Computer Organization and Architecture
B. E. 4th SEM BIT Durg
Session 2022-23

Usha Kiran

April 3, 2023

Unit-III The Memory System

1 Overview

- Various Technologies Used in Memory Design
- Memory Hierarchy
- Cache Memory
  - Cache Mapping Functions: Direct mapping, associative mapping, k-way set associative mapping
  - Replacement Algorithms
  - Write Policy
  - Cache Coherency
- Virtual Memory
- Main Memory
- Auxiliary Memory
- Associative Memory
- Multi-module Memories and Interleaving

2 Memory Technologies

The term Memory can be defined as a collection of data in a specific format. It is used to store instructions and process data. The memory comprises a large array or group of words or bytes, each with its own location. The primary motive of a computer system is to execute programs. These programs, along with the information they access, should be in the main memory during execution. The CPU fetches instructions from memory according to the value of the program counter.

To achieve a degree of multiprogramming and proper utilization of memory, memory management is important. Many memory management methods exist, reflecting various approaches, and the effectiveness of each algorithm depends on the situation.

2.1 Performance Measures

Memory latency is traditionally quoted using two measures—access time and cycle time. Access time is the time between when a read is requested and when the desired word arrives, cycle time is the minimum time between requests to memory. One reason that cycle time is greater than access time is that the memory needs the address lines to be stable between accesses.

3 Technologies Used in Memory Design

- RAM
  - DRAM
  - SRAM
- ROM
  - MROM
  - PROM
  - EPROM
  - EEPROM
  - Flash Memory

4 RAM

4.1 Introduction

RAM (Random Access Memory) is the hardware in a computing device where the operating system (OS), application programs and data in current use are kept so they can be quickly reached by the device's processor. RAM is the main memory in a computer. It is much faster to read from and write to than other kinds of storage, such as a hard disk drive (HDD), solid-state drive (SSD) or optical drive.

Random Access Memory is volatile. That means data is retained in RAM as long as the computer is on, but it is lost when the computer is turned off. When the computer is rebooted, the OS and other files are reloaded into RAM, usually from an HDD or SSD.

4.2 Function of RAM

Because of its volatility, RAM can't store permanent data. RAM can be compared to a person's short-term memory, and a hard disk drive to a person's long-term memory. Short-term memory is focused on immediate work, but it can only keep a limited number of facts in view at any one time. When a person's short-term memory fills up, it can be refreshed with facts stored in the brain's long-term memory.

A computer also works this way. If RAM fills up, the computer's processor must repeatedly go to the hard disk to overlay the old data in RAM with new data. This process slows the computer's operation.

Types of RAM

- Static Random Access Memory (SRAM)
- Dynamic Random Access Memory (DRAM)

4.3 SRAM

- Static random-access memory (static RAM or SRAM) is a type of random-access memory (RAM) that uses latching circuitry (flip-flop) to store each bit.
- SRAM is volatile memory; data is lost when power is removed.
- SRAM will hold its data permanently in the presence of power, while data in DRAM decays in seconds and thus must be periodically refreshed.
- SRAM is faster than DRAM but it is more expensive in terms of silicon area and cost; it is typically used for the cache and internal registers of a CPU while DRAM is used for a computer's main memory.
- Since SRAM requires more transistors per bit to implement, it is less dense and more expensive than DRAM and also has a higher power consumption during read or write access.
- Used in personal computers, workstations, routers and peripheral equipment: CPU register files, internal CPU caches, internal GPU caches and external burst mode SRAM caches, hard disk buffers, router buffers, etc.

4.4 DRAM Technology

- The main memory of virtually every desktop or server computer sold since 1975 is composed of semiconductor DRAMs,.
- Access memory address as accessing Matrix.
- Emphasis is on cost per bit and capacity.
- The dynamic nature of the circuits in DRAM require data to be written back after being read, hence the difference between the access time and the cycle time as well as the need to refresh.
- Dynamic random-access memory (dynamic RAM or DRAM) is a type of random-access semiconductor memory that stores each bit of data in a memory cell, usually consisting of a tiny capacitor and a transistor.
- While most DRAM memory cell designs use a capacitor and transistor, some only use two transistors.
- In the designs where a capacitor is used, the capacitor can either be charged or discharged; these two states are taken to represent the two values of a bit, conventionally called 0 and 1.
- The electric charge on the capacitors gradually leaks away; without intervention the data on the capacitor would soon be lost.
- To prevent this, DRAM requires an external memory refresh circuit which periodically rewrites the data in the capacitors, restoring them to their original charge.
- This refresh process is the defining characteristic of dynamic random-access memory, in contrast to static random-access memory (SRAM) which does not require data to be refreshed.
- Unlike flash memory, DRAM is volatile memory (vs. non-volatile memory), since it loses its data quickly when power is removed.
- DRAM typically takes the form of an integrated circuit chip, which can consist of dozens to billions of DRAM memory cells.
- DRAM chips are widely used in digital electronics where low-cost and high-capacity computer memory is required.
- One of the largest applications for DRAM is the main memory (colloquially called the "RAM" in modern computers and graphics cards (where the "main memory" is called the graphics memory).

5 ROM

ROM, which stands for read only memory, is a memory device or storage medium that stores information permanently. It is also the primary memory unit of a computer along with the random access memory (RAM). It is called read only memory as we can only read the programs and data stored on it but cannot write on it. It is restricted to reading words that are permanently stored within the unit.

The manufacturer of ROM fills the programs into the ROM at the time of manufacturing the ROM. After this, the content of the ROM can't be altered, which means you can't reprogram, rewrite, or erase its content later. However, there are some types of ROM where you can modify the data.

ROM contains special internal electronic fuses that can be programmed for a specific interconnection pattern (information). The binary information stored in the chip is specified by the designer and then embedded in the unit at the time of manufacturing to form the required interconnection pattern (information). Once the pattern (information) is established, it stays within the unit even when the power is turned off. So, it is a non-volatile memory as it holds the information even when the power is turned off, or you shut down your computer.

The information is added to a RAM in the form of bits by a process known as programming the ROM as bits are stored in the hardware configuration of the device. So, ROM is a Programmable Logic Device (PLD).

ROM is an abbreviation for Read Only Memory, and is also referred to as firmware. It is an integrated circuit programmed with certain data during the time of it's manufacturing. Read only memories are used not only in computer systems but also in many other electronic devices such as IoT devices like digital assistants, smart gadgets such as smartwatches, etc as well.

1. MROM
2. PROM

3. EPROM
4. EEPROM
5. Flash Memory

5.1 Types of ROM:

- Masked Read Only Memory (MROM): It is the oldest type of read only memory (ROM). It has become obsolete so it is not used anywhere in today's world. It is a hardware memory device in which programs and instructions are stored at the time of manufacturing by the manufacturer. So it is programmed during the manufacturing process and can't be modified, reprogrammed, or erased later.

- The MROM chips are made of integrated circuits. Chips send a current through a particular input-output pathway determined by the location of fuses among the rows and columns on the chip. The current has to pass along a fuse-enabled path, so it can return only via the output the manufacturer chooses. This is the reason the rewriting and any other modification is not impossible in this memory.

- Programmable Read Only Memory (PROM): PROM is a blank version of ROM. It is manufactured as blank memory and programmed after manufacturing. We can say that it is kept blank at the time of manufacturing. You can purchase and then program it once using a special tool called a programmer.

  In the chip, the current travels through all possible pathways. The programmer can choose one particular path for the current by burning unwanted fuses by sending a high voltage through them. The user has the opportunity to program it or to add data and instructions as per his requirement. Due to this reason, it is also known as the user-programmed ROM as a user can program it.

  To write data onto a PROM chip; a device called PROM programmer or PROM burner is used. The process or programming a PROM is known as burning the PROM. Once it is programmed, the data cannot be modified later, so it is also called as one-time programmable device. Uses: It is used in cell phones, video game consoles, medical devices, RFID tags, and more.

- 3) Erasable and Programmable Read Only Memory (EPROM): EPROM is a type of ROM that can be reprograme and erased many times. The method to erase the data is very different; it comes with a quartz window through which a specific frequency of ultraviolet light is passed for around 40 minutes to erase the data. So, it retains its content until it is exposed to the ultraviolet light. You need a special device called a PROM programmer or PROM burner to reprogram the EPROM.

  Uses: It is used in some micro-controllers to store program, e.g., some versions of Intel 8048 and the Freescale 68HC11.

- Electrically Erasable and Programmable Read Only Memory (EEPROM): ROM is a type of read only memory that can be erased and reprogrammed repeatedly, up to 10000 times. It is also known as Flash EEPROM as it is similar to flash memory. It is erased and reprogrammed electrically without using ultraviolet light. Access time is between 45 and 200 nanoseconds. The data in this memory is written or erased one byte at a time; byte per byte, whereas, in flash memory data is written and erased in blocks. So, it is faster than EEPROM. It is used for storing a small amount of data in computer and electronic systems and devices such as circuit boards.

  Uses: The BIOS of a computer is stored in this memory.

- FLASH ROM: It is an advanced version of EEPROM. It stores information in an arrangement or array of memory cells made from floating-gate transistors. The advantage of using this memory is that you can delete or write blocks of data around 512 bytes at a particular time. Whereas, in EEPROM, you can delete or write only 1 byte of data at a time. So, this memory is faster than EEPROM.

  It can be reprogrammed without removing it from the computer. Its access time is very high, around 45 to 90 nanoseconds. It is also highly durable as it can bear high temperature and intense pressure.

  Uses: It is used for storage and transferring data between a personal computer and digital devices. It is used in USB flash drives, MP3 players, digital cameras, modems and solid-state drives (SSDs). The BIOS of many modern computers are stored on a flash memory chip, called flash BIOS.

6 Computer Memory Hierarchy

Memory Hierarchy Design is divided into 2 main types:

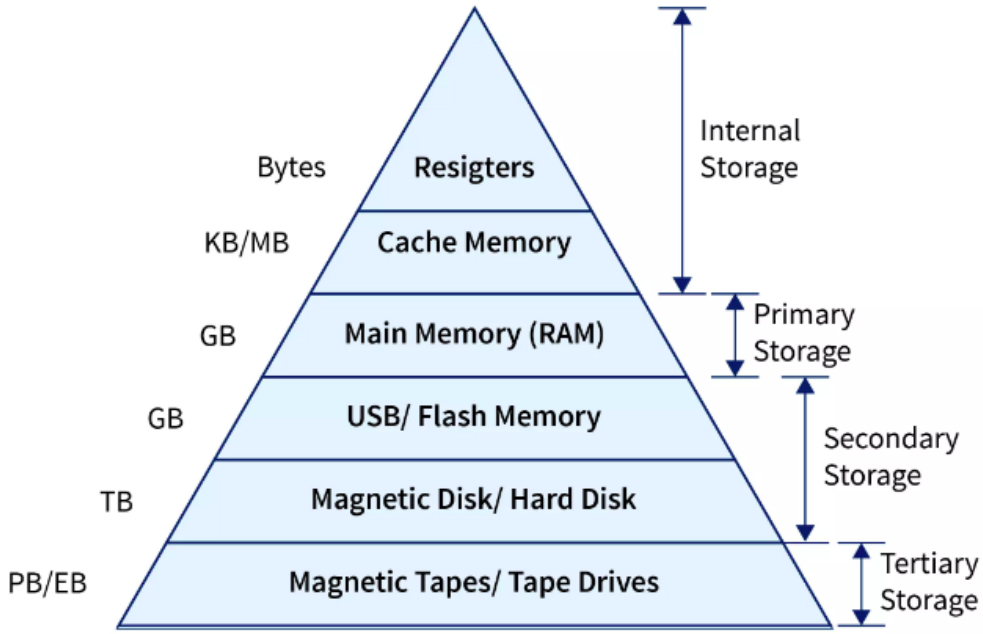

Figure 1: Computer Memory Hierarchy

- 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.

There are typically four levels of memory in a memory hierarchy:

- Registers: Registers are small, high-speed memory units located in the CPU. The register is usually an SRAM or static RAM in the computer processor that is used to hold the data word that is typically 64 bits or 128 bits. A majority of the processors make use of a status word register and an accumulator. They are used to store the most frequently used data and instructions. Registers have the fastest access time and the smallest storage capacity.
- Cache Memory: Cache memory is a small, fast memory unit located close to the CPU. It stores frequently used data and instructions that have been recently accessed from the main memory, hence, minimize the time to access data. We can also find cache memory in the processor. In case the processor has a single-core, it will rarely have multiple cache levels. The present multi-core processors would have three 2-levels for every individual core, and one of the levels is shared.
- Main Memory: Main memory, also known as RAM (Random Access Memory), is the primary memory of a computer system. It has a larger storage capacity than cache memory, but it is slower. Main memory is used to store data and instructions that are currently in use by the CPU.
- USB/ Flash Memory: A USB flash drive (also called a thumb drive in the US, or a memory stick in the UK)[1][note 1] is a data storage device that includes flash memory with an integrated USB interface. It is typically removable, re-writable and much smaller than an optical disc.
- Magnetic Disks: It is a secondary storage device, with high storage capacity, low cost and low speed. In a computer, the magnetic disks are circular plates that's fabricated with plastic or metal with a magnetised material. Two faces of a disk are frequently used, and many disks can be stacked on a single spindle by read/write heads that are obtainable on every plane. The disks in a computer jointly turn at high speed.
- Magnetic Tape: It is also known as tertiary storage. Magnetic tape refers to a normal magnetic recording designed with a slender magnetizable overlay that covers an extended, thin strip of plastic film. It is used mainly to back up huge chunks of data. When a computer needs to access a strip, it will first mount it to access the information. Once the information is allowed, it will then be unmounted. The actual access time of a computer memory would be slower within a magnetic strip, and it will take a few minutes for us to access a strip.

6.1 Characteristics of Memory

We can infer the following characteristics of Memory Hierarchy Design from Figure 1:

- 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.

Level	1	2	3	4
Name	Register	Cache	Main Memory	Secondary Memory
Size	¡1 KB	less than 16 MB	¡16GB	¿100 GB
Implementation	Multi-ports	On-chip/SRAM	DRAM (capacitor memory)	Magnetic
Access Time	0.25ns to 0.5ns	0.5 to 25ns	80ns to 250ns	50 lakh ns
Bandwidth	20000 to 1 lakh MBytes	5000 to 15000	1000 to 5000	20 to 150

Table 1: Caption

6.2 Processor Registers

- Quickly accessible to a computer's processor.
- Top of the memory hierarchy.
- Small capacity and fast accessing.
- May have specific hardware functions, and may be read-only or write-only mode.
- Used for arithmetic operations and is manipulated or tested by machine instructions.
- When a computer program accesses the same data repeatedly, this is called "locality of reference".
- Registers are normally measured by the number of bits they can hold, for example, an "8-bit register", "32-bit register", "64-bit register", or even more.

A processor often contains several kinds of registers.
Data Register

- User-accessible registers can be read or written by machine instructions. Types: Data Register, Address Register.
- Data registers can hold numeric data values such as integer and, in some architectures, floating-point values, as well as characters, small bit arrays and other data.

Address Register Address registers hold addresses and are used by instructions that indirectly access primary memory.

6.3 location of register in CPU

6.4 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. There are various different independent caches in a CPU, which store instructions and data.

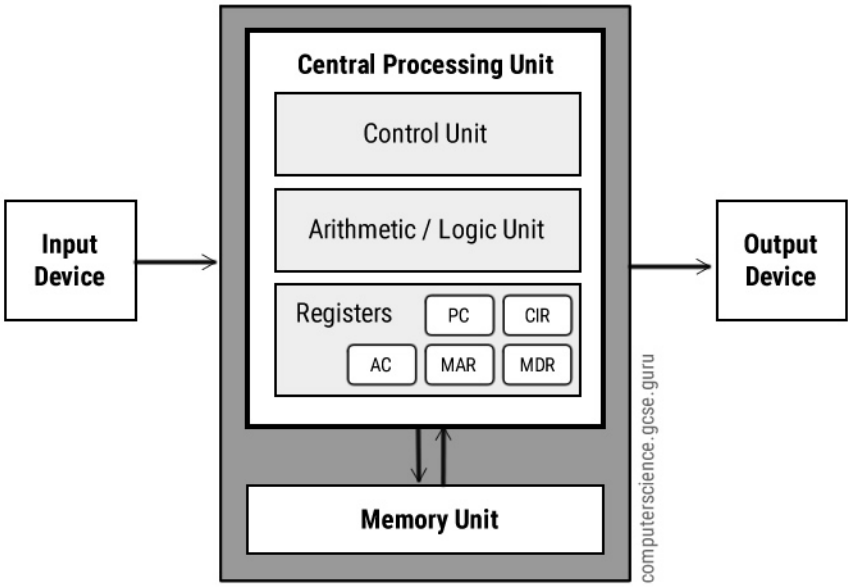

Figure 2: Register Position in CPU

6.5 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(H) = hit / (hit + miss) = no. of hits/total accesses Miss ratio = miss / (hit + miss) = no. of miss/total accesses = 1 - hit ratio(H)

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.

6.6 Application of Cache Memory:

- Usually, the cache memory can store a reasonable number of blocks at any given time, but this number is small compared to the total number of blocks in the main memory.
- The correspondence between the main memory blocks and those in the cache is specified by a mapping function.
- 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.
- 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.
- 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.
- 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.

7 Main Memory

The main memory is the fundamental storage unit in a computer system. It is associatively large and quick memory and saves programs and information during computer operations. The technology that makes the main

memory work is based on semiconductor integrated circuits.

RAM is the main memory. Integrated circuit Random Access Memory (RAM) chips are applicable in two possible operating modes are as follows.

Static: It consists of internal flip-flops, which store the binary information. The stored data remains solid considering power is provided to the unit. The static RAM is simple to use and has smaller read and write cycles. **Dynamic:** It saves the binary data in the structure of electric charges that are used to capacitors. The capacitors are made available inside the chip by Metal Oxide Semiconductor (MOS) transistors. The stored value on the capacitors contributes to discharge with time and thus, the capacitors should be regularly recharged through stimulating the dynamic memory.

8 Cache Mapping

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.

There are three types of cache mapping:

- Direct Mapping
- Associative/Fully-associative Mapping
- Set-associative Mapping

8.1 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

$$i = j \ module \ m$$

Where,

$i = Cache\ line\ number$
$j = Main\ memory\ block\ number$
$m = Number\ of\ lines\ in\ the\ cache$

In direct mapping cache, instead of storing total address information with data in cache only part of address bits is stored along with data.

The new data has to be stored only in a specified cache location as per the mapping rule for direct mapping. So it doesn't need replacement algorithm.

Advantages of direct mapping

- The direct mapping technique is simple and inexpensive to implement.
- Here only tag field is required to match while searching word that is why it fastest cache.
- Direct mapping cache is less expensive compared to associative cache mapping.

Disadvantages of direct mapping

- Its main disadvantage is that there is a fixed cache location for any given block. Thus, if a program happens to reference words repeatedly from two different blocks that map into the same line, then the blocks will be continually swapped in the cache, and the hit ratio will be low (a phenomenon known as thrashing).
- It has higher miss ratio which is also known as conflict miss. The reason behind this is, even if cache line is empty we cannot allocate insert block into it, because blocks can be inserted into fixed line number in cache.
- The performance of direct mapping cache is not good as requires replacement for data-tag value.

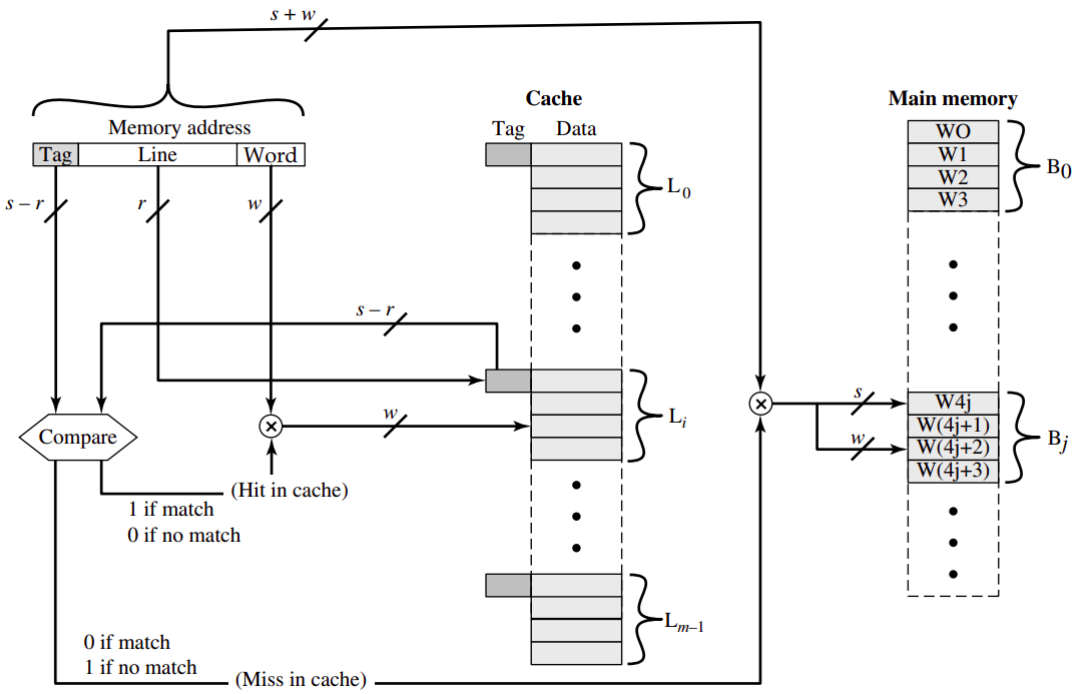

Figure 3: Direct Mapping

One approach to lower the miss penalty is to remember what was discarded in case it is needed again. Since the discarded data has already been fetched, it can be used again at a small cost. Such recycling is possible using a victim cache. Victim cache was originally proposed as an approach to reduce the conflict misses of direct mapped caches without affecting its fast access time. Victim cache is a fully associative cache, whose size is typically 4 to 16 cache lines, residing between a direct mapped L1 cache and the next level of memory.

8.2 Associative/Fully-associative Mapping

Associative mapping overcomes the disadvantage of direct mapping by permitting each main memory block to be loaded into any line of the cache. In this case, the cache control logic interprets a memory address simply as a Tag and a Word field. The Tag field uniquely identifies a block of main memory. To determine whether a block is in the cache, the cache control logic must simultaneously examine every line's tag for a match.

$Address\ length = (s + w) bits$

$Number\ of\ addressable\ units = 2^{s+w}\ words\ or\ bytes$

$Block\ size = line\ size = 2^{w}\ words\ or\ bytes$

$Number\ of\ blocks\ in\ main\ memory = w = 2^{s+w}/2^{w} = 2^{s}$

$Number\ of\ lines\ in\ cache = undetermined$ $\qquad$ $Size\ of\ tage = s\ bits$

Advantages of associative mapping

The principal disadvantage of asso- ciative mapping is the complex circuitry required to examine the tags of all cache lines in parallel.

- With associative mapping, there is flexibility as to which block to replace when a new block is read into the cache. Replacement algorithms are designed to maximize the hit ratio.
- Associative mapping is fast.
- Associative mapping is easy to implement.

Disadvantages of associative mapping

- The principal disadvantage of associative mapping is the complex circuitry required to examine the tags of all cache lines in parallel.

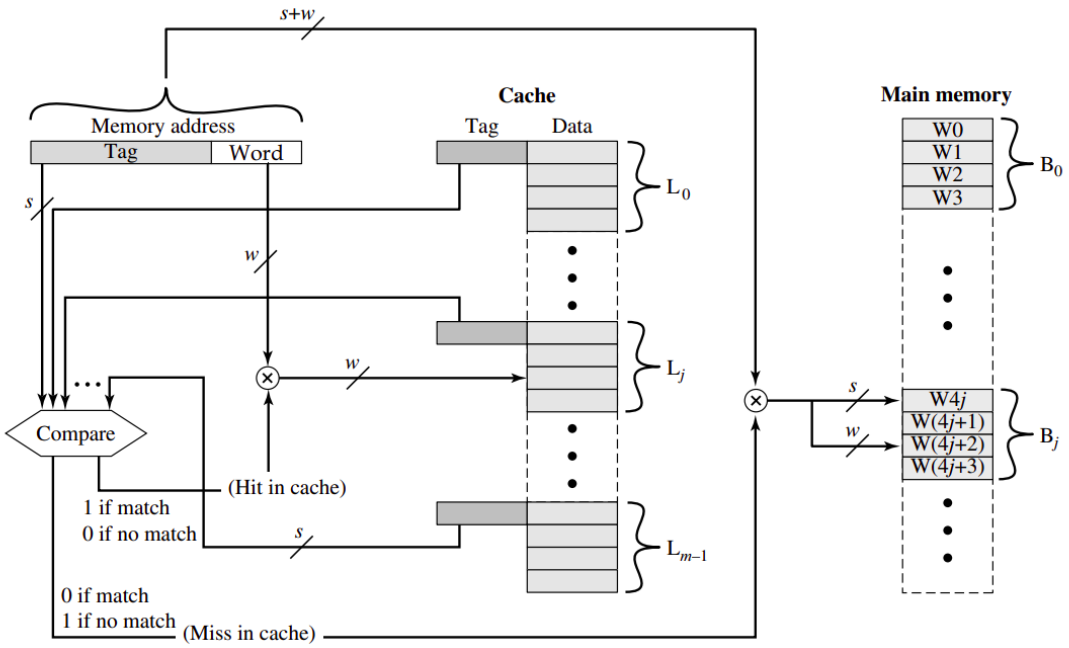

Figure 4: Associative Mapping

- Number of comparison increases.
- Number of bits in tag increases.

8.3 Set-Associative Mapping

In Set-Associative cache memory two or more words can be stored under the same index address.

Here every data word is stored along with its tag. The number of tag-data words under an index is said to form a text.

Advantages of 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 a number sets, each of which consists of a number of lines. The relationships are

$$m = v * k$$

$$i = j \ modulo \ v$$

Where,

$i = cache \ set \ number$
$j = main \ memory \ block \ number$
$m = number \ of \ lines \ in \ the \ cache$
$n = number \ of \ sets$
$k = number \ of \ lines \ in \ each \ set$

Advantages

Set-Associative cache memory has highest hit-ratio compared two previous two cache memory discussed above. Thus its performance is considerably better.

Disadvantages of Set-Associative mapping

Set-Associative cache memory is very expensive. As the set size increases the cost increases.

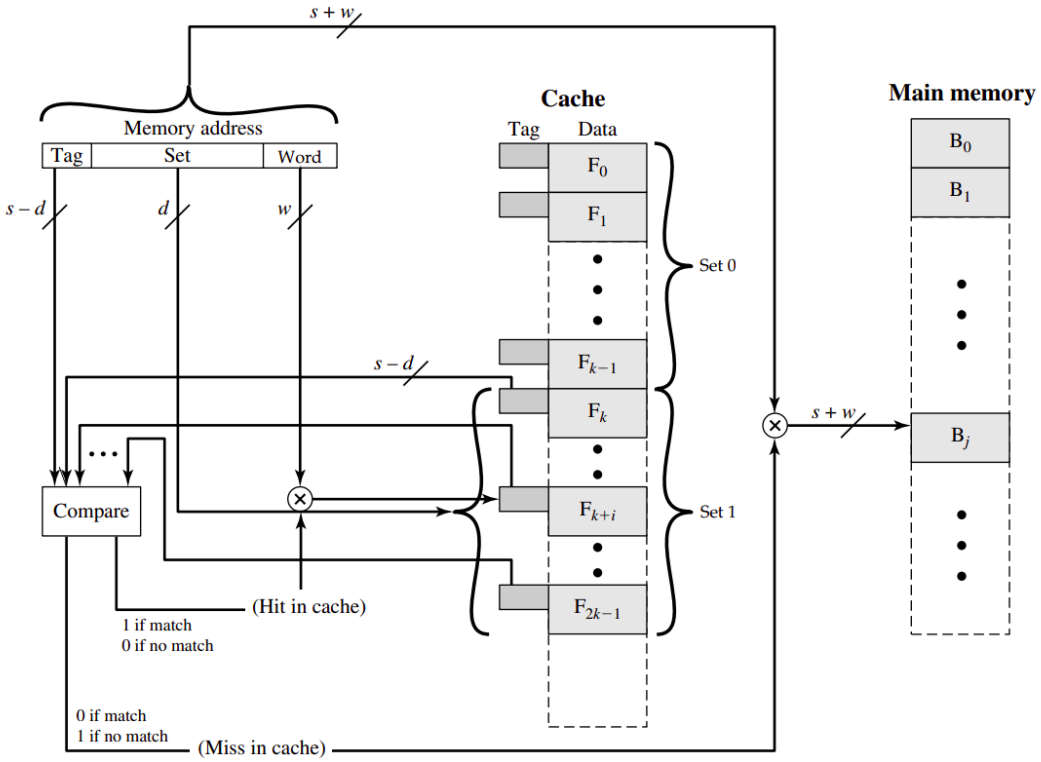

Figure 5: Set-associative Mapping

9 Replacement Algorithms

Once the cache has been filled, when a new block is brought into the cache, one of the existing blocks must be replaced. For direct mapping, there is only one possible line for any particular block, and no choice is possible. For the associative and set-associative techniques, a replacement algorithm is needed. To achieve high speed, such an algorithm must be implemented in hardware.

Following are some of the common and simpler types of cache replacement policies:

- First-in-first-out (FIFO) policy: The earliest inserted item in the cache will be evicted when a new item needs to be inserted.
- Last-in-first-out (LIFO) policy: The last item inserted in the cache will be evicted first.
- Least-recently used (LRU) policy: The item which is least recently used will be evicted first. This is one of the most simple and common cache replacement policies. It assumes that the more recently an item is accessed or used, the more likely it is to be used or accessed again (e.g., switching between tabs of a browser)
- Most-recently used (MRU) policy: The item which is most recently used will be evicted first. This policy is useful, when the chances of repeating the same request in the near future is unlikely (like scrolling through a social media feed or flipping through a photo album).
- Least-frequently-used(LFU) policy: The cache algorithm maintains a counter on the number of times an item in the cache is accessed. It will evict the least frequently accessed item in order to add a new item.
- Random replacement (RR) policy: The cache algorithm randomly selects an item to evict.

10 Write Policy

When a block that is resident in the cache is to be replaced, there are two cases to consider.

If the old block in the cache has not been altered, then it may be overwritten with a new block without first writing out the old block. If at least one write operation has been performed on a word in that line of the cache, then main memory must be updated by writing the line of cache out to the block of memory before bringing in the new block. A variety of write policies, with performance and economic trade-offs, is possible.

There are two problems to contend with. First, more than one device may have access to main memory. For example, an I/O module may be able to read-write directly to memory. If a word has been altered only in the cache, then the corresponding memory word is invalid. Further, if the I/O device has altered main memory, then the cache word is invalid. A more complex problem occurs when multiple processors are attached to the same bus and each processor has its own local cache. Then, if a word is altered in one cache, it could conceivably invalidate a word in other caches.

- **Write Through -** The simplest technique is called **write through.** Using this technique, all write operations are made to main memory as well as to the cache, ensuring that main memory is always valid. Any other processor–cache module can monitor traffic to main memory to maintain consistency within its own cache. The main disadvantage of this technique is that it generates substantial memory traffic and may create a bottleneck.

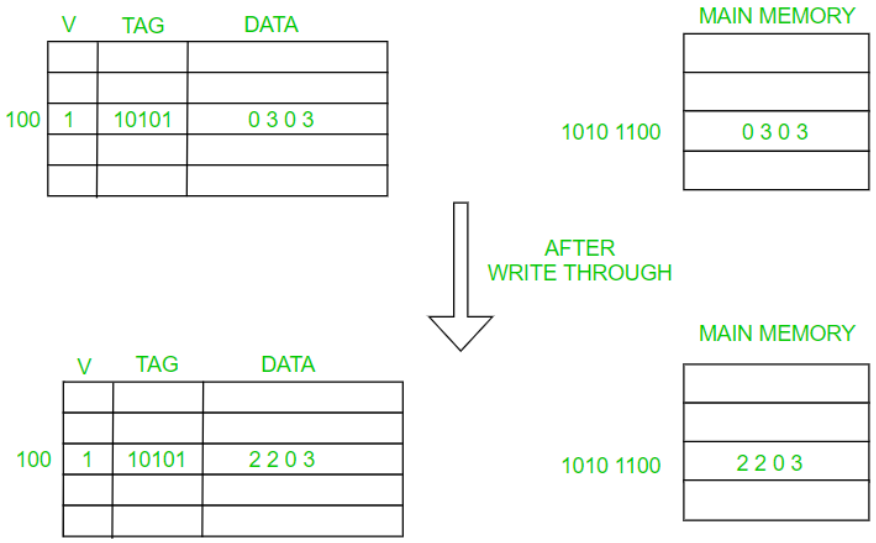

Figure 6: Write Through Policy

- **Write Back -** An alternative technique, known as **write back**, minimizes memory writes. With write back, updates are made only in the cache. When an update occurs, a dirty bit, or use bit, associated with the line is set. Then, when a block is replaced, it is written back to main memory if and only if the dirty bit is set. The problem with write back is that portions of main memory are invalid, and hence accesses by I/O modules can be allowed only through the cache. This makes for complex circuitry and a potential bottleneck.

Write Through Method	Write Back Method
In this method main memory is updated with every memory write operation as well as cache memory is updated in parallel if it contains the word at the specified address.	In this method only cache location is updated during write operation.
Main memory always contains same data as cache.	Main memory and cache memory may have different data.
Number of memory write operation in a typical program is more.	Number of memory write operation in a typical program is less.
When I/O device communicated through DMA would receive most recent data.	When I/O device communicated through DMA would not receive most recent data.
It is a process of writing cache and main memory simultaneously.	It is a process of writing cache and data is removed from cache, first copied to main memory.

Table 2: Difference Between Write Back and Write Through Policies

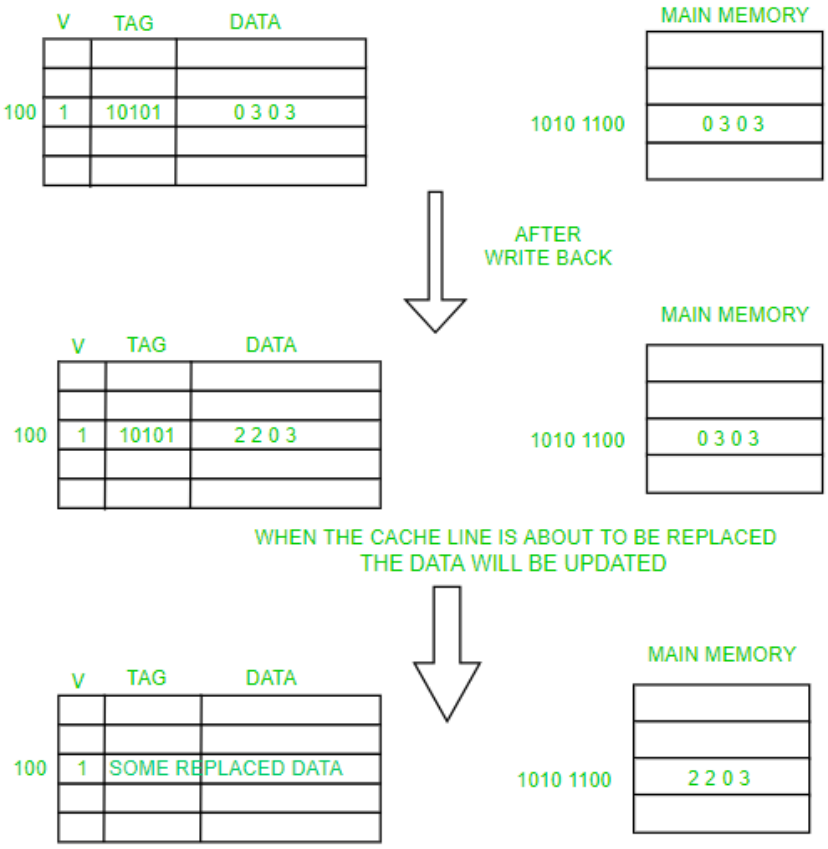

Figure 7: Write Back Policy

11 Cache Coherency

In a bus organization in which more than one device (typically a processor) has a cache and main memory is shared, a new problem is introduced. If data in one cache are altered, this invalidates not only the corresponding word in main memory, but also that same word in other caches (if any other cache happens to have that same word). Even if a write-through policy is used, the other caches may contain in-valid data. A system that prevents this problem is said to maintain cache coherency.

Possible approaches to cache coherency include the following:

- Bus watching with write through: Each cache controller monitors the address lines to detect write operations to memory by other bus masters. If another master writes to a location in shared memory that also resides in the cache memory, the cache controller invalidates that cache entry. This strategy depends on the use of a write-through policy by all cache controllers.
- Hardware transparency: Additional hardware is used to ensure that all updates to main memory via cache are reflected in all caches. Thus, if one processor modifies a word in its cache, this update is written to main memory. In addition, any matching words in other caches are similarly updated.
- Noncacheable memory: Only a portion of main memory is shared by more than one processor, and this is designated as noncacheable. In such a system, all accesses to shared memory are cache misses, because the shared memory is never copied into the cache. The noncacheable memory can be identified using chip-select logic or high-address bits.

12 Virtual Memory

Virtual memory is the partition of logical memory from physical memory. This partition supports large virtual memory for programmers when only limited physical memory is available.

Virtual memory can give programmers the deception that they have a very high memory although the computer has a small main memory. It creates the function of programming easier because the programmer no longer requires to worry about the multiple physical memory available.

Virtual memory works similarly, but at one level up in the memory hierarchy. A memory management unit (MMU) transfers data between physical memory and some gradual storage device, generally a disk. This storage area can be defined as a swap disk or swap file, based on its execution. Retrieving data from physical memory is much faster than accessing data from the swap disk.

There are two primary methods for implementing virtual memory are as follows

Paging

Paging is a technique of memory management where small fixed-length pages are allocated instead of a single large variable-length contiguous block in the case of the dynamic allocation technique. In a paged system, each process is divided into several fixed-size 'chunks' called pages, typically 4k bytes in length. The memory space is also divided into blocks of the equal size known as frames.

Advantages of Paging

There are the following advantages of Paging are:

In Paging, there is no requirement for external fragmentation.

In Paging, the swapping among equal-size pages and page frames is clear.

Paging is a simple approach that it can use for memory management.

Disadvantage of Paging

There are the following disadvantages of Paging are:

In Paging, there can be a chance of Internal Fragmentation.

In Paging, the page table employs more memory.

Because of Multi-level Paging, there can be a chance of memory reference overhead.

Cache Addresses	**Write Policy**
Logical	Write through
Physical	Write back
Cache Size	Write once
Mapping Function	**Line Size**
Direct	**Number of caches**
Associative	Single or two level
Set Associative	Unified or split
Replacement Algorithm	
Least recently used (LRU)	
First in first out (FIFO)	
Least frequently used (LFU)	
Random	

Figure 8: Cache Design Elements

13 Virtual Memory

Virtual Memory is a storage allocation scheme in which secondary memory can be addressed as though it were part of the main memory. The addresses a program may use to reference memory are distinguished from the addresses the memory system uses to identify physical storage sites, and program-generated addresses are translated automatically to the corresponding machine addresses.

The size of virtual storage is limited by the addressing scheme of the computer system and the amount of secondary memory is available not by the actual number of the main storage locations.

Dynamically translated into physical addresses at run time. This means that a process can be swapped in and out of the main memory such that it occupies different places in the main memory at different times during the course of execution. A process may be broken into a number of pieces and these pieces need not be continuously located in the main memory during execution. The combination of dynamic run-time address translation and use of page or segment table permits this. If these characteristics are present then, it is not necessary that all the pages or segments are present in the main memory during execution. This means that the required pages need to be loaded into memory whenever required. Virtual memory is implemented using Demand Paging or Demand Segmentation.

It is a technique that is implemented using both hardware and software. It maps memory addresses used by a program, called virtual addresses, into physical addresses in computer memory.

14 Memory module and Interleaving

Memory Interleaving is less or More an Abstraction technique. Though it's a bit different from Abstraction. It is a Technique that divides memory into a number of modules such that Successive words in the address space are placed in the Different modules.

14.1 Consecutive Word in a Module

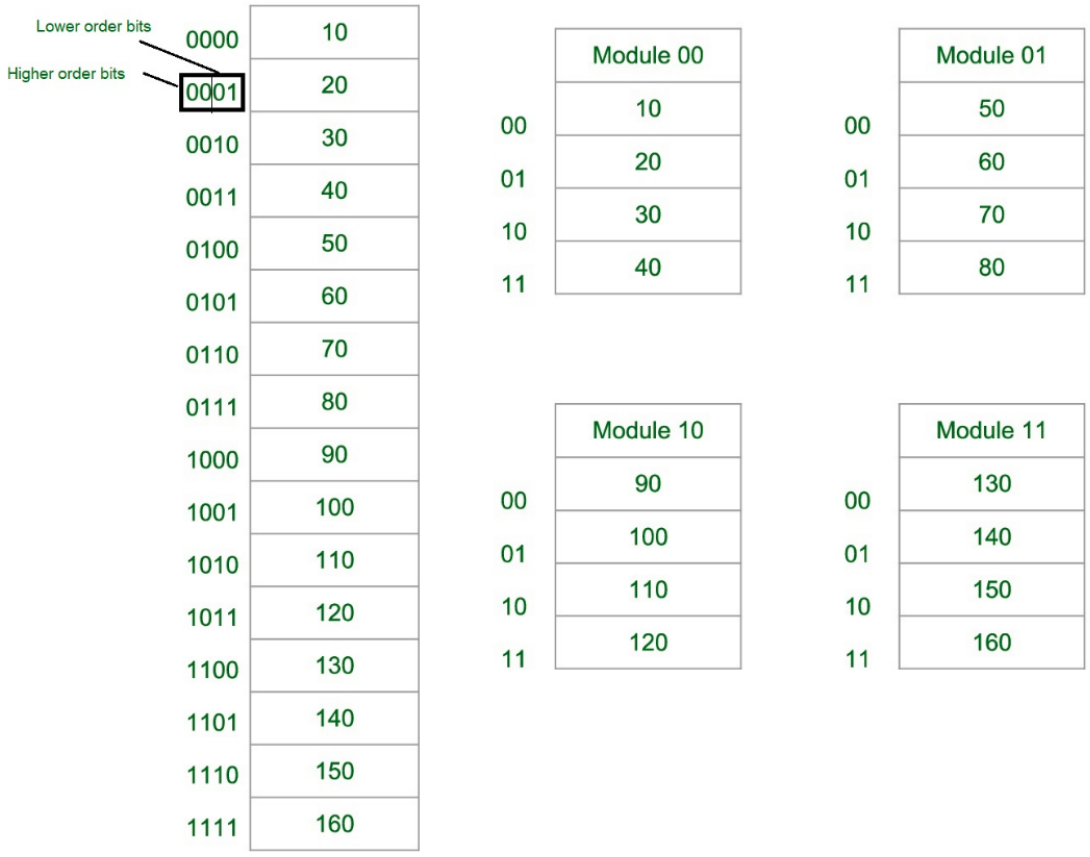

Figure 9: Consecutive word in a module

Let us assume 16 Data's to be Transferred to the Four Module. Where Module 00 be Module 1, Module 01 be Module 2, Module 10 be Module 3 & Module 11 be Module 4. Also, 10, 20, 30. . . .130 are the data to be transferred.

From the Figure 9 in Module 1, 10 [Data] is transferred then 20, 30 & finally, 40 which are the Data. That means the data are added consecutively in the Module till its max capacity.

Most significant bit (MSB) provides the Address of the Module & the least significant bit (LSB) provides the address of the data in the module.

For Example, to get 90 (Data) 1000 will be provided by the processor. This 10 will indicate that the data is in module 10 (module 3) & 00 is the address of 90 in Module 10 (module 3). So,

Module 1 Contains Data : 10, 20, 30, 40
Module 2 Contains Data : 50, 60, 70, 80
Module 3 Contains Data : 90, 100, 110, 120
Module 4 Contains Data : 130, 140, 150, 160

14.2 Consecutive Word in Consecutive Module:

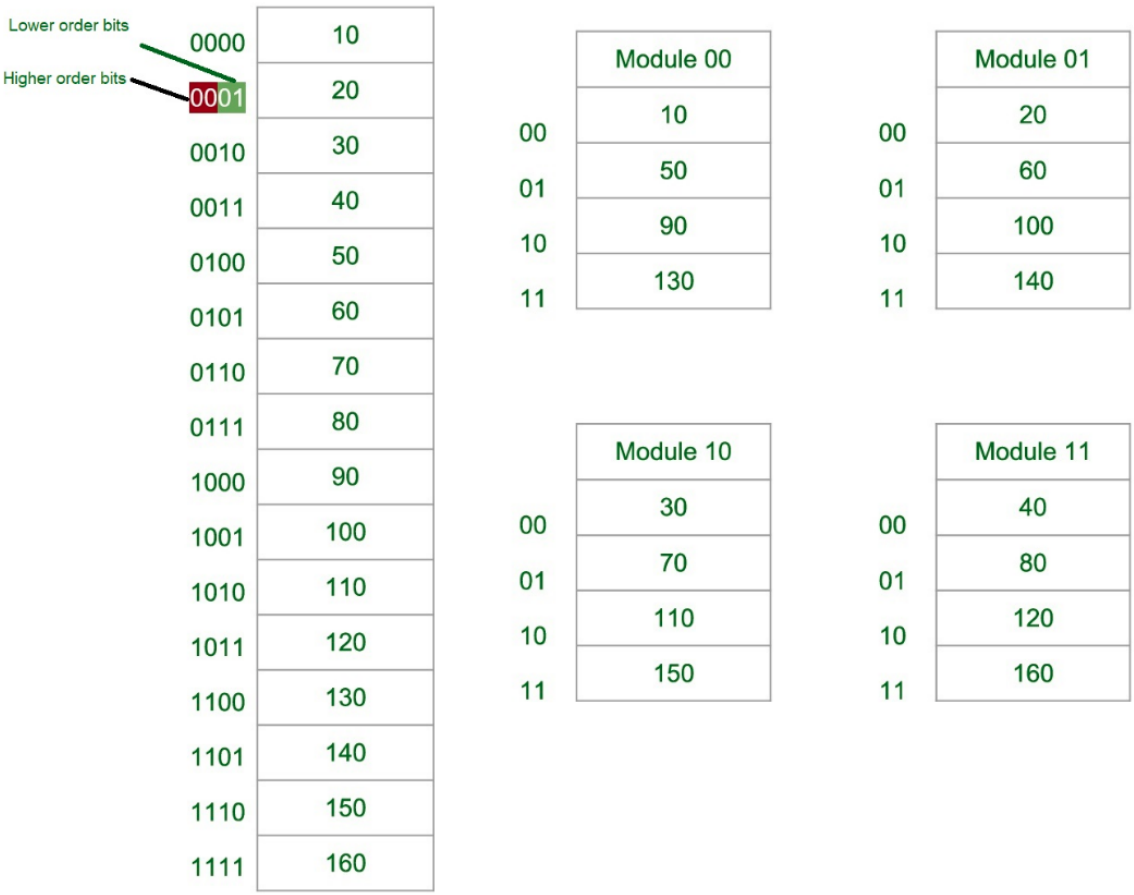

Figure 10: Consecutive word in consecutive module

Now again we assume 16 Data's to be transferred to the Four Module. But Now the consecutive Data are added in Consecutive Module. That is, 10 [Data] is added in Module 1, 20 [Data] in Module 2 and So on.

Least Significant Bit (LSB) provides the Address of the Module & Most significant bit (MSB) provides the address of the data in the module.

For Example, to get 90 (Data) 1000 will be provided by the processor. This 00 will indicate that the data is in module 00 (module 1) & 10 is the address of 90 in Module 00 (module 1). That is,

Module 1 Contains Data : 10, 50, 90, 130
Module 2 Contains Data : 20, 60, 100, 140
Module 3 Contains Data : 30, 70, 110, 150
Module 4 Contains Data : 40, 80, 120, 160

Why do we use Memory Interleaving? [Advantages]: Whenever Processor requests Data from the main memory. A block (chunk) of Data is Transferred to the cache and then to Processor. So whenever a cache miss occurs the Data is to be fetched from the main memory. But main memory is relatively slower than the cache. So to improve the access time of the main memory interleaving is used.

We can access all four Modules at the same time thus achieving Parallelism. From Figure 10 the data can be acquired from the Module using the Higher bits. This method Uses memory effectively.

15 Associative Memory

An associative memory can be treated as a memory unit whose saved information can be recognized for approach by the content of the information itself instead of by an address or memory location. Associative memory is also known as Content Addressable Memory (CAM).

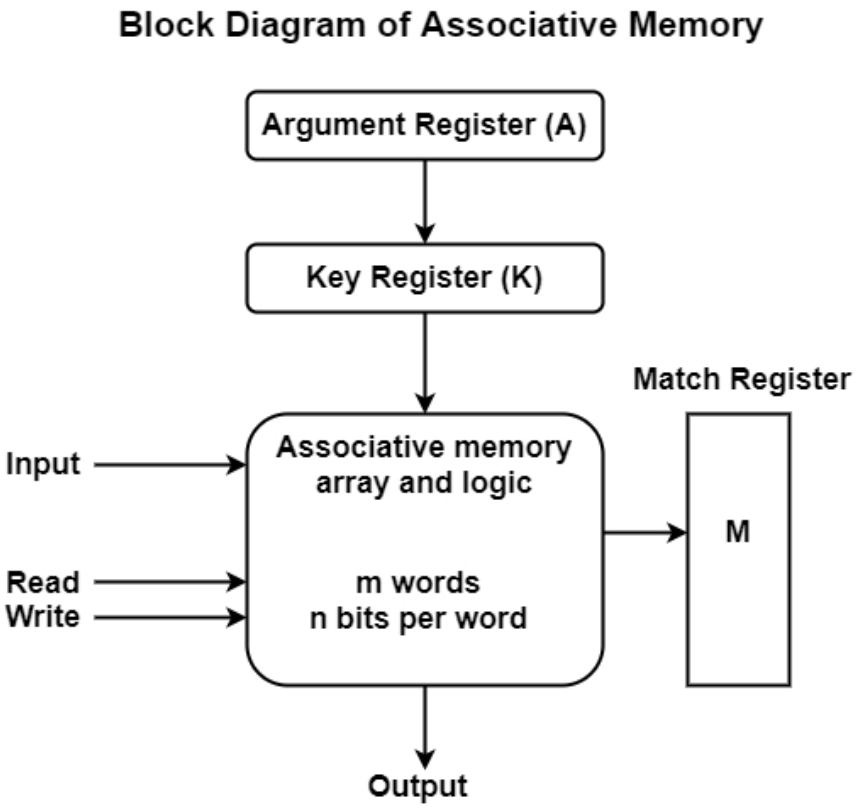

Figure 11: Associative Memory

The block diagram of associative memory is shown in the Figure 11. It includes a memory array and logic for m words with n bits per word. The argument register A and key register K each have n bits, one for each bit of a word.

The match register M has m bits, one for each memory word. Each word in memory is related in parallel with the content of the argument register.

The words that connect the bits of the argument register set an equivalent bit in the match register. After the matching process, those bits in the match register that have been set denote the fact that their equivalent words have been connected.

Reading is proficient through sequential access to memory for those words whose equivalent bits in the match register have been set.

The key register supports a mask for selecting a specific field or key in the argument word. The whole argument is distinguished with each memory word if the key register includes all 1's.

Hence, there are only those bits in the argument that have 1's in their equivalent position of the key register are compared. Therefore, the key gives a mask or recognizing a piece of data that determines how the reference to memory is created.

The following Figure 12 can define the relation between the memory array and the external registers in associative memory.

The cells in the array are considered by the letter C with two subscripts. The first subscript provides the word number and the second determines the bit position in the word. Therefore cell C_{ij} is the cell for bit j in word i.

A bit in the argument register is compared with all the bits in column j of the array supported that $K_j = 1$. This is completed for all columns j = 1, 2 . . . , n.

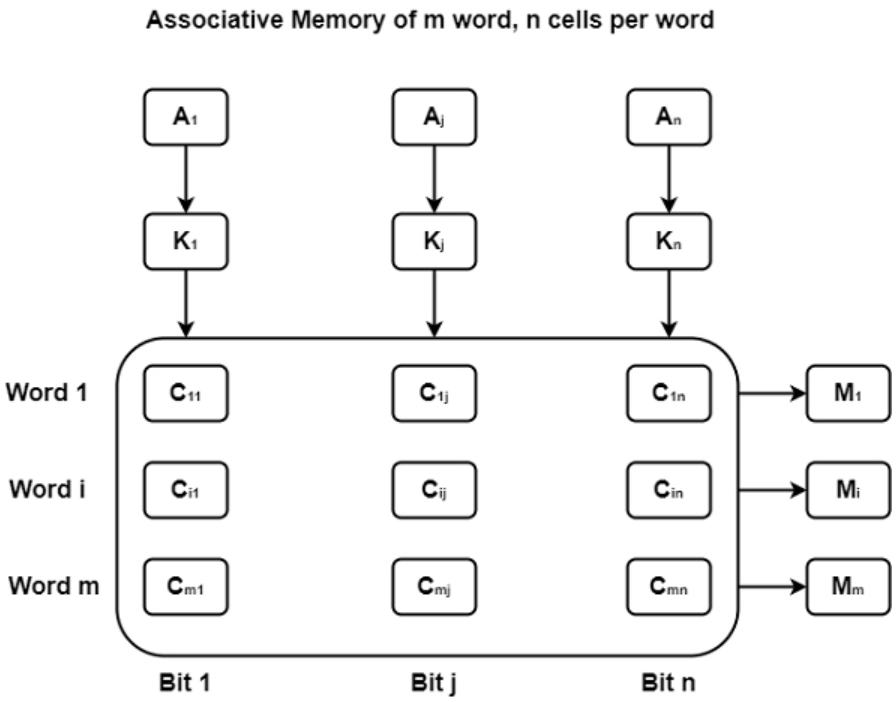

Figure 12: Memory Array

If a match appears between all the unmasked bits of the argument and the bits in word i, the equivalent bit M_i in the match register is set to 1. If one or more unmasked bits of the argument and the word do not match, M_i is cleared to 0.

Associative memory of conventional semiconductor memory (usually RAM) with added comparison circuity that enables a search operation to complete in a single clock cycle. It is a hardware search engine, a special type of computer memory used in certain very high searching applications.

Applications of Associative memory :-

It can be only used in memory allocation format. It is widely used in the database management systems, etc.

Advantages of Associative memory :-

It is used where search time needs to be less or short. It is suitable for parallel searches. It is often used to speedup databases. It is used in page tables used by the virtual memory and used in neural networks.

Disadvantages of Associative memory :-

It is more expensive than RAM. Each cell must have storage capability and logical circuits for matching its content with external argument.

16 Auxiliary Memory

An Auxiliary memory is referred to as the lowest-cost, highest-space, and slowest-approach storage in a computer system. It is where programs and information are preserved for long-term storage or when not in direct use. The most typical auxiliary memory devices used in computer systems are magnetic disks and tapes.

Magnetic Disks

A magnetic disk is a round plate generated of metal or plastic coated with magnetized material. There are both sides of the disk are used and multiple disks can be stacked on one spindle with read/write heads accessible on each surface.

All disks revolve together at high speed and are not stopped or initiated for access purposes. Bits are saved in the magnetized surface in marks along concentric circles known as tracks. The tracks are frequently divided into areas known as sectors.

In this system, the lowest quantity of data that can be sent is a sector. The subdivision of one disk surface into tracks and sectors is displayed in the Figure 13.

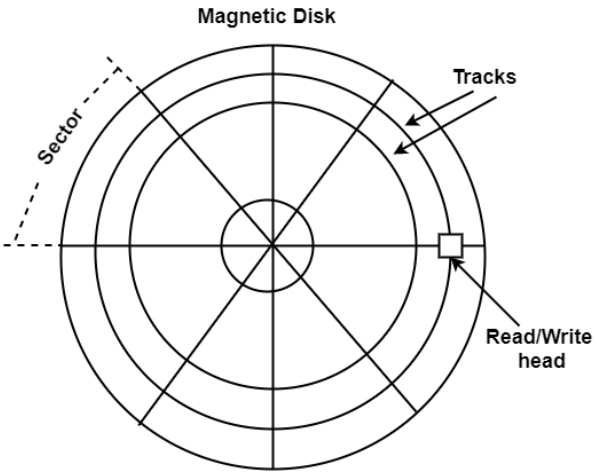

Figure 13: Magnetic Disk

17 Magnetic Tape

Magnetic tape transport includes the robotic, mechanical, and electronic components to support the methods and control structure for a magnetic tape unit. The tape is a layer of plastic coated with a magnetic documentation medium.

Bits are listed as a magnetic stain on the tape along various tracks. There are seven or nine bits are recorded together to form a character together with a parity bit. Read/write heads are mounted one in each track therefore that information can be recorded and read as a series of characters.

Magnetic tape units can be stopped, initiated to move forward, or in the opposite, or it can be reversed. However, they cannot be initiated or stopped fast enough between single characters. For this reason, data is recorded in blocks defined as records. Gaps of unrecorded tape are added between records where the tape can be stopped.

The tape begins affecting while in a gap and achieves its permanent speed by the time it arrives at the next record. Each record on tape has a recognition bit design at the starting and end. By reading the bit design at the starting, the tape control recognizes the data number.

17.1 Memory allocation strategies

First Fit Allocation
These are Contiguous memory allocation techniques. First-Fit Memory Allocation: This method keeps the free/busy list of jobs organized by memory location, low-ordered to high-ordered memory. In this method, first job claims the first available memory with space more than or equal to it's size. The operating system doesn't search for appropriate partition but just allocate the job to the nearest memory partition available with sufficient size.

Advantages of First-Fit Allocation in Operating Systems:

- Simple and efficient search algorithm
- Minimizes memory fragmentation
- Fast allocation of memory

Disadvantages of First-Fit Allocation in Operating Systems:

- Poor performance in highly fragmented memory
- May lead to poor memory utilization
- May allocate larger blocks of memory than required.

Best-Fit Allocation in Operating System
Best-Fit Memory Allocation: This method keeps the free/busy list in order by size – smallest to largest. In this method, the operating system first searches the whole of the memory according to the size of the given job and allocates it to the closest-fitting free partition in the memory, making it able to use memory efficiently. Here the jobs are in the order from smallest job to largest job.

Advantages of Best-Fit Allocation :

- Memory Efficient. The operating system allocates the job minimum possible space in the memory, making memory management very efficient.
- To save memory from getting wasted, it is the best method.
- Improved memory utilization
- Reduced memory fragmentation
- Minimizes external fragmentation

Disadvantages of Best-Fit Allocation :

- It is a Slow Process. Checking the whole memory for each job makes the working of the operating system very slow. It takes a lot of time to complete the work.
- Increased computational overhead
- May lead to increased internal fragmentation
- Can result in slow memory allocation times.

Worst-Fit Allocation

In this allocation technique, the process traverses the whole memory and always search for the largest hole/partition, and then the process is placed in that hole/partition. It is a slow process because it has to traverse the entire memory to search the largest hole.

Advantages of Worst-Fit Allocation :

Since this process chooses the largest hole/partition, therefore there will be large internal fragmentation. Now, this internal fragmentation will be quite big so that other small processes can also be placed in that leftover partition.

Disadvantages of Worst-Fit Allocation :

It is a slow process because it traverses all the partitions in the memory and then selects the largest partition among all the partitions, which is a time-consuming process.