One SRAM memory cell is made from six CMOS transistors.
- But there is a drawback. SRAM requires much more chips for the same amount of memory as DRAM (since it uses six transistors).
- It is costly and exists on the processors between the processor and main memory.

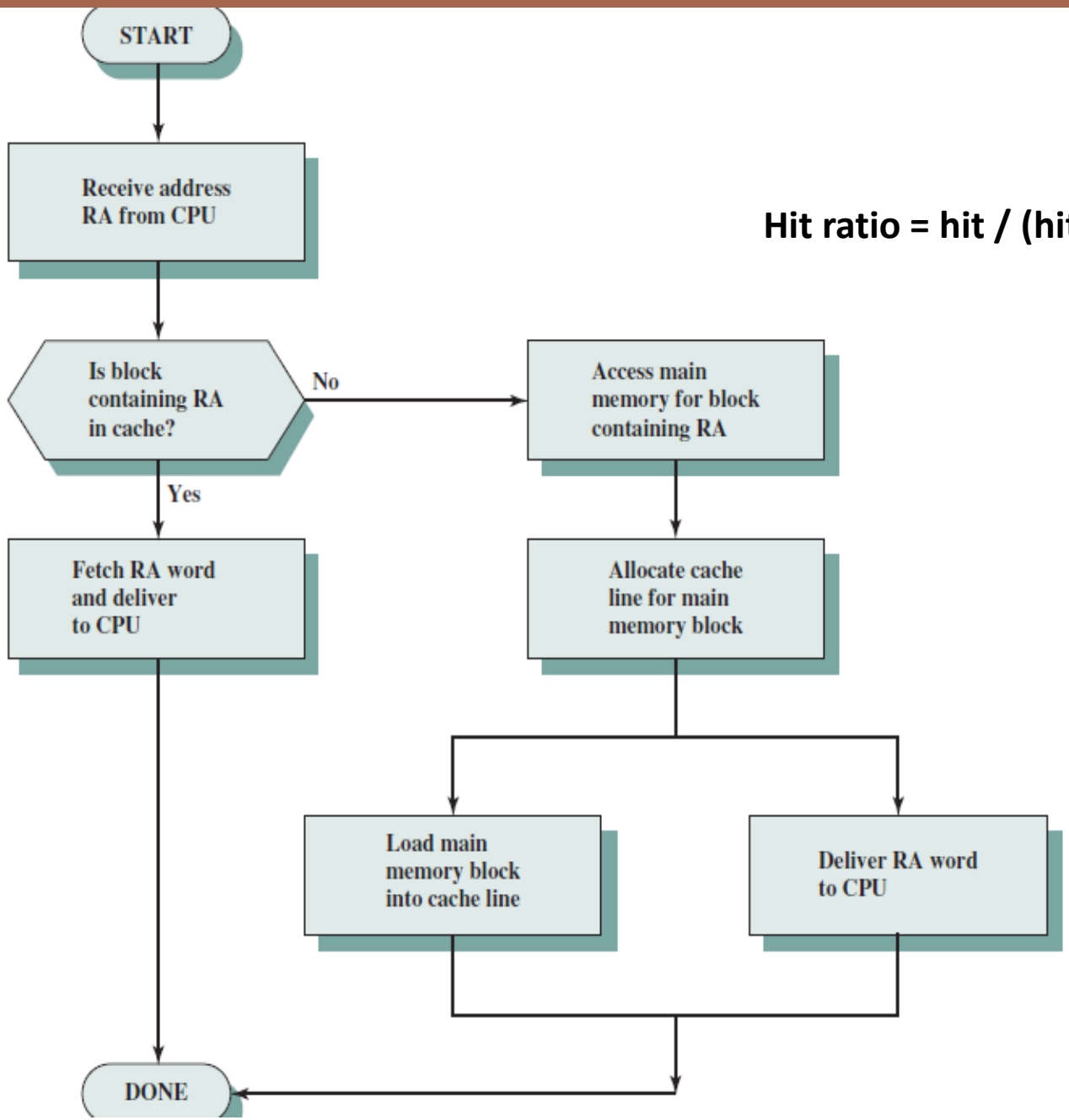
- Dynamic RAM:

1. Dynamic RAM is made from one transistor and one capacitor.
2. Since a capacitor is used, it needs to be refreshed from time to time to maintain the charge.
3. It is slower than SRAM because memory cells need to be continuously refreshed.
4. It consumes less power because the information is stored in one capacitor.
5. DRAM is less expensive than SRAM.
6. One memory cell is made up of one transistor and one capacitor so it occupies less space on the same-sized chip, providing you with more memory than an SRAM of similar size.

Cache Memory

- The data or contents of the main memory that are used frequently by CPU are stored in the cache memory so that the processor can easily access that data in a shorter time.
- Whenever the CPU needs to access memory, it first checks the cache memory. If the data is not found in cache memory, then the CPU moves into the main memory.
- Cache memory is placed between the CPU and the main memory.





Hit ratio = hit / (hit+miss)

Advantages and Disadvantages of Cache Memory

Advantages:

- It is faster than the main memory.
- The access time is quite less in comparison to the main memory.
- The speed of accessing data increases hence, the CPU works faster.
- Moreover, the performance of the CPU also becomes better.
- The recent data stores in the cache and therefore, the outputs are faster.

Disadvantages:

- It is quite expensive.
- The storage capacity is limited.

Cache Size

- There are several other motivations for minimizing cache size. The larger the cache, the larger the number of gates involved in addressing the cache. The result is that large caches tend to be slightly slower than small ones— even when built with the same integrated circuit technology and put in the same place on chip and circuit board.

Mapping Function

- Because there are fewer cache lines than main memory blocks, an algorithm is needed for mapping main memory blocks into cache lines. Further, a means is needed for determining which main memory block currently occupies a cache line.
- The choice of the mapping function dictates how the cache is organized.

- Three techniques can be used:

1. Direct
2. Associative
3. Set-associative.

Direct Mapping

- The simplest technique, known as direct mapping, maps each block of main memory into only one possible cache line. The mapping is expressed as

$$i = j \text{ modulo } m$$

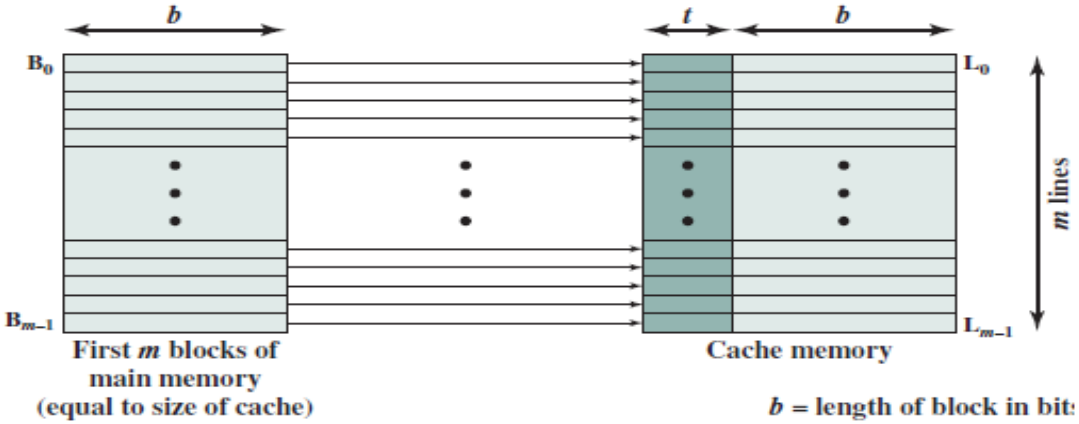
where

i = cache line number

j = main memory block number

m = number of lines in the cache

- Figure shows the mapping for the first m blocks of main memory. Each block of main memory maps into one unique line of the cache. The next m blocks of main memory map into the cache in the same fashion; that is, block Bm of main memory maps into line L0 of cache, block Bm+1 maps into line L1, and so on.



(a) Direct mapping

Example: Let Given 32 words in Main Memory and 16 words in cache memory. One block size is 4 words find Corresponding P.A bits split & Tag directory size.

Step1 : Identify Number of block in Main Memory=(M.M size/Block size)

i.e 32/4=8 memory block in M.M.

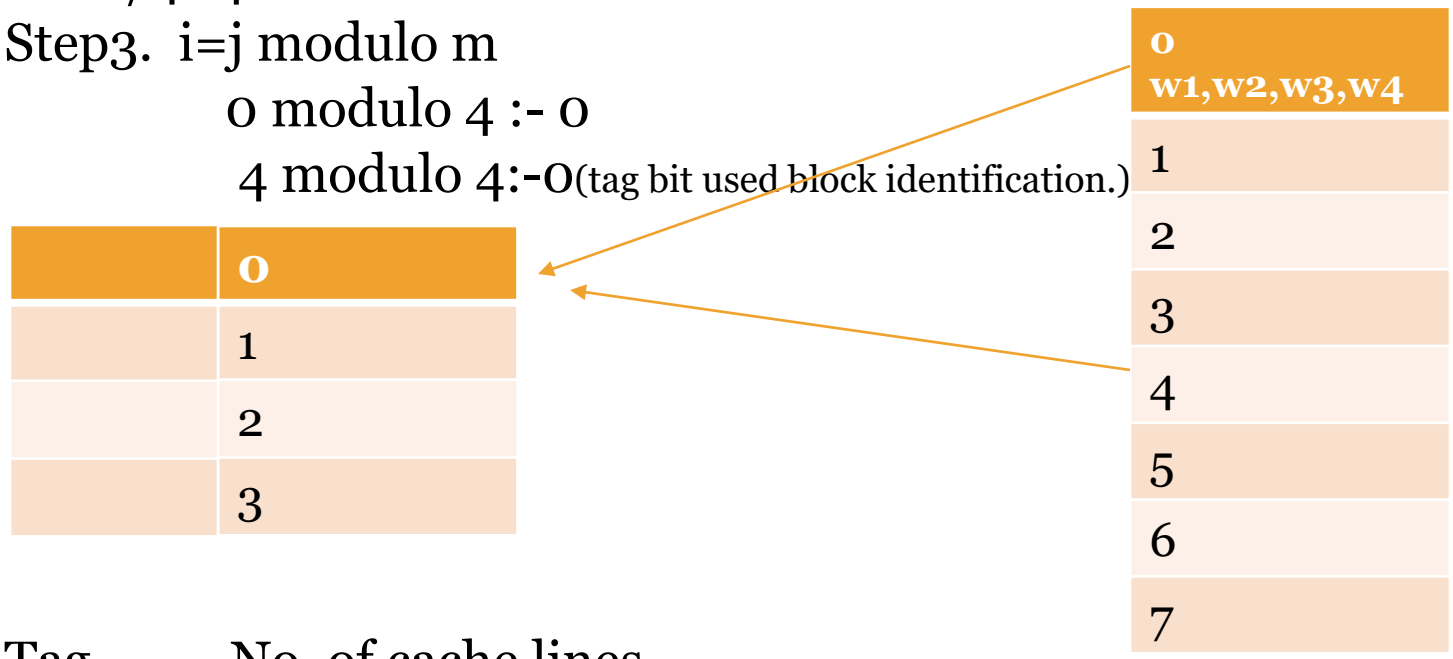
Step2: Identify Number of lines in cache.=(C.M. Size/Block Size)

i.e 16/4=4 block in C.M.

Step3. i=j modulo m

0 modulo 4 :- 0

4 modulo 4:-0(tag bit used block identification.)



Tag No. of cache lines

No. of Memory Block

Step 4: Physical Address:

TAG	Index/Cache Block offset	Word/Word Offset

M.M size: 32 words

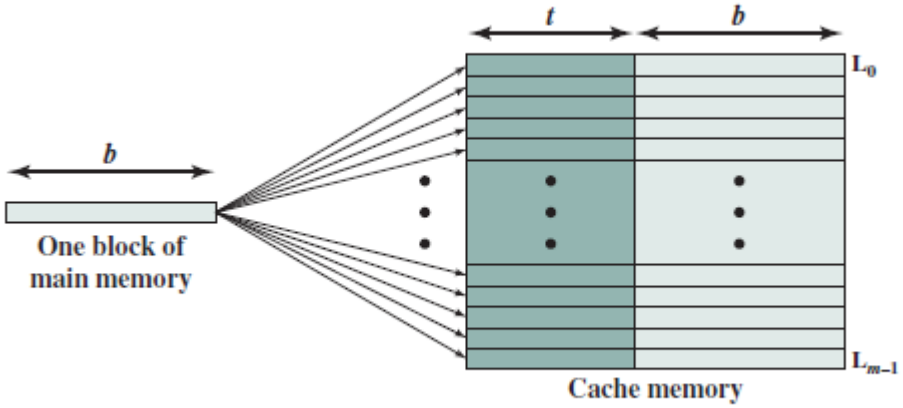
2^5 : 5 Bit address

No. of bit for PA:=log2(Size of M.M)
=log2(32)
=5 bits
Word= log2 (Block Size)
log2(4): 2
Index= log2(No. of Cache lines)
log2(4):2

Physical Address is of 5 bit length
Index is 2 bit
Word is 2 bit.
Tag is 1 bit.(5-2-2)
Tag:=log2(m/n)
m: Number of M.M block
n: Number of CM block
Tag Directory size= No. of lines in cache * No. of bits in tag
=4*1=4 .

Associative Mapping

- In this type of mapping, the associative memory is used to store content and addresses of the memory word.
- Any block can go into any line of the cache.
- This means that the word id bits are used to identify which word in the block is needed, but the tag becomes all of the remaining bits.
- This enables the placement of any word at any place in the cache memory.
- It is considered to be the fastest and the most flexible mapping form.



(b) Associative mapping

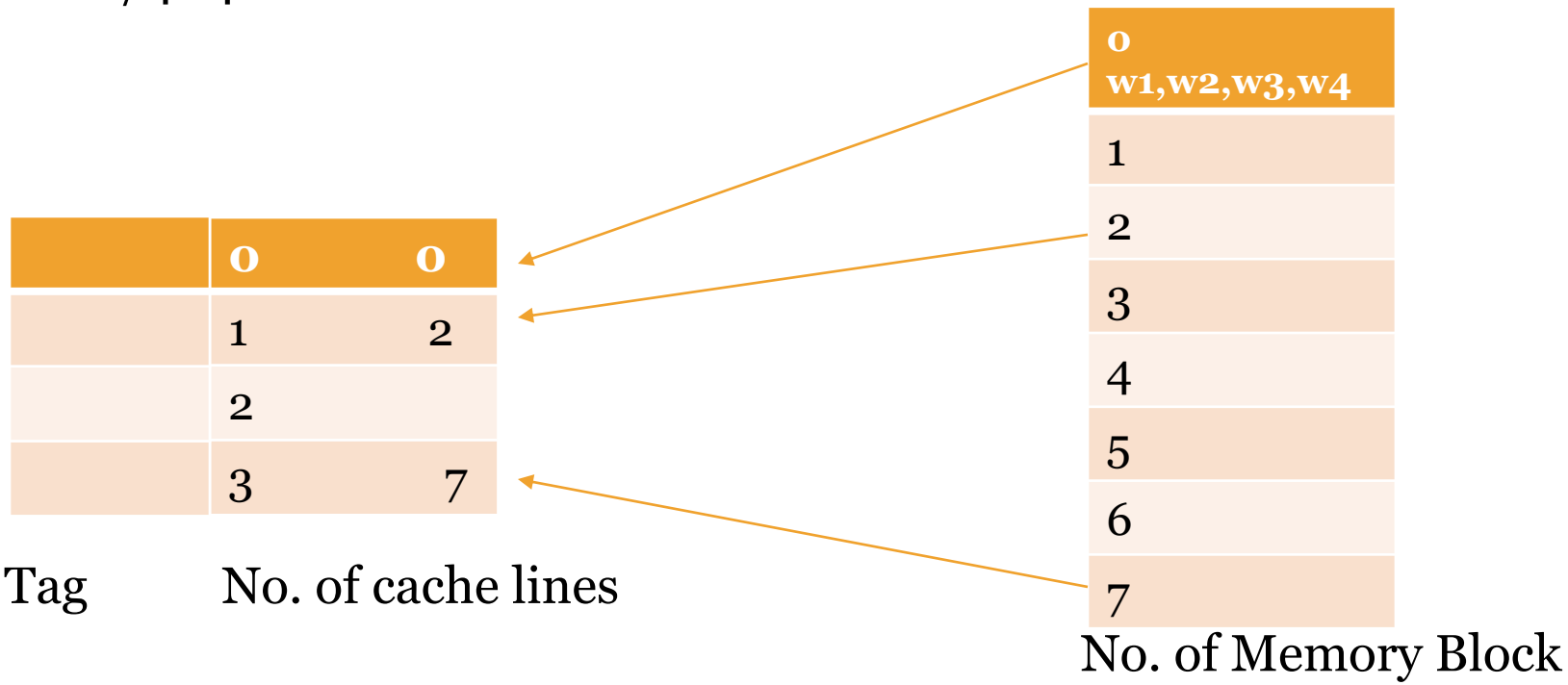
Example: Let Given 32 words in Main Memory and 16 words in cache memory. One block size is 4 words find Corresponding physical address split and Tag directory size.

Step1 : Identify Number of block in Main Memory=(M.M.Size/Block size)

i.e 32/4=8 memory block in M.M.

Step2: Identify Number of lines in cache.=(C.M Size/Block size)

i.e 16/4=4 block in C.M.



Step 3: Physical Address:

TAG	Word

M.M size: 32 words

2^5 : 5 Bit address

No. of bit for PA:=log2(Size of M.M)
=log2(32)
=5 bits.

Word= log2 (Block size)
log2(4): 2

Tag=log2(No. of Blocks in M.M)
log2(8)
=3 Bit

Tag Directory size: = No. of Cache line * No. of Bits in tag
=4*3
=12

set-associative mapping

Set- associative mapping is a compromise that exhibits the strengths of both the direct and associative approaches while reducing their disadvantages. In this case, the cache consists of number sets, each of which consists of a number of lines. The relationships are

$$m = v * k$$

$$i = j \text{ modulo } v$$

where

i = cache set number

j = main memory block number

m = number of lines in the cache

v = number of sets

k = number of lines in each set

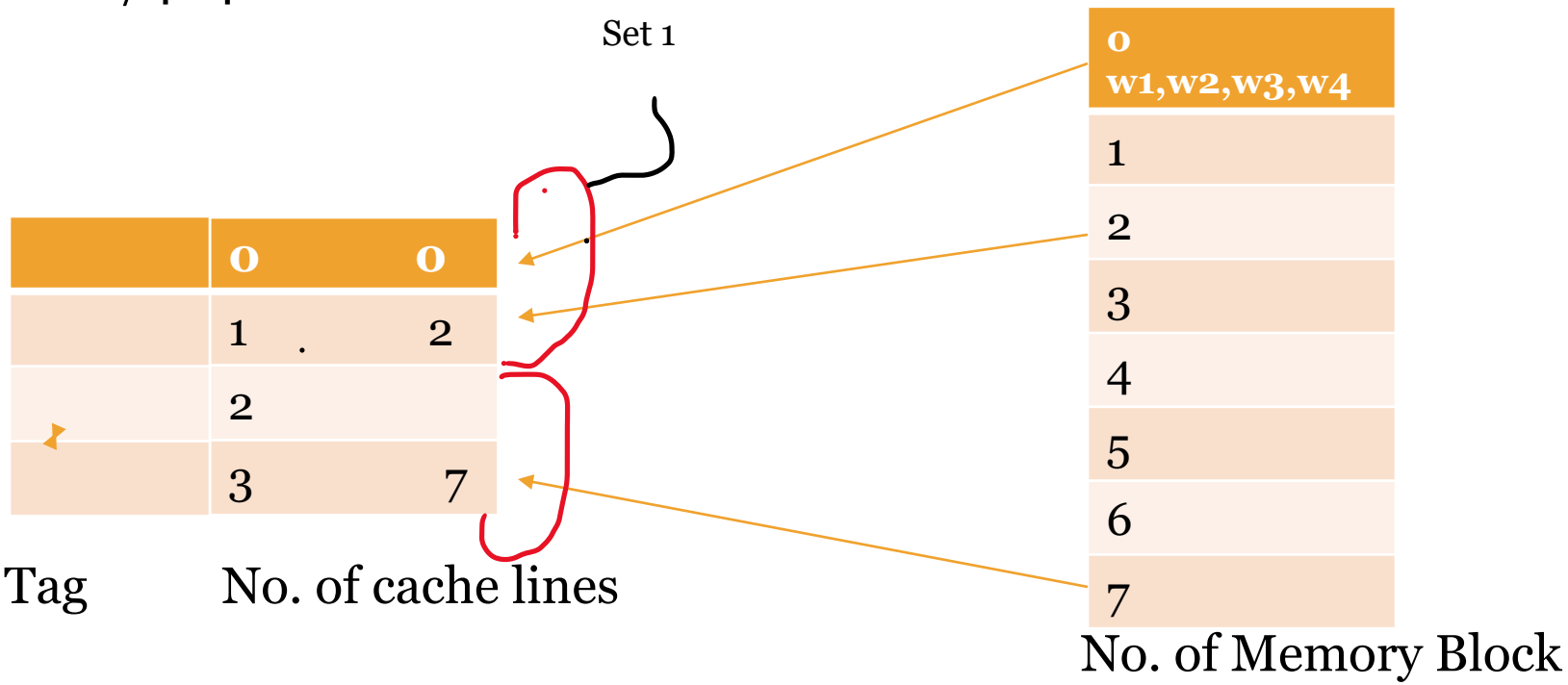
Example: Let Given 32 words in Main Memory and 16 words in cache memory. One block size is 4 words find Corresponding tag, set and word.(2 way set associative)

Step1 : Identify Number of block in Main Memory=(M.M.Size/Block size)

i.e 32/4=8 memory block in M.M.

Step2: Identify Number of lines in cache.=(C.M Size/Block size)

i.e 16/4=4 block in C.M.



Step 3: Physical Address:

Tag	Set Offset	Word Offset

M.M size: 32 words

2^5 : 5 Bit address

No. of bit for PA:=log2(Size of M.M)
=log2(32)
=5 bits.

Word= log2 (block size)
= log2(4)
=2

Set Offset:=log2((No.of. Cache line)/Set Associative Number i.e k)
=log2(4/2)
= log2(2)
=1

Tag= Total- Word – Set(5-2-1)=2

Virtual Memory

- *Virtual memory in computer organization architecture is a technique and not actually a memory in physical form present in computer system. This is the reason it is known as* ***virtual memory****.*
- Virtual Memory (VM) Concept is similar to the Concept of Cache Memory.
- While Cache solves the speed up requirements in memory access by CPU, Virtual Memory solves the Main Memory (MM) Capacity requirements with a mapping association to Secondary Memory i.e Hard Disk.
- Both Cache and Virtual Memory are based on the Principle of Locality of Reference.
- Virtual Memory provides an illusion of unlimited memory being available to the Processes/ Programmers.