Group-7 presentaton

ARAVIND
106119016

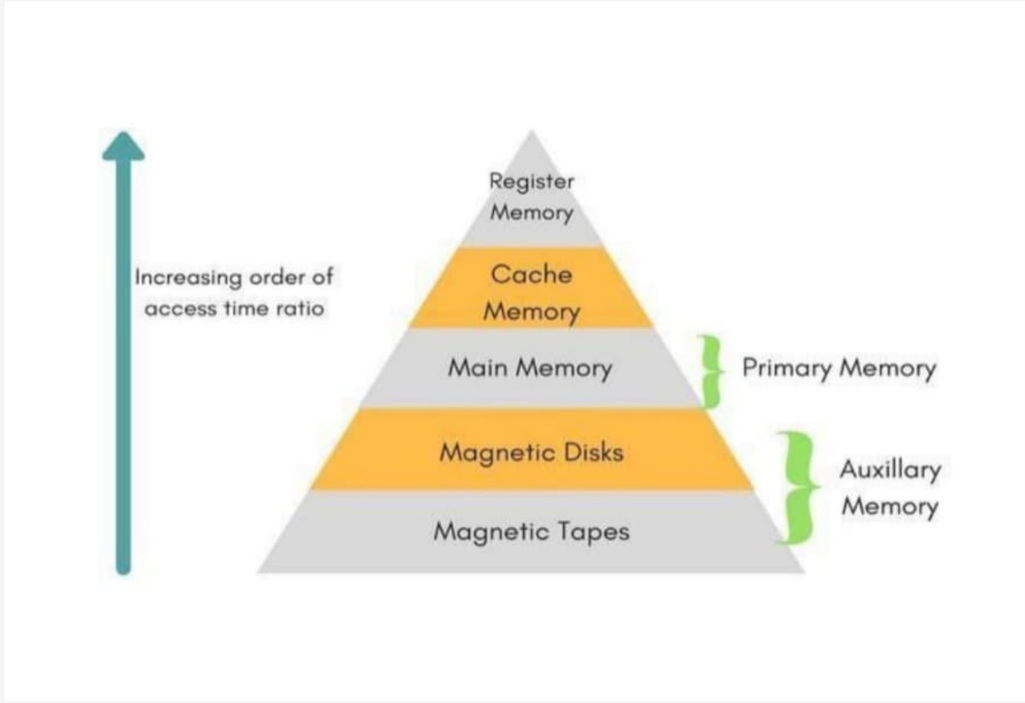
PRANAY
106119066

VAMSI
106119126

MEMORY HIERARCHY

Memory Hierarchy is an enhancement to organize the memory such that it can minimize the access time.

The Memory Hierarchy was developed based on a program behavior known as locality of references



This Memory Hierarchy Design is divided into Two main types

- External Memory or Secondary Memory – Comprising of Magnetic Disk, Optical Disk, Magnetic Tape i.e. peripheral storage devices which are accessible by the processor via I/O Module.

- Internal Memory or Primary Memory – Comprising of Main Memory, Cache Memory & CPU registers. This is directly accessible by the processor.

We can infer the following characteristics of Memory Hierarchy Design

- **Capacity:** It is the global volume of information the memory can store. As we move from top to bottom in the Hierarchy, the capacity increases.
- **Access Time:** It is the time interval between the read/write request and the availability of the data. As we move from top to bottom in the Hierarchy, the access time increases.
- **Performance:** Earlier when the computer system was designed without Memory Hierarchy design, the speed gap increases between the CPU registers and Main Memory due to large difference in access time. This results in lower performance of the system and thus, enhancement was required. This enhancement was made in the form of Memory Hierarchy Design because of which the performance of the system increases. One of the most significant ways to increase system performance is minimizing how far down the memory hierarchy one has to go to manipulate data.
- **Cost per bit**: As we move from bottom to top in the Hierarchy, the cost per bit increases i.e. Internal Memory is costlier than External Memory

SECONDARY MEMORY

- The secondary storage devices which are built into the computer or connected to the computer are known as a secondary memory of the computer. It is also known as external memory or auxiliary storage.

- There are several types of secondary memory, including hard disk drives (HDDs), solid-state drives (SSDs), optical disks (CDs, DVDs, and Blu-ray disks), and flash memory

HARD DISK

- It is a rigid magnetic disc that stores data permanently, as it is a non-volatile storage device.
- A circuit board, which is called the disk controller or interface board, is present on the back of a hard drive. It allows the hard drive to communicate with the computer..
- The main components of a hard drive include a head actuator, read/write actuator arm, read/write head, platter, and spindle
- The data stored on a computer's hard drive generally includes the operating system, installed software, and the user's files and programs, including pictures, music, videos, text documents, etc.

SOLID-STATE DRIVE

- SSD (Solid State Drive) is also a non-volatile storage medium that is used to hold and access data. Unlike a hard drive, it does not have moving components, so it offers many advantages over SSD, such as faster access time, noiseless operation, less power consumption, and more.

- As the cost of SSD has come down, it has become an ideal replacement for a standard hard drive in desktop and laptop computers. It is also suitable for notebooks, and tablets that don't require lots of storage.

SOME MORE DEVICES

PENDRIVE

SD CARD

CD

Problem

Suppose we have a disk with 8,192 sectors and a rotation time of 10 ms. The seek time for a sector is 6 ms, and the transfer time for a sector is 0.2 ms. What is the total time to read a sector, and what is the average access time?

Solution

Total Time = Rotation Time + Seek Time + Transfer Time

Average Access Time = (Seek Time / 3) + Rotation Time + (Transfer Time / 2)

Given,

Rotation time = 10ms , seek time = 6ms , Transfer time = 0.2ms

therefore total time = 10ms + 6ms +0.2ms = 16.2ms

Average Access Time = (Seek Time / 3) + Rotation Time + (Transfer Time / 2)

Average Access Time = (6 ms / 3) + 10 ms + (0.2 ms / 2)

Average Access Time = 2 ms + 10 ms + 0.1 ms

Average Access Time = 12.1 ms.

MAIN MEMORY

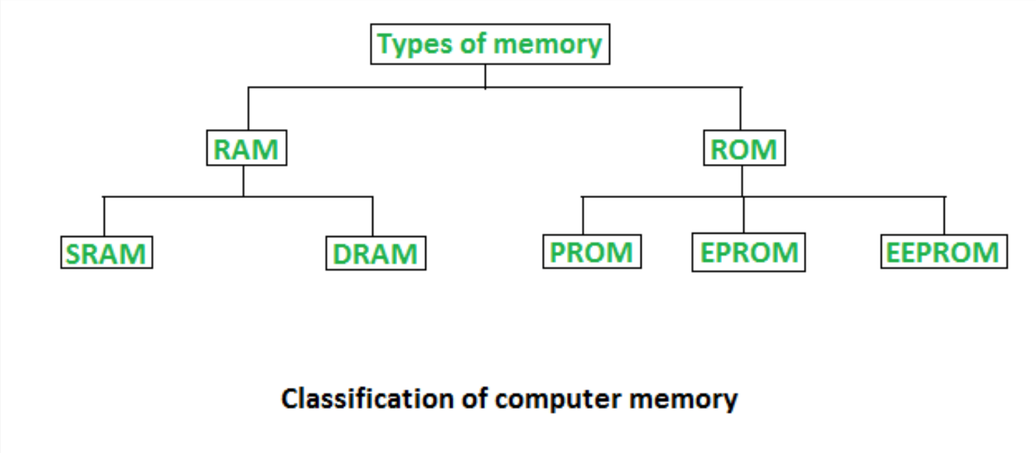
- The main memory also known as primary memory acts as the central storage unit in a computer system. It is a relatively large and fast memory which is used to store programs and data during the run time operations.

- The primary technology used for the main memory is based on semiconductor integrated circuits

Classification of Primary memory

Primary memory can be broadly classified into two parts:

1. Read-Only Memory (ROM)
2. Random Access Memory (RAM)

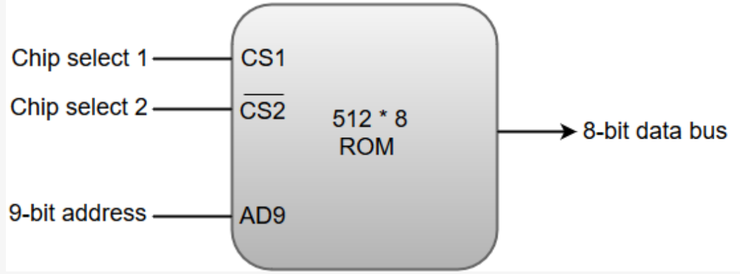


Classification of computer memory

Read-only Memory(ROM)

- Any data which need not be altered are stored in ROM. ROM includes those programs which run on booting of the system (know as a bootstrap program that initializes OS) along with data like algorithm required by OS.
- The following block diagram demonstrates the chip interconnection in a 512 * 8 ROM chip.
- a ROM can only perform read operation; the data bus can only operate in an output mode.
- The 9-bit address lines in the ROM chip specify any one of the 512 bytes stored in it.
- Content of ROM is non-volatile in nature, they are stored even when power is lost.

Typical ROM chip:



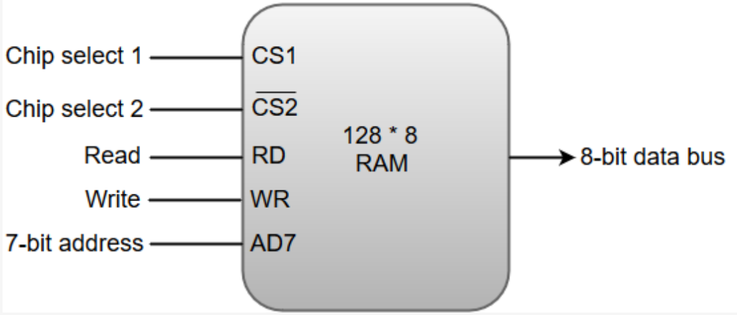
Types of ROM

- **MROM:** Masked ROM are hardwired and pre-programmed ROM. Any content that is once written cannot be altered anyhow.
- **PROM:** Programmable ROM can be modified once by the user. The user buys a blank PROM and writes the desired content but once written content cannot be altered.
- **EPROM:** Erasable and Programmable ROM Content can be changed by erasing the initial content which can be done by exposing EPROM to UV radiation. This exposure to ultra-violet light dissipates the charge on ROM and content can be rewritten on it.
- **EEPROM:** Electrically Erasable and Programmable ROM Content can be changed by erasing the initial content which could be easily erased electrically. However, one byte can be erased at a time instead of deleting in one go. Hence, reprogramming of EEPROM is a slow process.

Random Access Memory

- Any process in the system which needs to be executed is loaded in RAM which is processed by the CPU as per Instructions in the program. Like if we click on applications like Browser, firstly browser code will be loaded by the Operating system into the RAM after which the CPU will execute and open up the Browser.
- The following block diagram demonstrates the chip interconnection in a 128 * 8 RAM chip.
- The 8-bit bidirectional data bus allows the transfer of data either from memory to CPU during a read operation or from CPU to memory during a write operation.
- Content of RAM is volatile in nature, it vanishes when power is lost.

Typical RAM chip:



Types of RAM

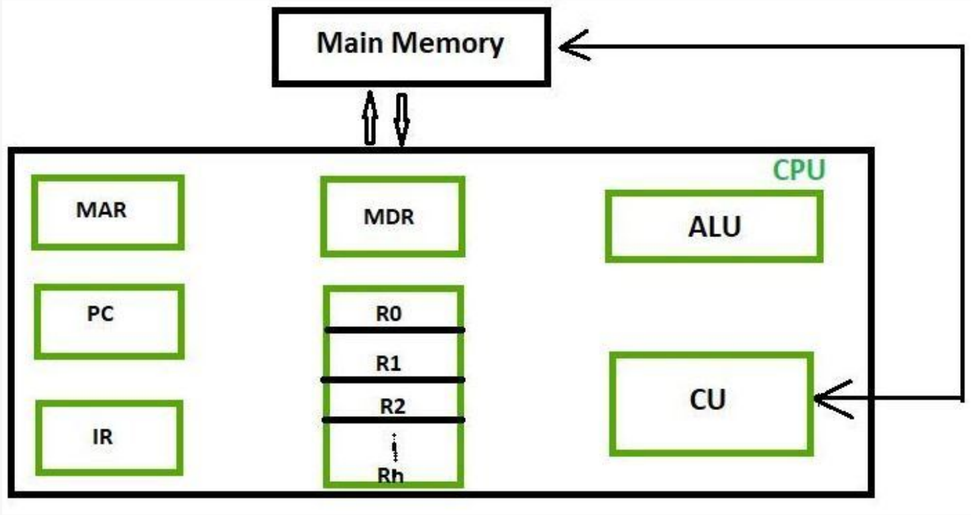
RAM can be broadly classified into SRAM (Static RAM) and DRAM (Dynamic RAM) based on their behavior:

- **DRAM:** Dynamic RAM or DRAM needs to periodically refresh in few milliseconds to retain data. DRAM is made up of capacitors and transistors and electric charge leaks from capacitors and DRAM needs to be charged periodically. DRAM is widely used in home PCs and servers as it is cheaper than SRAM.

- **SRAM:** Static RAM or SRAM keeps the data as long as power is supplied to the system. SRAM uses Sequential circuits like a flip-flop to store a bit and hence need not be periodically refreshed. SRAM is expensive and hence only used where speed is the utmost priority.

REGISTERS

- Registers are a type of computer memory used to quickly accept, store, and transfer data and instructions that are being used immediately by the CPU.
- Registers are very fast computer memory which are used to execute programs and operations efficiently.
- The sole purpose of having register is fast retrieval of data for processing by CPU.



Types of Registers

Accumulator:

- This is the most frequently used register used to store data taken from memory. It is in different numbers in different microprocessors.

Memory Address Registers (MAR):

- It holds the address of the location to be accessed from memory. MAR and MDR (Memory Data Register) together facilitate the communication of the CPU and the main memory.

Memory Data Registers (MDR):

- It contains data to be written into or to be read out from the addressed location.

General Purpose Registers:

- These are numbered as R0, R1, R2....Rn-1, and used to store temporary data during any ongoing operation. Its content can be accessed by assembly programming

Program Counter (PC):

- Program Counter (PC) is used to keep the track of execution of the program. It contains the memory address of the next instruction to be fetched.
- PC points to the address of the next instruction to be fetched from the main memory when the previous instruction has been successfully completed.
- Program Counter (PC) also functions to count the number of instructions.
- The incrementation of PC depends on the type of architecture being used.
- If we are using 32-bit architecture, the PC gets incremented by 4 every time to fetch the next instruction.

Instruction Register (IR):

- The IR holds the instruction which is just about to be executed. The instruction from PC is fetched and stored in IR. As soon as the instruction in placed in IR, the CPU starts executing the instruction and the PC points to the next instruction to be executed.

Condition code register (CCR) :

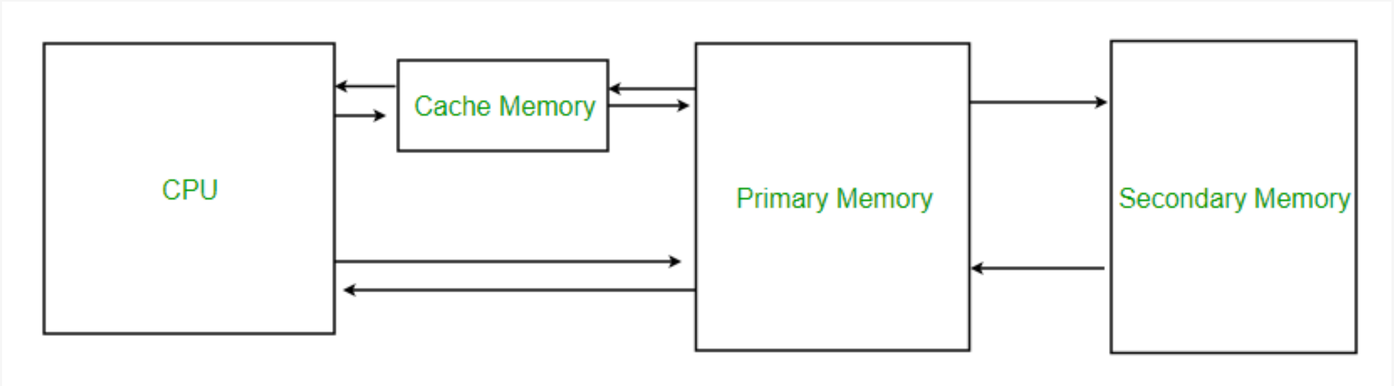
- Condition code registers contain different flags that indicate the status of any operation. For instance lets suppose an operation caused creation of a negative result or zero, then these flags are set high accordingly. And the flags are

1. Carry C: Set to 1 if an add operation produces a carry or a subtract operation produces a borrow; otherwise cleared to 0.
2. Overflow V: Useful only during operations on signed integers.
3. Zero Z: Set to 1 if the result is 0, otherwise cleared to 0.
4. Negate N: Meaningful only in signed number operations. Set to 1 if a negative result is produced.
5. Extend X: Functions as a carry for multiple precision arithmetic operations.

After registers we have cache memory, which are faster but less faster than registers.

Cache Memory

- Cache Memory is a special very high-speed memory. It is used to speed up and synchronize with high-speed CPU.
- Cache memory is costlier than main memory or disk memory but more economical than CPU registers.
- Cache memory is an extremely fast memory type that acts as a buffer between RAM and the CPU.
- It holds frequently requested data and instructions so that they are immediately available to the CPU when needed.
- Cache memory is used to reduce the average time to access data from the Main memory.
- The cache is a smaller and faster memory that stores copies of the data from frequently used main memory locations.



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.

Cache Operation

- It is based on the principle of locality of reference. There are two ways with which data or instruction is fetched from main memory and get stored in cache memory

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.

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.

Tavg = Average time to access memory

Tavg = h * Tc + (1-h)*(Tm + Tc)

Cache Performance:

When the processor needs to read or write a location in main memory, it first checks for a corresponding entry in the cache.

- If the processor finds that the memory location is in the cache, a cache hit has occurred and data is read from the cache.
- If the processor does not find the memory location in the cache, a cache miss has occurred. For a cache miss, the cache allocates a new entry and copies in data from main memory, then the request is fulfilled from the contents of the cache.

The performance of cache memory is frequently measured in terms of a quantity called Hit ratio.

Hit ratio = hit / (hit + miss) = no. of hits/total accesses

We can improve Cache performance using higher cache block size, and higher associativity, reduce miss rate, reduce miss penalty, and reduce the time to hit in the cache.

Problem

A cache memory that has a hit ratio of 0.8 has an access latency 10ns and miss penalty 100ns. An optimization is done on the cache to reduce the miss rate. However, the optimization results in a increase of cache access latency to 15ns, where as the miss penalty is not affected. The minimum hit rate needed after the optimization such that it should not increase the average memory access time is___

Answer = 0.85.

Explanation :

Tavg = Tc +(1-h) Tm

Initial Tavg= 10 + (1-0.8) *100

= 10+20= 30ns.

New,

Tavg = 15 + (1-h) * 100

30 = 15+(1-h) * 100

0.15= 1-h

Therefore, h=0.85

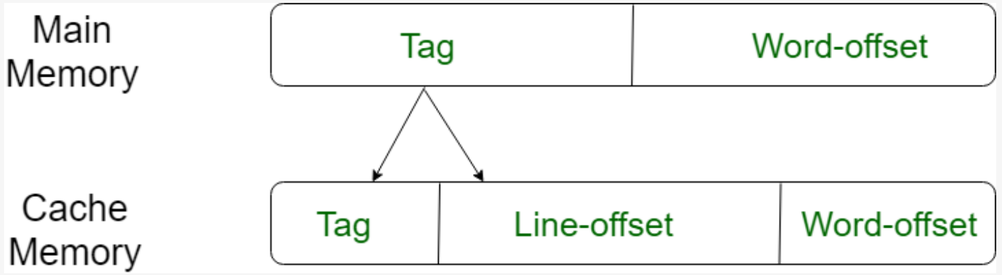
Cache Mapping:

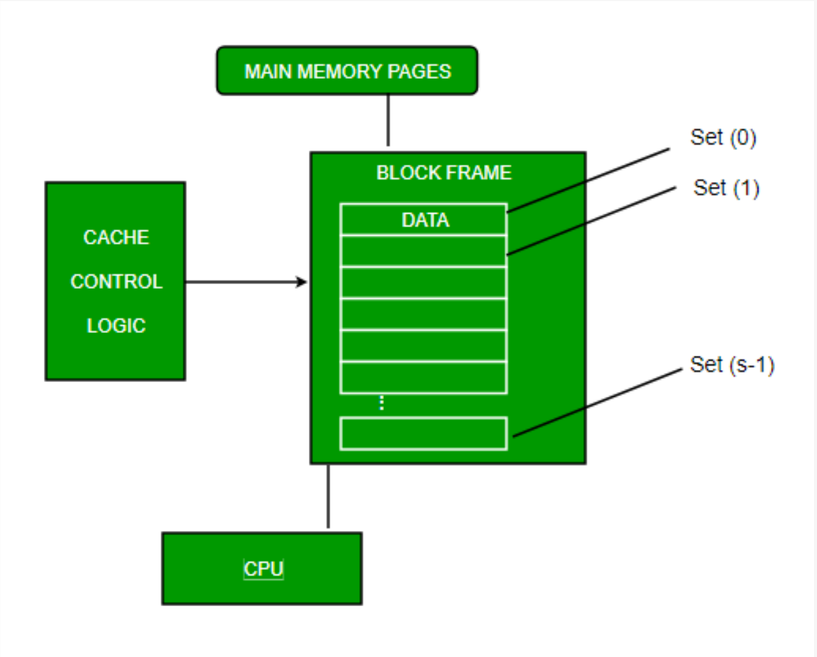
A. Direct Mapping

- The simplest technique, known as direct mapping, maps each block of main memory into only one possible cache line. In Direct mapping, assign each memory block to a specific line in the cache.
- If a line is previously taken up by a memory block when a new block needs to be loaded, the old block is trashed. An address space is split into two parts index field and a tag field.
- The cache is used to store the tag field whereas the rest is stored in the main memory.
- Direct mapping`s performance is directly proportional to the Hit ratio.

.

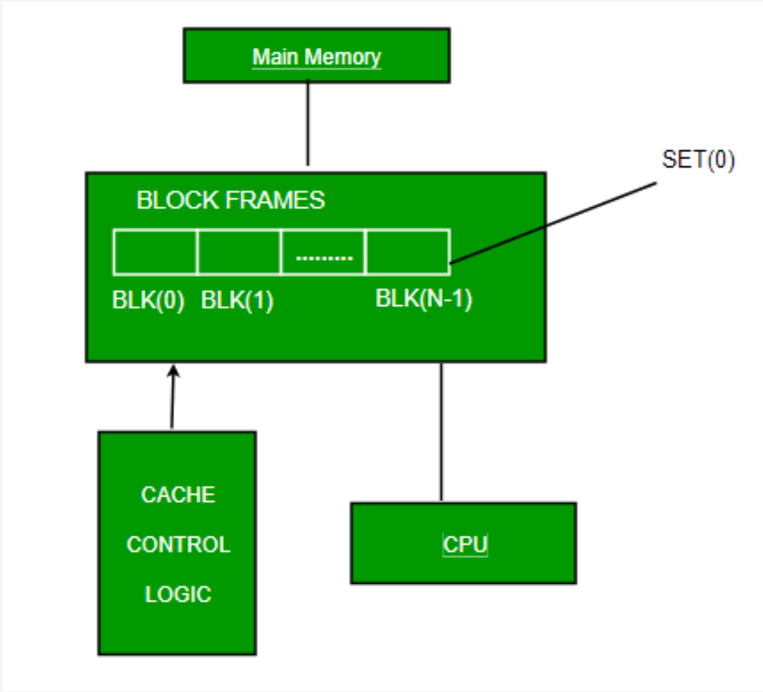
- For purposes of cache access, each main memory address can be viewed as consisting of three fields. The least significant w bits identify a unique word or byte within a block of main memory
- In most contemporary machines, the address is at the byte level.
- The remaining s bits specify one of the 2s blocks of main memory. The cache logic interprets these s bits as a tag of s-r bits (most significant portion) and a line field of r bits.
- This latter field identifies one of the m=2r lines of the cache.
- Line offset is index bits in the direct mapping.





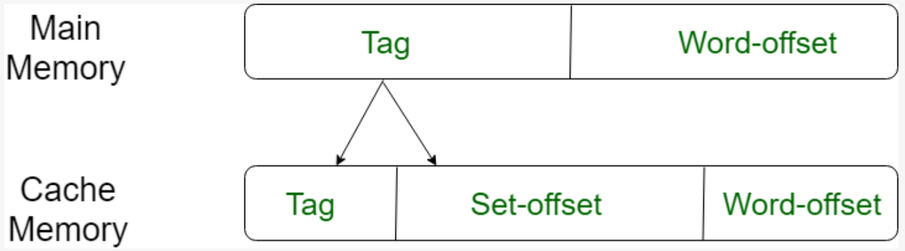
B.Associative Mapping

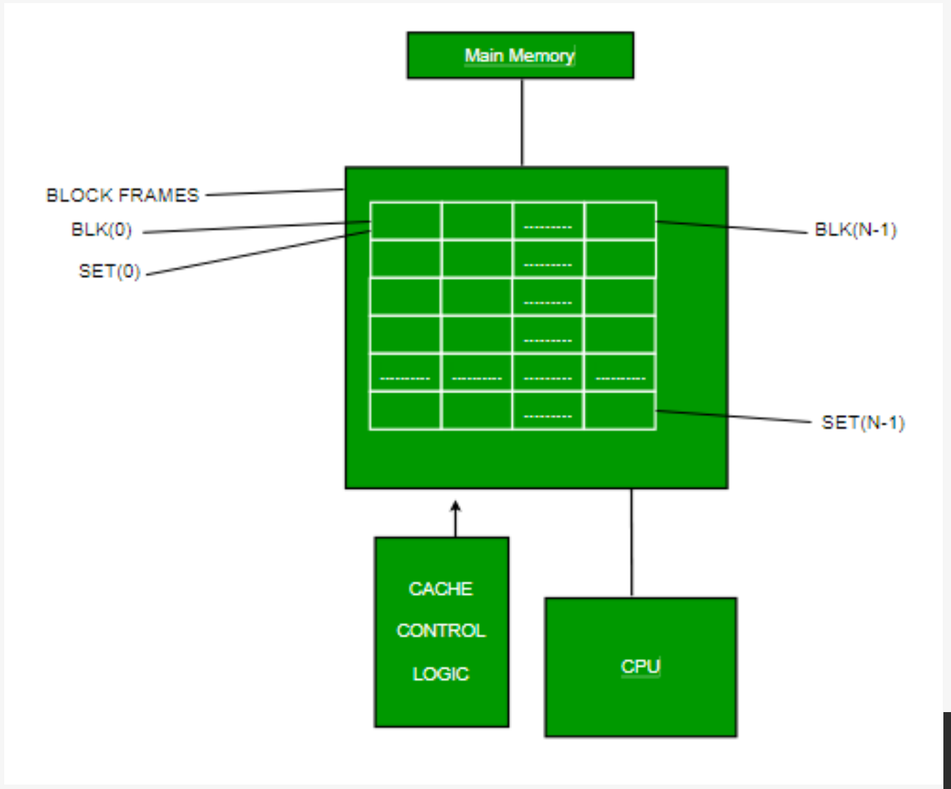
- in this type of mapping, the associative memory is used to store content and addresses of the memory word. Any block can go into any line of the cache.
- This means that the word id bits are used to identify which word in the block is needed, but the tag becomes all of the remaining bits. This enables 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.
- In associative mapping the index bits are zero.



C. Set-associative Mapping

- This form of mapping is an enhanced form of direct mapping where the drawbacks of direct mapping are removed. Set associative addresses the problem of possible thrashing in the direct mapping method.
- Set-associative mapping allows that each word that is present in the cache can have two or more words in the main memory for the same index address.
- Set associative cache mapping combines the best of direct and associative cache mapping techniques. In set associative mapping the index bits are given by the set offset bits.





Application of Cache Memory:

1. Primary Cache – A primary cache is always located on the processor chip. This cache is small and its access time is comparable to that of processor registers.
2. Secondary Cache – Secondary cache is placed between the primary cache and the rest of the memory. It is referred to as the level 2 (L2) cache. Often, the Level 2 cache is also housed on the processor chip.
3. Spatial Locality of reference This says that there is a chance that the element will be present in close proximity to the reference point and next time if again searched then more close proximity to the point of reference.
4. Temporal Locality of reference In this Least recently used algorithm will be used. Whenever there is page fault occurs within a word will not only load word in main memory but complete page fault will be loaded because the spatial locality of reference rule says that if you are referring to any word next word will be referred to in its register that's why we load complete page table so the complete block will be loaded.

Problem

A 4-way set-associative cache memory unit with a capacity of 16 KB is built using a block size of 8 words. The word length is 32 bits. The size of the physical address space is 4 GB. The number of bits for the TAG field is _____

(A) 5

(B) 15

(C) 20

(D) 25

Answer: (C)

Explanation: 4way cache => set size = number of lines of cache/4 = (16KB/Word size) / 4

= 4 * 2^10 B / 8 * 4 B = 2^7 = 128

=> set bits = 7

block size = 8*4 bytes = 2^5 Bytes

word size = 5, total memory = 4GB = 2^32

=> tag bits = 32-5-7 = 20

Problem

Consider a computer system with a byte-addressable primary memory of size 232 bytes. Assume the computer system has a direct-mapped cache of size 32 KB (1 KB = 210 bytes), and each cache block is of size 64 bytes.

The size of the tag field is _____ bits.

(A) 17

(B) 18

(C) 15

(D) 9

Answer: (A)

Given, byte-addressable primary memory of size 232 bytes = 32 bits of MM width.

Cache block size = 64 bytes = 26 bytes = 6 bits in mem block width.

Number of cache block (#lines) = Cache size / Cache block size = 32 KB / 64 bytes = 512 = 29 = 9 bits in LO.

Hence, Tag bits are = 32 – 9 – 6 = 17 bits.

Problem

A certain processor uses a fully associative cache of size 16 kB, The cache block size is 16 bytes. Assume that the main memory is byte addressable and uses a 32-bit address. How many bits are required for the Tag and the Index fields respectively in the addresses generated by the processor?

(A) 24 bits and 0 bits

(B) 28 bits and 4 bits

(C) 24 bits and 4 bits

(D) 28 bits and 0 bits

Answer: (D)

We know in fully associative mapping, Line size = block size = frame size

Number of bits in tag can be founded using given below formula

Number of Tag bits = Total number of bits in Physical Address - no of bits in Block offset

Here number of bits in block offset is not given. It can be found using

ceil(log2 Cache block size) = ceil(log2 16) = 4 **So, Number of Tag bits = 32-4 = 28**

No index bits is there in fully associative mapping, hence Index bits = 0

Problem

Consider a machine with a byte addressable main memory of 220 bytes, block size of 16 bytes and a direct mapped cache having 212 cache lines. Let the addresses of two consecutive bytes in main memory be (E201F)16 and (E2020)16. What are the tag and cache line address (in hex) for main memory address (E201F)16?

(A) E, 201

(B) F, 201

(C) E, E20

(D) 2, 01F

Answer: (A)

Explanation:

Block Size = 16 bytes, Block Offset = 4

No. of sets or cache lines = 212, Number of index bits = 12

Size of main memory = 220, Number of tag bits = 20 – 12 – 4 = 4

Let us consider the hex address E201F , Tag lines = First 4 bits = E (in hex)

Cache lines = Next 12 bits = 201 (In Hex)