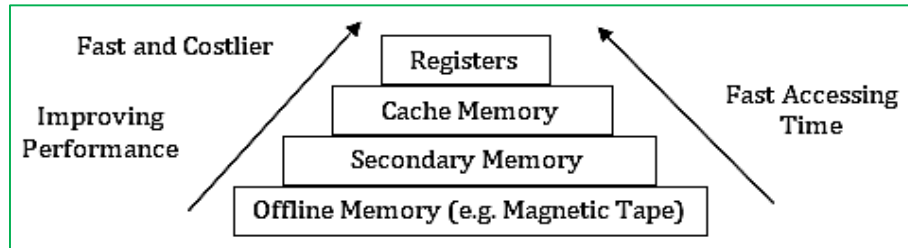


## CHAPTER – 6

### MEMORY ORGANIZATION

#### MEMORY HIERARCHY:



*Fig: Memory Hierarchy*

To achieve greatest performance the memory must be able to keep up with the processor. The designer would like to use memory technologies that provide larger capacity memory because capacity is needed for better performance of the system.

A typical hierarchy is shown in figure above. As we go down to the hierarchy the following features occurs:

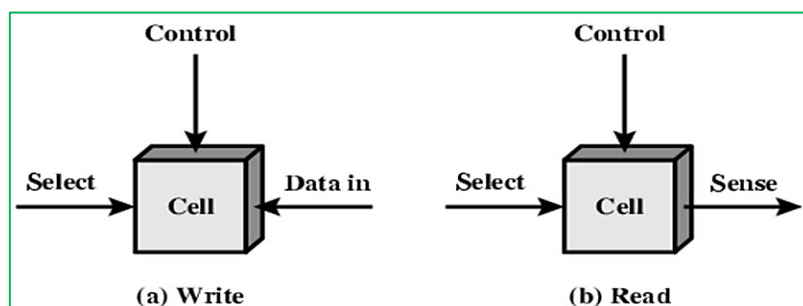
- a. Decreasing cost per bit.
- b. Increasing access time.
- c. Decreasing frequency of access of the memory by the processor.

Thus, smaller, more expensive, faster memory are supplemented by larger, cheaper and slower memory. The key to the success of this organization is decreasing the frequency of access.

#### SEMICONDUCTOR MAIN MEMORY:

The basic element of semiconductor memory is the memory cell. Although a variety of electronic technologies are used, all semiconductor memory cells share certain properties:

- a. They exhibit two stable (or semistable) states, which can be used to represent binary 1 and 0.
- b. They are capable of being written into (at least once), to set the state.
- c. They are capable of being read to sense the state.



*Fig: Memory Cell Operation*

Figure above shows the operation of memory cell. Most commonly the cell has three functional terminals capable of carrying an electrical signal. The select terminal, selects a memory cell for

a read or write operation. The control terminal indicates read or write. For writing, the other terminal provides an electrical signal that sets the state of the cell to 0 or 1. For reading, that terminal is used for output of the cell state.

### **RAM (RANDOM ACCESS MEMORY):**

RAM stands for "Random Access Memory" and is often called primary/main memory because it is made up of semiconductor chips. It is the working space used by the computer to hold the program that is currently running along with the necessary data and instructions. It is fast and expensive memory which allows the computer to access the data and instructions very quickly.

We can read from RAM as well as write into it. Hence is also called "Read-Write" memory. The main drawback of RAM is that it is volatile memory so the contents of RAM are lost when the computer is switched off.

It is made of millions of microscopic cells which are distinctly numbered so that each cell can be identified and located. Each cell can be electrically charged or not. The charged cell represents 1 and not charged cell represents 0 in binary format. RAM is also of two types:

#### **1. DRAM:**

DRAM stands for "Dynamic Random Access Memory". It is made up of capacitors which is capable of storing the electric charge. Due to the leakage of charges, the capacitors discharge gradually and the memory cells lose their contents. So, to recharge the capacitors to retain its memory contents it has to be refreshed periodically. DRAM is slower than SRAM but it is dense, consumes less electricity, smaller in size and less expensive.

Synchronous Dynamic Random Access Memory "SDRAM" is DRAM that has a synchronous interface which is widely used in present computers. Traditionally, DRAM has a synchronous interface, which means that it control inputs. SDRAM has a synchronous interface, meaning that it waits for a clock signal before synchronized with the computers system bus.

Example: DDR "Dual Data Rate", DDR2, DDR3, EDO DRAM, SDRAM, RIMM, etc.

#### **2. SRAM:**

SRAM stands for "Static Random Access Memory" and is made up of transistors. It is called static because it can remember or retain its memory contents without being refreshed or recharged as long as there is power. SRAM is faster than DRAM but more expensive, loser in density and bigger in size and consumes more electricity.

### **ROM (READ ONLY MEMORY):**

ROM stands for "Read Only Memory" and it is called ROM because only read operation can be performed on it. The binary information stored in ROM is written permanently by the manufacturer and it cannot be altered. ROM is necessary to store such software which enables the computer to boot up because booting instructions does not need modification.

ROM is non-volatile memory because it can retain its contents even after the computer is turned off. It is also made semiconductor chips. The program stored permanently in ROM is called firmware. Hence, firmware is immediately available when a device is powered on to start up the PC or other electronic equipment like mobile, PDA and others. ROM is of three types:

## 1. PROM:

PROM stands for "Programmable Read Only Memory". Initially it is the blank chip which can be written or programmed only one time by using a special machine called ROM programmer or ROM burner. Once the PROM is written, it cannot be modified and becomes ROM.

## 2. EPROM:

EPROM stands for "Erasable Programmable Read Only Memory". It is a special chip which can be re-programmed to record different information. The data and information are erased by exposing it to intense ultra violet light for about 20 minutes. These types of memory are used in product development and experimental projects.

## 3. EEPROM:

EEPROM stands for "Electrically Erasable Programmable Read Only Memory". These types of chips can be erased and re-programmed repeatedly with special electrical pulses. It does not require a special device to write into it. EEPROM can be re-programmed without removing it from the computer. It also has limited life span i.e. the number of times it can be re-programmed is limited to tens or hundreds or thousands of times.

### DIFFERENTIATE BETWEEN RAM AND ROM:

RAM	ROM
+ RAM stands for Random Access Memory.	+ ROM stands for Read Only Memory.
+ RAM is volatile memory, if power fails data and information will be lost.	+ ROM is inherently non-volatile memory, if power fails data and information will not be lost.
+ RAM is used for currently running programs of computer system.	+ ROM is used to store firmware of computer system and system software for embedded system.
+ RAM is read/write memory.	+ ROM is read only memory.
+ The cost of RAM is higher than ROM.	+ The cost of ROM is lower is than RAM.
+ There are two types of RAM: SRAM and DRAM.	+ There are three types of ROM: PROM, EPROM and EEPROM

### DIFFERENTIATE BETWEEN DRAM AND SRAM:

DRAM	SRAM
+ DRAM stands for Dynamic Random Access Memory.	+ SRAM stands for Static Random Access Memory.
+ DRAM is made up of capacitors.	+ SRAM is made up of transistors.
+ DRAM is high density RAM. In one chip larger memory can be constructed.	+ SRAM is low density of RAM. In one chip small memory can be constructed.
+ Power consumption of DRAM is higher than SRAM.	+ Power consumption of SRAM is lesser than DRAM.
+ DRAM need to be periodically refreshed. (System automatically refreshed the RAM cells.)	+ SRAM does not need to be periodically refreshed.
+ The cost of DRAM is lower than SRAM.	+ The cost of SRAM is higher than DRAM.

+ The data access time is larger than SRAM, typically requires larger than 40 nanoseconds. Hence they are slow.	+ The data access time is smaller than DRAM, typically less than 30 nanoseconds. Hence they are fast.
+ DRAM is generally used for low cost high capacity memory for computers.	+ SRAM is generally used to create memory of critical section like cache memory.
+ Example: DDR, DDR2, DDR3 (Dual Data Rate), EDO DRAM, SDRAM, RIMM, etc.	+ Example: cache memory of microprocessor.

### CHARACTERISTICS OF MEMORY SYSTEM:

#### **1. LOCATION:**

Internal (Example: Registers, Main Memory, and Cache)  
External (Example: Magnetic Disk, Optical Disk, etc.)

#### **2. CAPACITY:**

Number of words  
Number of bytes

#### **3. UNITS OF TRANSFER:**

Word  
Block

#### **4. ACCESS METHOD:**

Sequential, Direct  
Random, Associative

#### **5. PERFORMANCE:**

Access Time  
Cycle Time  
Transfer Rate

#### **6. PHYSICAL CHARACTERISTICS:**

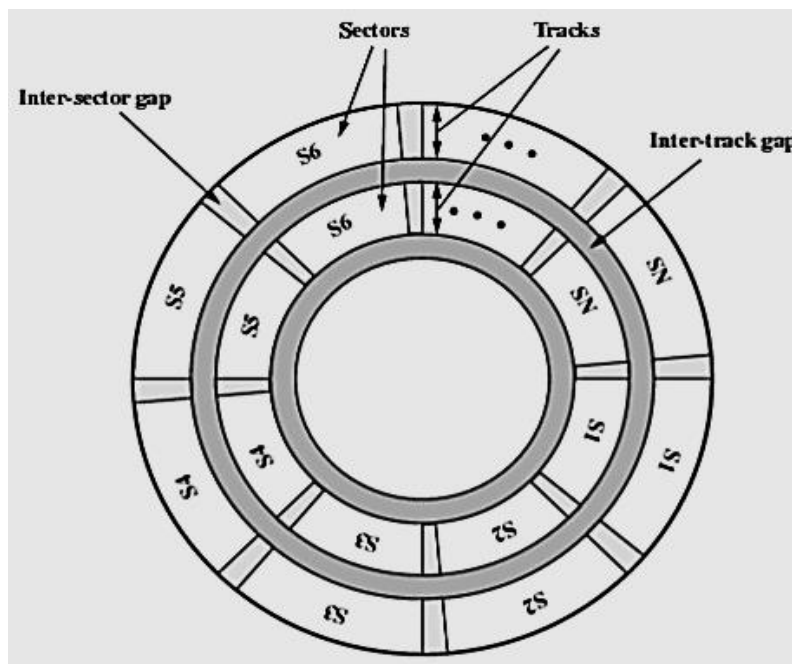
Volatile or Non-volatile  
Erasable or Non-erasable

### AUXILIARY MEMORY:

#### **1. MAGNETIC DISK:**

A disk is a circular platter constructed of metal or of plastic coated with magnetic materials. Data are recorded on and later retrieved from the disk through a conducting coil known as head. During a read or write operations the head is stationary while the platter rotates it. Writing is achieved by producing a magnetic field which records a magnetic pattern on magnetic surface.

The figure below shows the data layout of disk. The head is capable of reading or writing. The data is transferred to and from the disk in blocks. The sectors may be of fixed or variable length. The platter in a disk may be of single or multiple.

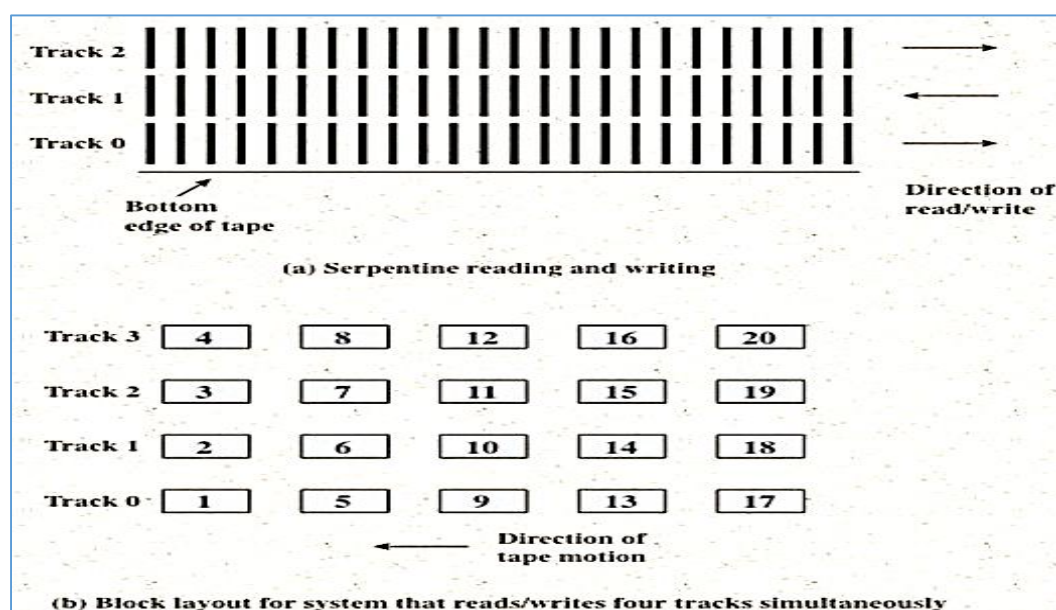


*Fig: Disk Data Layout*

## 2. MAGNETIC TAPE:

Tape systems use same reading and recording techniques as disk systems. The way of recording the data is parallel. It stores one bytes of data at a time. Data on the tape are structured as a number of parallel tracks running lengthwise.

The data are read and written in contiguous track called physical records. Blocks on the tape are separated by gaps called interrecord gaps. It is a system for storing digital information on tape using digital recording. The device that performs writing or reading of a data is a tape drive.



*Fig: Magnetic Tape Features*

### **3. OPTICAL DISK:**

An optical disc is an electronic data storage medium that can be written to and read using a low-powered laser beam. Originally developed in the late 1960s, the first optical disc, created by James T. Russell, stored data as micron-wide dots of light and dark. A laser read the dots, and the data was converted to an electrical signal, and finally to audio or visual output. However, the technology didn't appear in the marketplace until Philips and Sony came out with the compact disc (CD) in 1982. Since then, there has been a constant succession of optical disc formats, first in CD formats, followed by a number of DVD formats.

Optical disc offers a number of advantages over magnetic storage media. An optical disc holds much more data. The greater control and focus possible with laser beams (in comparison to tiny magnetic heads) means that more data can be written into a smaller space. Storage capacity increases with each new generation of optical media. Emerging standards, such as Blu-ray, offer up to 27 gigabytes (GB) on a single-sided 12-centimeter disc. In comparison, a diskette, for example, can hold 1.44 megabytes (MB). Optical discs are inexpensive to manufacture and data stored on them is relatively impervious to most environmental threats, such as power surges, or magnetic disturbances.

### **4. FLASH DRIVES:**

A small, portable flash memory card that plugs into a computer's USB port and functions as a portable hard drive. USB flash drives are touted as being easy-to-use as they are small enough to be carried in a pocket and can plug into any computer with a USB drive. USB flash drives have less storage capacity than an external hard drive, but they are smaller and more durable because they do not contain any internal moving parts.





USB flash drives also are called thumb drives, jump drives, pen drives, key drives, tokens, or simply USB drives.

A flash drive consists of a small printed circuit board carrying the circuit elements and a USB connector, insulated electrically and protected inside a plastic, metal, or rubberized case which can be carried in a pocket or on a key chain. Most flash drives use a standard type-A USB connection allowing connection with a port on a personal computer, but drives for other interfaces also exist.

### **5. REVIEW OF RAID (REDUNDANT ARRAY OF INDEPENDENT DISKS):**

RAID is a set of physical disk drives viewed by the operating system as a single logical drive. Data are distributed across the physical drives of an array in a scheme known as striping. Redundant disk capacity is used to store parity information which guarantees data recoverability in case of disk failure. There are different levels of RAID and they are:

#### **a. RAID 0:**

-  Often called striping
-  Break a file into blocks of data
-  Simple to implement
-  Provides no redundancy or error detection

#### **b. RAID 1:**

-  Complete file is stored in a single disk

- ✚ A second disk contains an exact copy of file
- ✚ Provides complete redundancy of data

**c. RAID 2:**

- ✚ Strips data across disk similar to level – 0
- ✚ A parity disk is used to reconstruct corrupted or lost data.
- ✚ Uses Error Checking Code (ECC) to monitor correctness of information

**d. RAID 3:**

- ✚ It eliminates the problem of level – 2 that is disk need to detect which disk has an error.
- ✚ Modern disk can already determine if there is an error.

**e. RAID 4:**

- ✚ It consists of a block level striping with dedicated parity.
- ✚ It allows multiple small input/output to be done at once.

**f. RAID 5:**

- ✚ It consists of block level striping with distributed parity.
- ✚ Here, also parity information is distributed among the drives.

**g. RAID 6:**

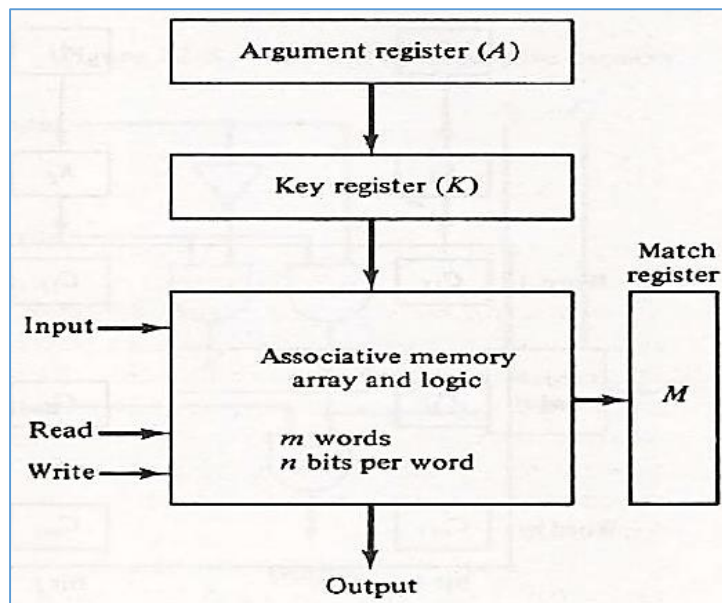
- ✚ Consists of block level striping with double distributed parity.
- ✚ Double parity provides fault tolerance upto two failed drives.

## ASSOCIATIVE MEMORY:

### **1. HARDWARE ORGANIZATION:**

The time required to find an item stored in memory can be reduced considerably if stored data can be identified for access by the content of the data itself rather than by address. A memory unit accessed by the content is called Associative Memory or Content Accessible Memory. This type of memory is accessed simultaneously and parallel on the basis of data content rather than by specific address or location. When a word is written in associative memory no address is given.





*Fig: Block Diagram of Associative Memory*

Associative Memory is organized in such a way:

- a. **Argument Register (A):** It contains the word to be searched. It has 'n' bits (one for each bit of the word).
- b. **Key Register (K):** This specifies which part of the argument word needs to be compared with words in memory. If all bits in register are 1, the entire word should be compared. Otherwise, only the bits having K-bits set to 1 will be compared.
- c. **Associative Memory Array:** It contains the words which are to be compared with the argument word.
- d. **Match Register (M):** It has 'm' bits, one bit corresponding to each word in the memory array. After the matching process, the bits corresponding to matching words in match register are set to 1.

## 2. ADDRESS MATCHING LOGIC:

Key register provide the mask for choosing the particular field in A register. The entire content of A register is compared if key register content all 1. Otherwise, only bit that have 1 in key register are compared. If the compared data is matched corresponding bits in the match register are set. Reading is accomplished by sequential access in memory for those words whose bit are set. Example:

A	101 111100	
K	111 000000	
Word 1	100 111100	no match
Word 2	101 000001	match

Let us include key register. If  $K_j = 0$  then there is no need to compare  $A_j$  and  $F_{ij}$ . Only when  $k_j = 1$ , comparison is needed. This achieved by ORing each term with  $K_j$ .  $M_i = (X_1 + K'_1) (X_2 + K'_2) (X_3 + K'_3) \dots (X_n + K'_n)$



### 3. READ/WRITE OPERATIONS:

#### Read Operations:

When a word is to be read from an associative memory, the contents of the word, or a part of the word is specified. If more than one word match with the content, all the matched words will have 1 in the corresponding bit position in match register. Matched words are then read in sequence by applying a read signal to each word line. In most application, the associative memory stores a table with no two identical items under a given key.

#### Write Operations:

If the entire memory is loaded with new information at once prior to search operation then writing can be done by addressing each location in sequence. Tag register contain as many bits as there are words in memory. It contains 1 for active word and 0 for inactive word. If the word is to be inserted, tag register is scanned until 0 is found and word is written at that position and bit is change to 1.

### 4. TYPES OF ASSOCIATIVE MEMORY:

There are two types of Associative Memory, which both are used in different conditions.

#### a. Auto-Associative:

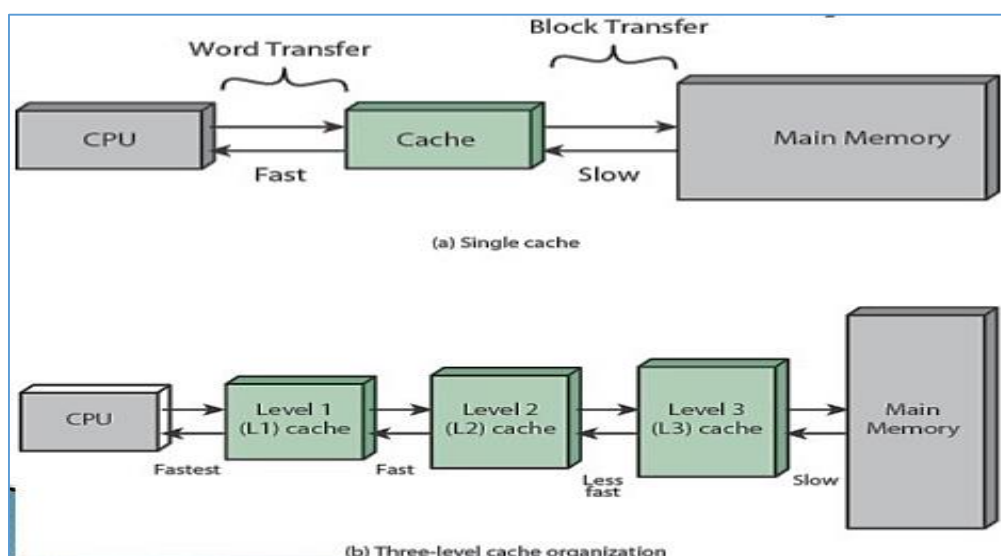
Auto-associative memory takes back (retrieves) a previously stored pattern that most closely resembles the current pattern.

#### b. Hetero-Associative:

Hetero-associative memory, the retrieved pattern is in general, different from the input pattern not only in content but possibly also in type and format. Neutral network are used to implement these associative memory models called NAM (Neutral Associative Memory).

### CACHE MEMORY:

#### 1. CACHE INITIALIZATION:



The cache contains a copy of the portion main memory. When the processor attempts to read a word of memory a check is made to determine if the word is in the cache. If so, the word is delivered to the processor if not block of main memory is read and transfer to the cache and is delivered to the processor.

Because of the phenomena of locality of reference, when block of data is fetched into a cache it is likely that there will be a future reference to the same memory location. As shown in figure above a block of data is transferred between cache and main memory where as word of data is transferred between CPU and Cache.

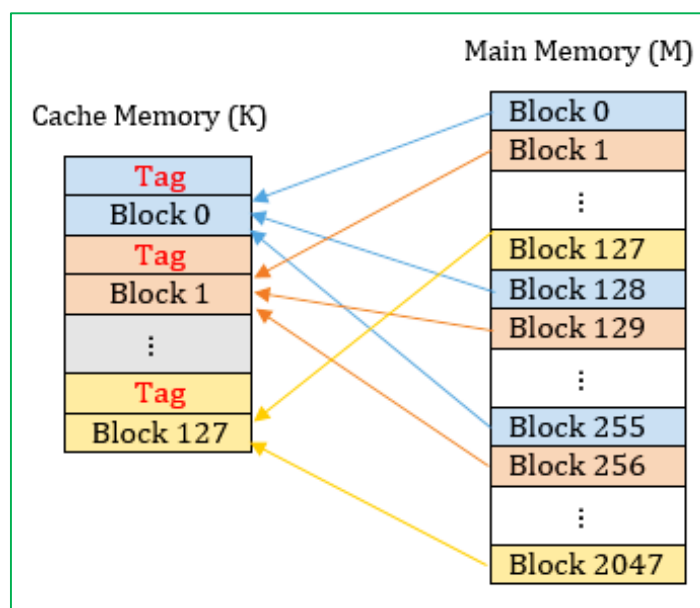
Different levels of caches can be created on the basis of uses. For example on the basis of speed level 1 cache is faster than level 2 and level 2 is faster than level 3.

The performance of cache memory is measured in terms of quantity called hit ratio. **When the CPU refers to memory and finds the word in cache it is said to produce hit. If the word is not found in cache it counts a miss. The ratio of number of hits divided by total CPU references to memory is called hit ratio.**

## 2. MAPPING CACHE MEMORY:

Transformation of data from main memory to cache memory is known as mapping process. There are 3 types of mapping process and they are:

### a. Direct Mapping:

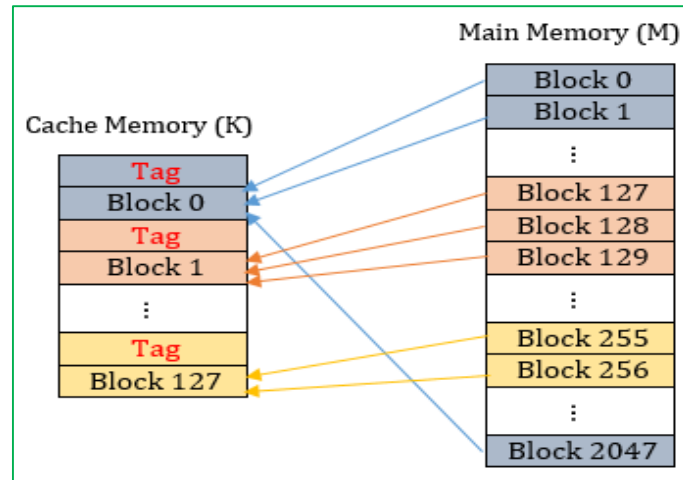


This is the simplest mapping technique in which block 'M' of main memory is mapped into block 'K' of cache memory. Since, more than one main memory block is mapped into a given cache block position contention may arise for that position even if the cache is not full.

A main memory address can be divided into 3 fields i.e. tag, block and word. The tag bit is required to identify a main memory block when it is resident in cache. When a new block enters the cache, the cache block field determines the cache position. The main memory address can be divided into 3 fields are:

Main Memory Address		
5	7	4
Tag	Block	Word

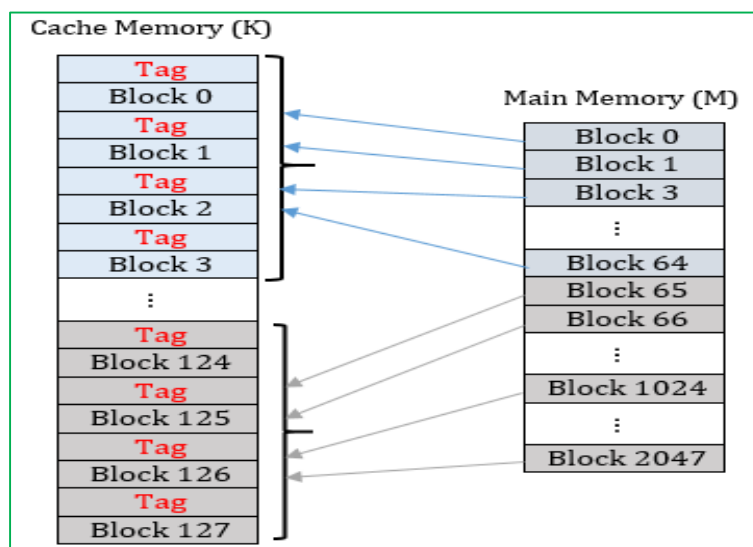
**b. Associative Mapping:**



This is much more flexible mapping technique, here any main memory block can be loaded to any cache block position. In this case, 12 tag bits are required to identify main memory block. The tag bits of an address received from CPU are compared with the tag bits of each cache blocks to see if the desire block is present in the cache. The main memory address can be divided into 2 fields are:

Main Memory Address	
12	4
Tag	Block

**c. Block Set Associative Mapping:**



In block set associative mapping technique, blocks of the cache are grouped into sets and the mapping allows a block of main memory to reside in any block of particular set. As shown in diagram above a cache with four block per set is used for mapping technique.

The six bit set field of the cache might contain the address block. As shown in diagram above two kilobyte of main memory is used to transfer its data to cache memory. The contention problem of the direct method is overcome by having few choices for block replacement.

Similarly, the hardware cost is reduced by reducing the size of associative search which are its advantages. The main memory address can be divided into 3 fields are:

Main Memory Address		
6	6	4
Tag	Block	Word

### 3. WRITE POLICY:

When a system writes data to cache, it must at some point write that data to the backing store as well. The timing of this write is controlled by what is known as the **write policy**.

There are two basic writing approaches:

- a. **Write-through:** write is done synchronously both to the cache and to the backing store.
- b. **Write-back (also called write-behind):** initially, writing is done only to the cache. The write to the backing store is postponed until the cache blocks containing the data are about to be modified/replaced by new content.

A write-back cache is more complex to implement, since it needs to track which of its locations have been written over, and mark them as dirty for later writing to the backing store. The data in these locations are written back to the backing store only when they are evicted from the cache, an effect referred to as a lazy write. For this reason, a read miss in a write-back cache (which requires a block to be replaced by another) will often require two memory accesses to service: one to write the replaced data from the cache back to the store, and then one to retrieve the needed data.

Other policies may also trigger data write-back. The client may make many changes to data in the cache, and then explicitly notify the cache to write back the data. No data is returned on write operations, thus there are two approaches for situations of write-misses:

- a. **Write Allocate (Also Called Fetch on Write):** data at the missed-write location is loaded to cache, followed by a write-hit operation. In this approach, write misses are similar to read misses.
- b. **No-Write Allocate (Also Called Write-No-Allocate or Write Around):** data at the missed-write location is not loaded to cache, and is written directly to the backing store. In this approach, only the reads are being cached.

Both write-through and write-back policies can use either of these write-miss policies, but usually they are paired in this way:

A write-back cache uses write allocate, hoping for subsequent writes (or even reads) to the same location, which is now cached. A write-through cache uses no-write allocate. Here, subsequent writes have no advantage, since they still need to be written directly to the backing store.

Entities other than the cache may change the data in the backing store, in which case the copy in the cache may become out-of-date or stale. Alternatively, when the client updates the data in the

cache, copies of those data in other caches will become stale. Communication protocols between the cache managers which keep the data consistent are known as coherency protocols.

#### **4. REPLACEMENT ALGORITHMS:**

A cache algorithm is a detailed list of instructions that directs which items should be discarded in a computing device's cache of information. Examples of cache algorithms include:

##### **a. First In First Out (FIFO):**

Using this algorithm the cache behaves in the same way as a FIFO queue. The cache evicts the first block accessed first without any regard to how often or how many times it was accessed before.

##### **b. Last In First Out (LIFO):**

Using this algorithm the cache behaves in the exact opposite way as a FIFO queue. The cache evicts the block accessed most recently first without any regard to how often or how many times it was accessed before.

##### **c. Least Frequently Used (LFU):**

This cache algorithm uses a counter to keep track of how often an entry is accessed. With the LFU cache algorithm, the entry with the lowest count is removed first. This method isn't used that often, as it does not account for an item that had an initially high access rate and then was not accessed for a long time.

##### **d. Least Recently Used (LRU):**

This cache algorithm keeps recently used items near the top of cache. Whenever a new item is accessed, the LRU places it at the top of the cache. When the cache limit has been reached, items that have been accessed less recently will be removed starting from the bottom of the cache. This can be an expensive algorithm to use, as it needs to keep "age bits" that show exactly when the item was accessed. In addition, when a LRU cache algorithm deletes an item, the "age bit" changes on all the other items.

##### **e. Adaptive Replacement Cache (ARC):**

Developed at the IBM Almaden Research Center, this cache algorithm keeps track of both LFU and LRU, as well as evicted cache entries to get the best use out of the available cache.

##### **f. Most Recently Used (MRU):**

This cache algorithm removes the most recently used items first. A MRU algorithm is good in situations in which the older an item is, the more likely it is to be accessed.

##### **g. Random Replacement (RR):**

Randomly selects a candidate item and discards it to make space when necessary. This algorithm does not require keeping any information about the access history. For its simplicity, it has been used in ARM processors. It admits efficient stochastic simulation.