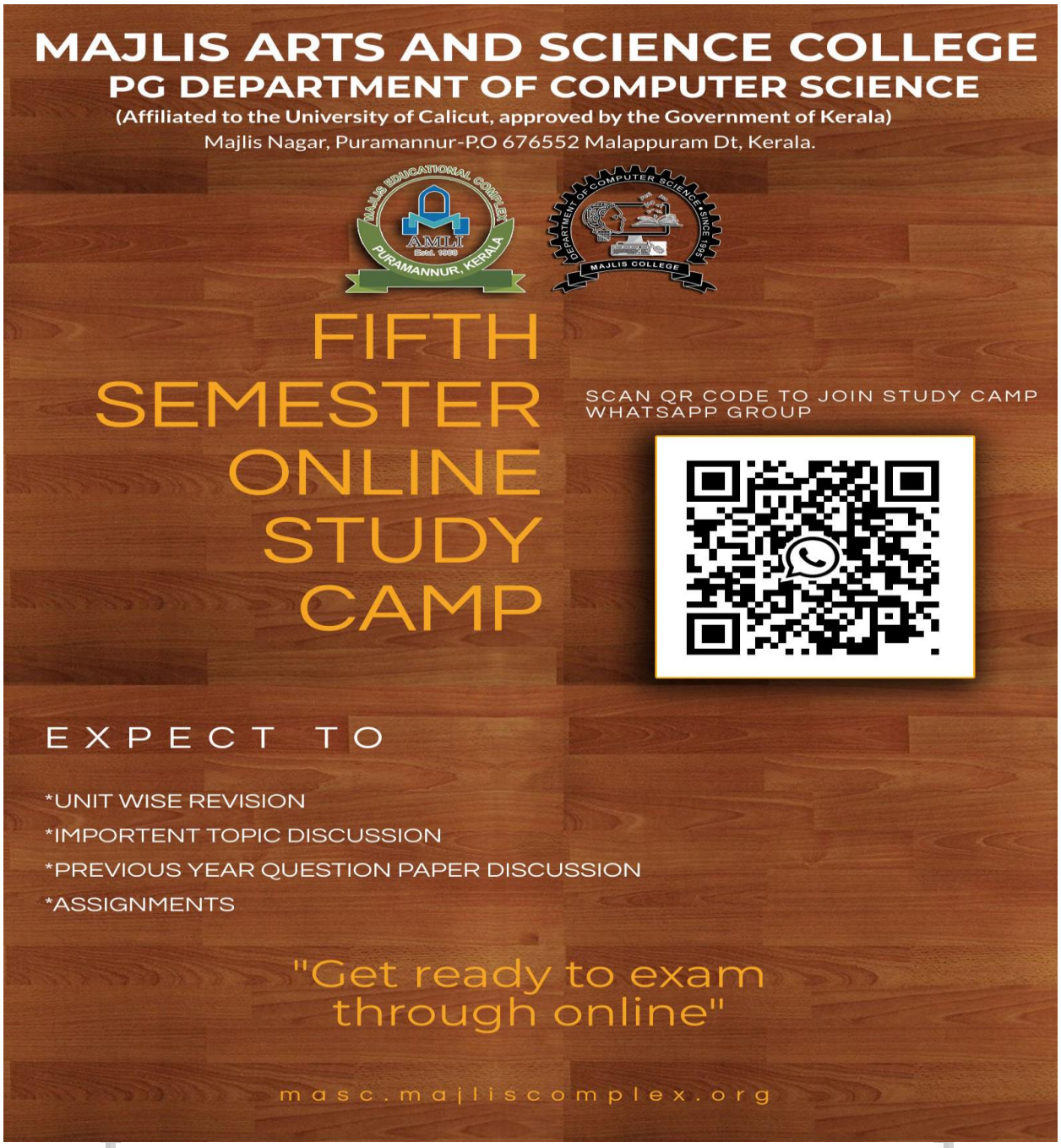
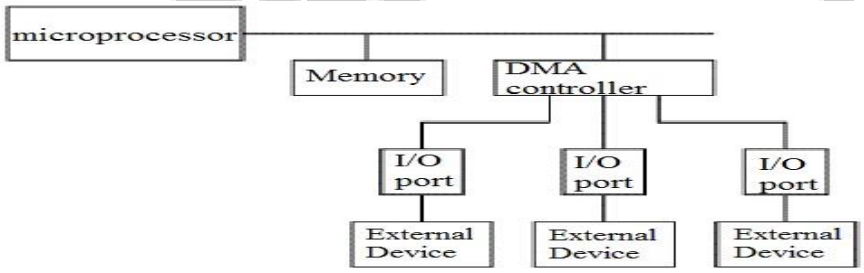


COMPUTER ORGANIZATION AND ARCHITECTURE
MODULE 5

1. Explain the basic principle and working of DMA.

The term DMA stands for direct memory access. The hardware device used for direct memory access is called the DMA controller. DMA controller is a control unit, part of I/O device's interface circuit, which can transfer blocks of data between I/O devices and main memory with minimal intervention fromtheprocessor.

DMA controller provides an interface between the bus and the input-output devices. Although it transfers data without intervention of processor, it is controlled by the processor. The processor initiates the DMA controller by sending the starting address, Number of words in the data block and direction of transfer of data .i.e. from I/O devices to the memory or from main memory to I/O devices. More than one external device can be connected to the DMA controller.



DMA controller contains an address unit, for generating addresses and selecting I/O device for transfer. It also contains the control unit and data count for keeping counts of the number of blocks transferred and indicating the direction of transfer of data. When the transfer is completed, DMA informs the processor by raising an interrupt.

2. Explain the procedure to initiate DMA by the CPU

The first information is whether the data has to be read from memory or the data has to be written to the memory. It passes this information via **read or write control lines** that is between the processor and DMA controllers **control logic unit**.

The processor also provides the **starting address** of/ for the data block in the memory, from where the data block in memory has to be read or where the data block has to be written in memory. DMA controller stores this in its **address register**. It is also called the **starting address register**.

The processor also sends the **word count**, i.e. how many words are to be read or written. It stores this information in the **data count** or the **word count** register.

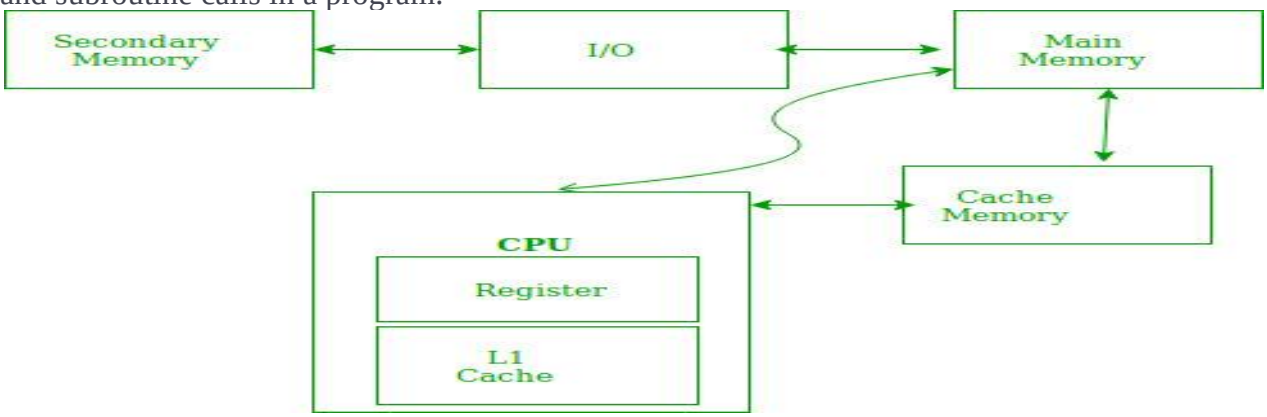
3. Mention the advantages of memory.

Cache **memory** is faster than main **memory**. It consumes less access time as compared to main **memory**. It stores the program that can be executed within a short period of time. It stores data for temporary use.

4. Explain locality of reference.

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

addresses are near one another. The property of locality of reference is mainly shown by loops and subroutine calls in a program.

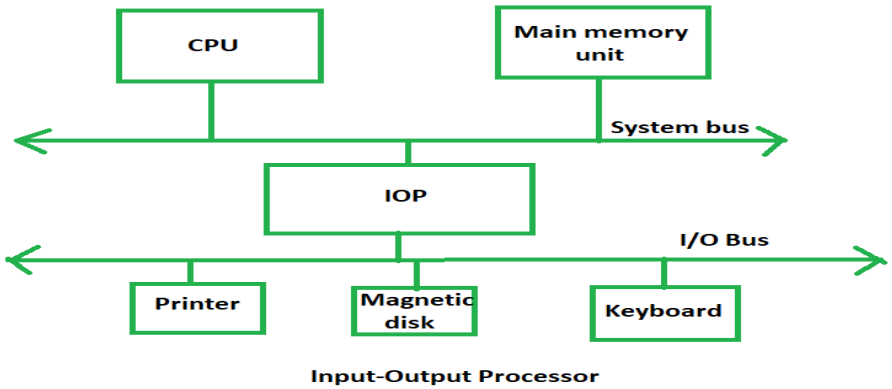


1. In case of loops in program control processing unit repeatedly refers to the set of instructions that constitute the loop.
2. In case of subroutine calls, everytime the set of instructions are fetched from memory.
3. References to data items also get localized that means same data item is referenced again and again.

5. Write a note on I/O processors

The Input Output Processor (IOP) is just like a CPU that handles the details of I/O operations. It is more equipped with facilities than those are available in typical DMA controller. The IOP can fetch and execute its own instructions that are specifically designed to characterize I/O transfers. In addition to the I/O – related tasks, it can perform other processing tasks like arithmetic, logic, branching and code translation. The main memory unit takes the pivotal role. It communicates with processor by the means of DMA.

The block diagram –



Input-Output Processor

The Input Output Processor is a specialized processor which loads and stores data into memory along with the execution of I/O instructions. It acts as an interface between system and devices. It involves a sequence of events to executing I/O operations and then store the results into the memory.

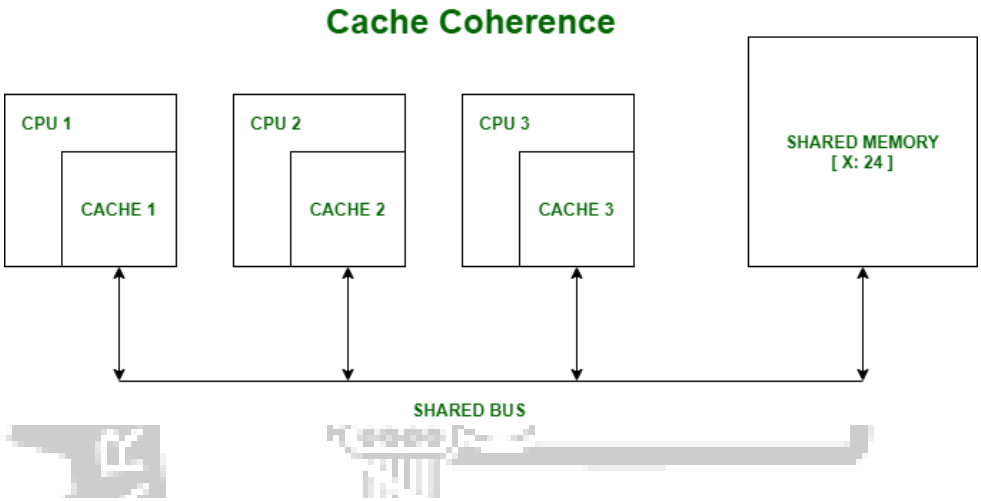
Advantages –

- The I/O devices can directly access the main memory without the intervention by the processor in I/O processor based systems.

- It is used to address the problems that are arises in Direct memory access method.

6. explain cache coherence

In a multiprocessor system, data inconsistency may occur among adjacent levels or within the same level of the memory hierarchy.
In a shared memory multiprocessor with a separate cache memory for each processor, it is possible to have many copies of any one instruction operand: one copy in the main memory and one in each cache memory. When one copy of an operand is changed, the other copies of the operand must be changed also.
Example:
Cache and the main memory may have inconsistent copies of the same object.



Suppose there are three processors, each having cache. Suppose the following scenario:-
- **Processor 1 read X :** obtains 24 from the memory and caches it.
- **Processor 2 read X :** obtains 24 from memory and caches it.
- **Again, processor 1 writes as X :** 64, Its locally cached copy is updated. Now, processor 3 reads X, what value should it get?
- Memory and processor 2 thinks it is 24 and processor 1 thinks it is 64.

As multiple processors operate in parallel, and independently multiple caches may possess different copies of the same memory block, this creates a cache coherence problem.
Cache coherence is the discipline that ensures that changes in the values of shared operands are propagated throughout the system in a timely fashion.

7. Discuss about Asynchronous Data Transfer.

if the registers in the interface(I/O interface) share a common clock with CPU registers, then transfer between the two units is said to be synchronous.But in most cases, the internal timing in each unit is independent from each other in such a way that each uses its own private clock for its internal registers.In that case, the two units are said to be asynchronous to each other, and if data transfer occur between them this data transfer is said to be **Asynchronous Data Transfer**.

But, the Asynchronous Data Transfer between two independent units requires that control signals be transmitted between the communicating units so that the time can be indicated at which they send data.

This asynchronous way of data transfer can be achieved by two methods:

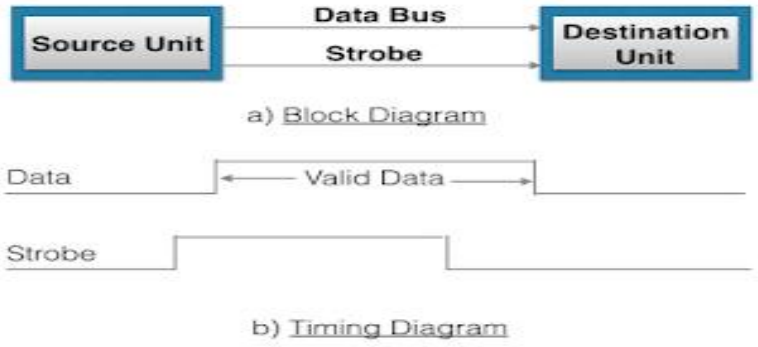
1. One way is by means of strobe pulse which is supplied by one of the units to other unit.When transfer has to occur.This method is known as "**Strobe Control**".
2. Another method commonly used is to accompany each data item being transferred with a control signal that indicates the presence of data in the bus.The unit receiving the data item responds with another signal to acknowledge receipt of the data.This method of data transfer between two independent units is said to be "**Handshaking**".

Strobe Control:

The Strobe Control method of asynchronous data transfer employs a single control line to time each transfer .This control line is also known as strobe and it may be achieved either by source or destination, depending on which initiate transfer.

Source initiated strobe for data transfer:

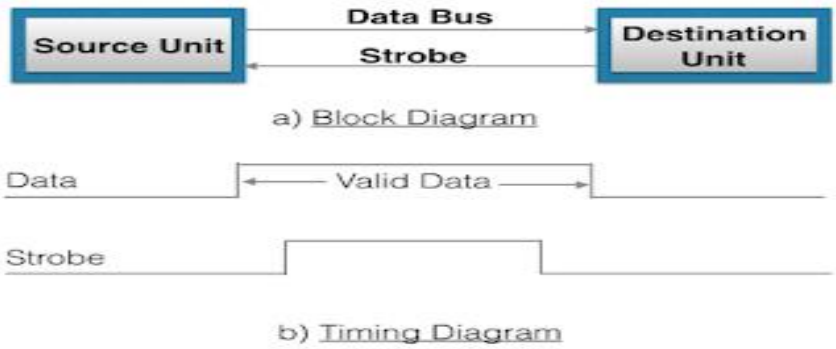
The block diagram and timing diagram of strobe initiated by source unit is shown in figure below:



In block diagram we see that strobe is initiated by source, and as shown in timing diagram, the source unit first places the data on the data bus.After a brief delay to ensure that the data settle to a steady value, the source activates a strobe pulse.The information on data bus and strobe control signal remain in the active state for a sufficient period of time to allow the destination unit to receive the data.Actually, the destination unit, uses a falling edge of strobe control to transfer the contents of data bus to one of its internal registers.The source removes the data from the data bus after it disables its strobe pulse.New valid data will be available only after the strobe is enabled again.

Destination-initiated strobe for data transfer:

The block diagram and timing diagram of strobe initiated by destination is shown in figure below:



In block diagram, we see that, the strobe initiated by destination, and as shown in timing diagram, the destination unit first activates the strobe pulse, informing the source to provide the data.The source unit responds by placing the requested binary information on the data bus.The data must be valid and remain in the bus long enough for the destination unit to accept it.The falling edge of strobe pulse can be used again to trigger a destination register.The destination unit then disables the strobe.And source removes the data from data bus after a per determine time interval.

2. Handshaking:

The disadvantage of strobe method is that source unit that initiates the transfer has no way of knowing whether the destination has actually received the data that was placed in the bus.Similarly, a destination unit that initiates the transfer has no way of knowing whether the source unit, has actually placed data on the bus.

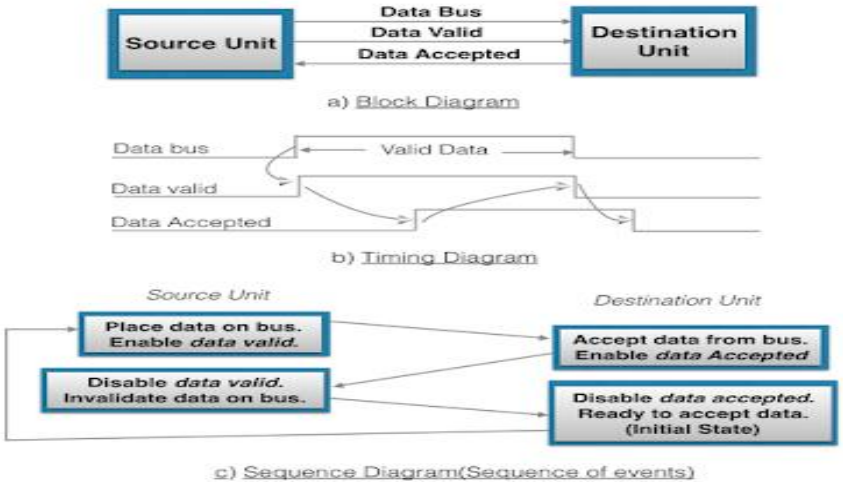
This problem can be solved by handshaking method.

Hand shaking method introduce a second control signal line that provides a replay to the unit that initiates the transfer.

In it, one control line is in the same direction as the data flow in the bus from the source to destination.It is used by source unit to inform the destination unit whether there are valid data in the bus.The other control line is in the other direction from destination to the source.It is used by the destination unit to inform the source whether it can accept data.And in it also, sequence of control depends on unit that initiate transfer.Means sequence of control depends whether transfer is initiated by source and destination.Sequence of control in both of them are described below:

Source initiated Handshaking:

The source initiated transfer using handshaking lines is shown in figure below:

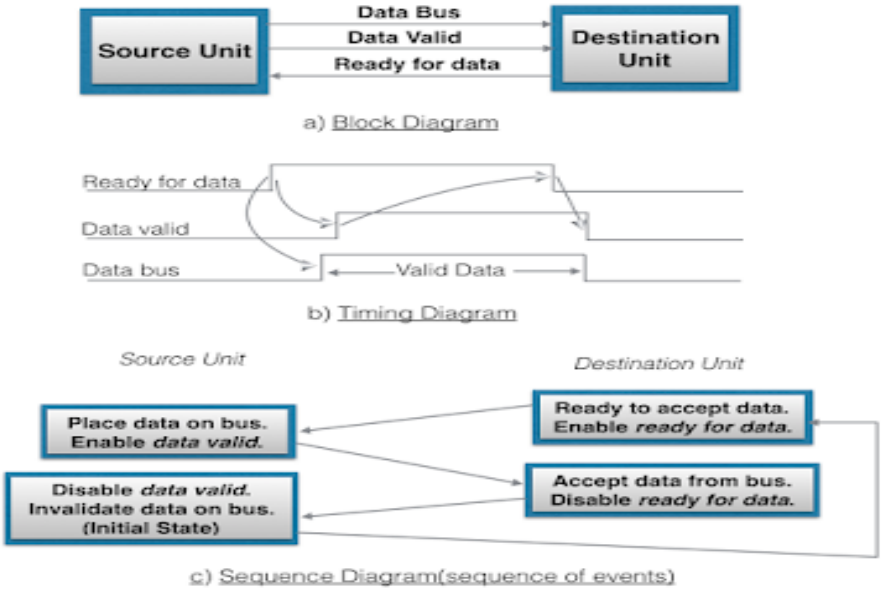


In its block diagram, we se that two handshaking lines are "data valid", which is generated by the source unit, and "data accepted", generated by the destination unit.

The timing diagram shows the timing relationship of exchange of signals between the two units.Means as shown in its timing diagram, the source initiates a transfer by placing data on the bus and enabling its data valid signal.The data accepted signal is then activated by destination unit after it accepts the data from the bus.The source unit then disable its data valid signal which invalidates the data on the bus.After this, the destination unit disables its data accepted signal and the system goes into initial state.The source unit does not send the next data item until after the destination unit shows its readiness to accept new data by disabling the data accepted signal.

Destination initiated handshaking:

The destination initiated transfer using handshaking lines is shown in figure below:



In its block diagram, we see that the two handshaking lines are "data valid", generated by the source unit, and "ready for data" generated by destination unit.Note that the name of signal data accepted generated by destination unit has been changed to ready for data to reflect its new meaning.

In it, transfer is initiated by destination, so source unit does not place data on data bus until it receives ready for data signal from destination unit.After that, hand shaking process is some as that of source initiated.

The sequence of event in it are shown in its sequence diagram and timing relationship between signals is shown in its timing diagram.

8. What is an interrupt?

   An **interrupt** is a signal sent to the processor that **interrupts** the current process. It may be generated by a hardware device or a software program.

9. Explain how priority interrupts are served

Priority Interrupt

A priority interrupt is a system which decides the priority at which various devices, which generates the interrupt signal at the same time, will be serviced by the CPU. The system has authority to decide which conditions are allowed to interrupt the CPU, while some other interrupt is being serviced. Generally, devices with high speed transfer such as *magnetic disks* are given high priority and slow devices such as *keyboards* are given low priority.
When two or more devices interrupt the computer simultaneously, the computer services the device with the higher priority first.

Types of Interrupts:

Following are some different types of interrupts:

Hardware Interrupts

When the signal for the processor is from an external device or hardware then this interrupts is known as **hardware interrupt**.
Let us consider an example: when we press any key on our keyboard to do some action, then this pressing of the key will generate an interrupt signal for the processor to perform certain action. Such an interrupt can be of two types:

- **Maskable Interrupt**

  The hardware interrupts which can be delayed when a much high priority interrupt has occurred at the same time.

- **Non Maskable Interrupt**

  The hardware interrupts which cannot be delayed and should be processed by the processor immediately.

Software Interrupts

The interrupt that is caused by any internal system of the computer system is known as a **software interrupt**. It can also be of two types:

- **Normal Interrupt**

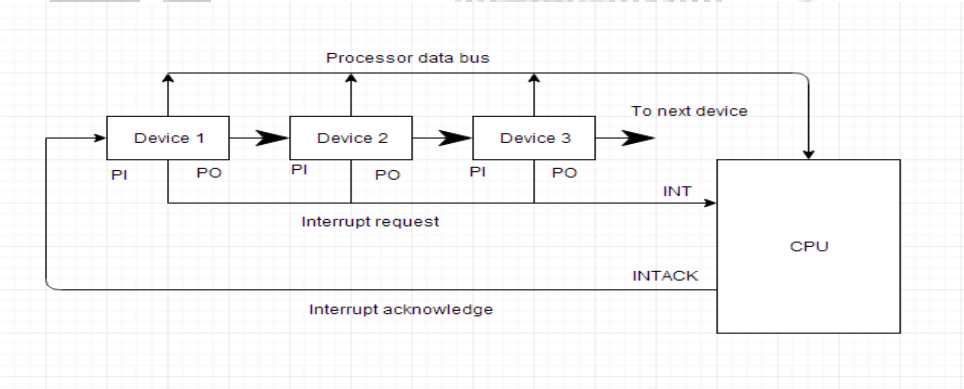
  The interrupts that are caused by software instructions are called **normal software interrupts**.

- **Exception**

  Unplanned interrupts which are produced during the execution of some program are called **exceptions**, such as division by zero.

Daisy Chaining Priority

This way of deciding the interrupt priority consists of serial connection of all the devices which generates an interrupt signal. The device with the highest priority is placed at the first position followed by lower priority devices and the device which has lowest priority among all is placed at the last in the chain.
In daisy chaining system all the devices are connected in a serial form. The interrupt line request is common to all devices. If any device has interrupt signal in low level state then interrupt line goes to low level state and enables the interrupt input in the CPU. When there is no interrupt the interrupt line stays in high level state. The CPU respond to the interrupt by enabling the interrupt acknowledge line. This signal is received by the device 1 at its PI input. The acknowledge signal passes to next device through PO output only if device 1 is not requesting an interrupt.
The following figure shows the block diagram for daisy chaining priority system.



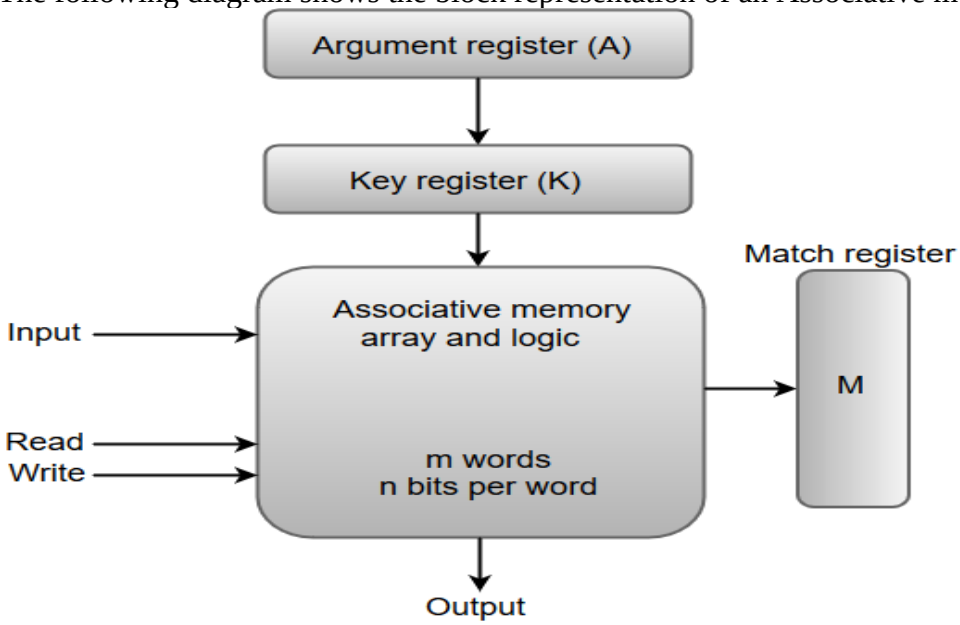
10. describe in detail about associative memory

An associative memory can be considered as a memory unit whose stored data can be identified for access by the content of the data itself rather than by an address or memory location. Associative memory is often referred to as **Content Addressable Memory (CAM)**.

When a write operation is performed on associative memory, no address or memory location is given to the word. The memory itself is capable of finding an empty unused location to store the word.

when the word is to be read from an associative memory, the content of the word, or part of the word, is specified. The words which match the specified content are located by the memory and are marked for reading.

The following diagram shows the block representation of an Associative memory.



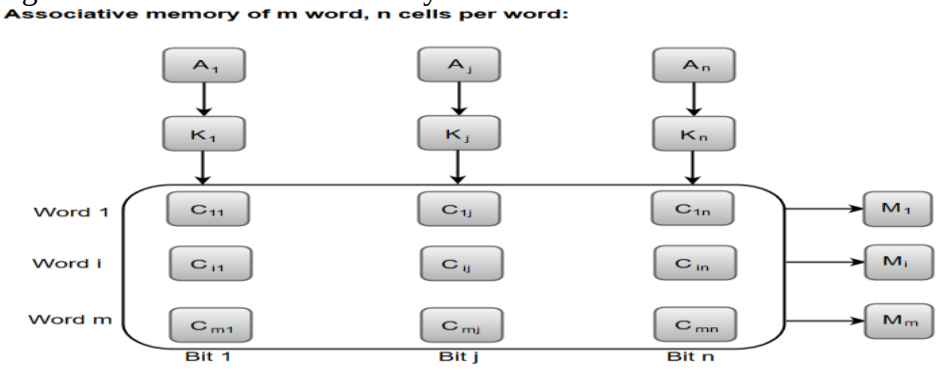
From the block diagram, we can say that an associative memory consists of a memory array and logic for 'm' words with 'n' bits per word.

The functional registers like the argument register **A** and key register **K** each have **n** bits, one for each bit of a word. The match register **M** consists of **m** bits, one for each memory word.

The words which are kept in the memory are compared in parallel with the content of the argument register.

The key register (K) provides a mask for choosing a particular field or key in the argument word. If the key register contains a binary value of all 1's, then the entire argument is compared with each memory word. Otherwise, only those bits in the argument that have 1's in their corresponding position of the key register are compared. Thus, the key provides a mask for identifying a piece of information which specifies how the reference to memory is made.

The following diagram can represent the relation between the memory array and the external registers in an associative memory



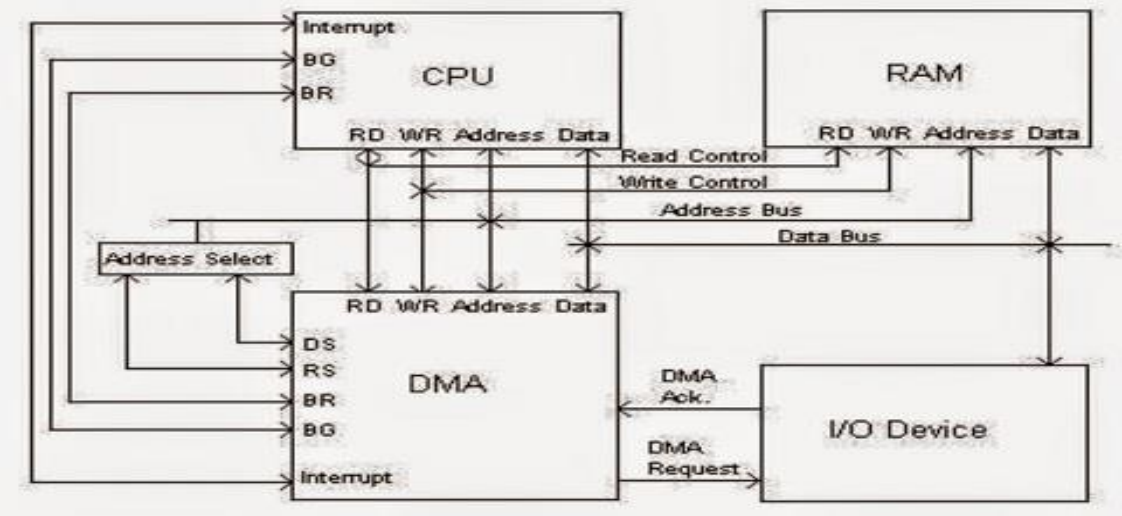
The cells present inside the memory array are marked by the letter C with two subscripts. The first subscript gives the word number and the second specifies the bit position in the word. For instance, the cell C_{ij} is the cell for bit **j** in word **i**.
A bit A_j in the argument register is compared with all the bits in column **j** of the array provided that $K_j = 1$. This process is done for all columns **j** = 1, 2, 3....., n.
If a match occurs between all the unmasked bits of the argument and the bits in word **i**, the corresponding 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

11. explain DMA controller

The DMA controller has three registers as follows.

- **Address register –**It contains the address to specify the desired location in memory.
- **Word count register –**It contains the number of words to be transferred.
- **Control register –**It specifies the transfer mode.

All registers in the DMA appear to the CPU as I/O interface registers. Therefore, the CPU can both read and write into the DMA registers under program control via the data bus.



The CPU initializes the DMA by sending the given information through the data bus.

- The starting address of the memory block where the data is available (to read) or where data are to be stored (to write).
- It also sends word count which is the number of words in the memory block to be read or write.
- Control to define the mode of transfer such as read or write.
- A control to begin the DMA transfer.

12. What is hit ration in cache memory?

A **cache hit ratio** is calculated by dividing the number of **cache hits** by the total number of **cache hits** and misses, and it measures how effective a **cache** is at fulfilling requests for content.

13.explain daisy chain priority interrupt

The daisy-chaining method involves connecting all the devices that can request an interrupt in a serial manner. This configuration is governed by the priority of the devices. The device with the highest priority is placed first followed by the second highest priority device and so on. The given figure depicts this arrangement.

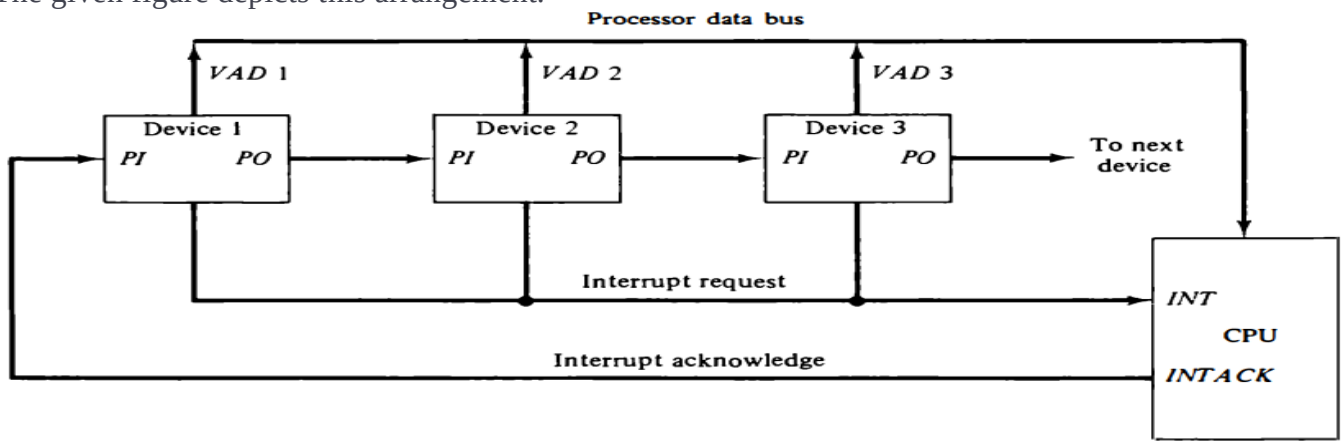


Figure 12 Daisy-chain priority interrupt.

WORKING:

There is an interrupt request line which is common to all the devices and goes into the CPU.

- When no interrupts are pending, the line is in HIGH state. But if any of the devices raises an interrupt, it places the interrupt request line in the LOW state.
- The CPU acknowledges this interrupt request from the line and then enables the interrupt acknowledge line in response to the request.
- This signal is received at the PI(Priority in) input of device 1.
- If the device has not requested the interrupt, it passes this signal to the next device through its PO(priority out) output. (PI = 1 & PO = 1)
- However, if the device had requested the interrupt, (PI =1 & PO = 0)
    - The device consumes the acknowledge signal and block its further use by placing 0 at its PO(priority out) output.
    - The device then proceeds to place its interrupt vector address(VAD) into the data bus of CPU.
    - The device puts its interrupt request signal in HIGH state to indicate its interrupt has been taken care of.

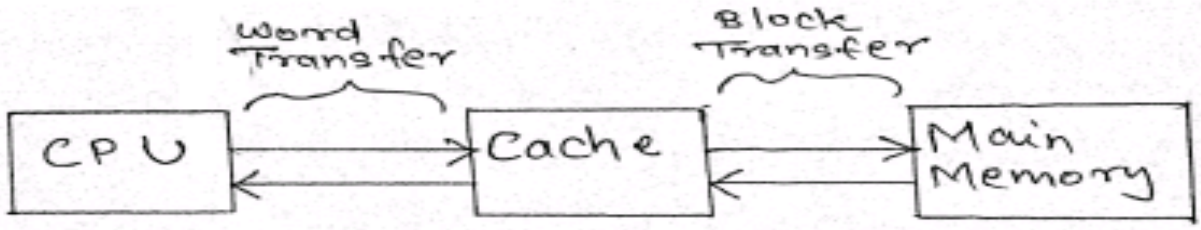
  NOTE: VAD is the address of the service routine which services that device.
- If a device gets 0 at its PI input, it generates 0 at the PO output to tell other devices that acknowledge signal has been blocked. (PI = 0 & PO = 0)

Hence, the device having PI = 1 and PO = 0 is the highest priority device that is requesting an interrupt. Therefore, by daisy chain arrangement we have ensured that the highest priority interrupt gets serviced first and have established a hierarchy. The farther a device is from the first device, the lower its priority.

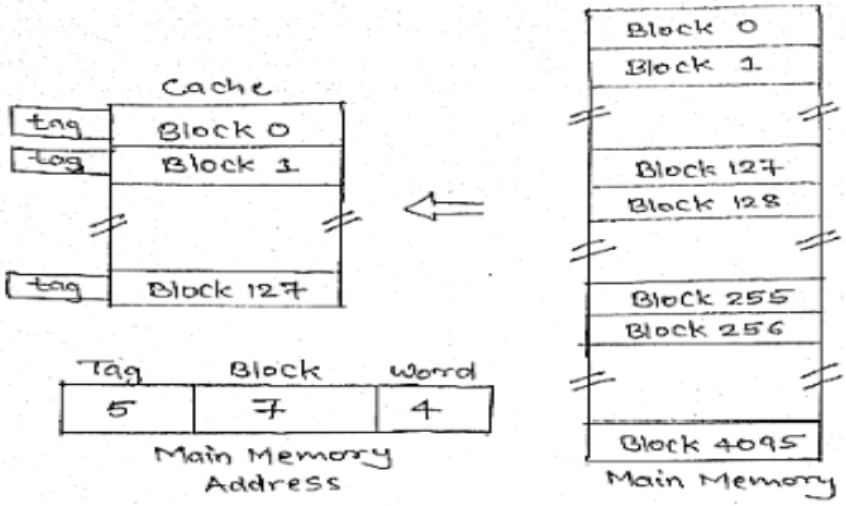
14.describe the various mapping techniques used with cache memory.

1) Cache Memory is very high speed memory used to increase the speed of program by making current program & data available to the CPU at a rapid rate.

2) Access time to cache memory is less compared to main memory. It contains a copy of potions of the main memory.
3) When CPU attempts to read a word from main memory, check is made to determine if the word is in cache. It so, then word is delivered form cache.
4) If word is not there in cache then a block of main memory consisting some word along with that word, is read into cache and the required word is delivered to CPU. This is called "Principle of Locality of Reference".
5) During a miss if there are no empty blocks in the cache, then some replacement policies such as FIFO, LRU, LFU, etc. are used.



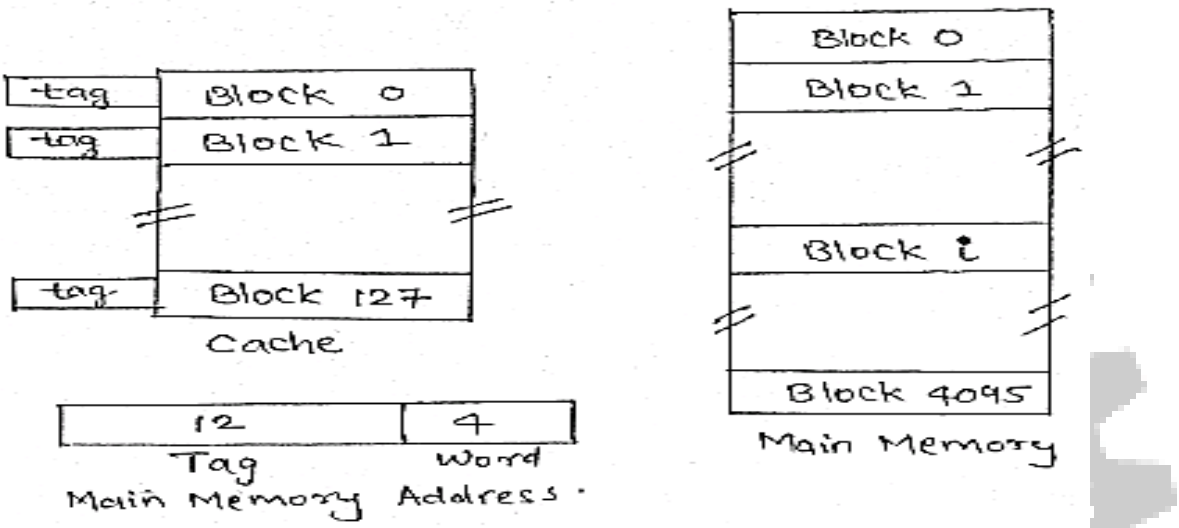
Cache Mapping Technique:-
The different Cache mapping technique are as follows:-
1) Direct Mapping
2) Associative Mapping
3) Set Associative Mapping
Consider a cache consisting of 128 blocks of 16 words each, for total of 2048(2K) works and assume that the main memory is addressable by 16 bit address. Main memory is 64K which will be viewed as 4K blocks of 16 works each.
(1) Direct Mapping:-



1) The simplest way to determine cache locations in which store Memory blocks is direct Mapping technique.
2) In this block J of the main memory maps on to block J modulo 128 of the cache. Thus main memory blocks 0,128,256,….is loaded into cache is stored at block 0. Block 1,129,257,….are stored at block 1 and so on.
3) Placement of a block in the cache is determined from memory address. Memory address is divided into 3 fields, the lower 4-bits selects one of the 16 words in a block.

4) When new block enters the cache, the 7-bit cache block field determines the cache positions in which this block must be stored.
5) The higher order 5-bits of the memory address of the block are stored in 5 tag bits associated with its location in cache. They identify which of the 32 blocks that are mapped into this cache position are currently resident in the cache.
6) It is easy to implement, but not Flexible

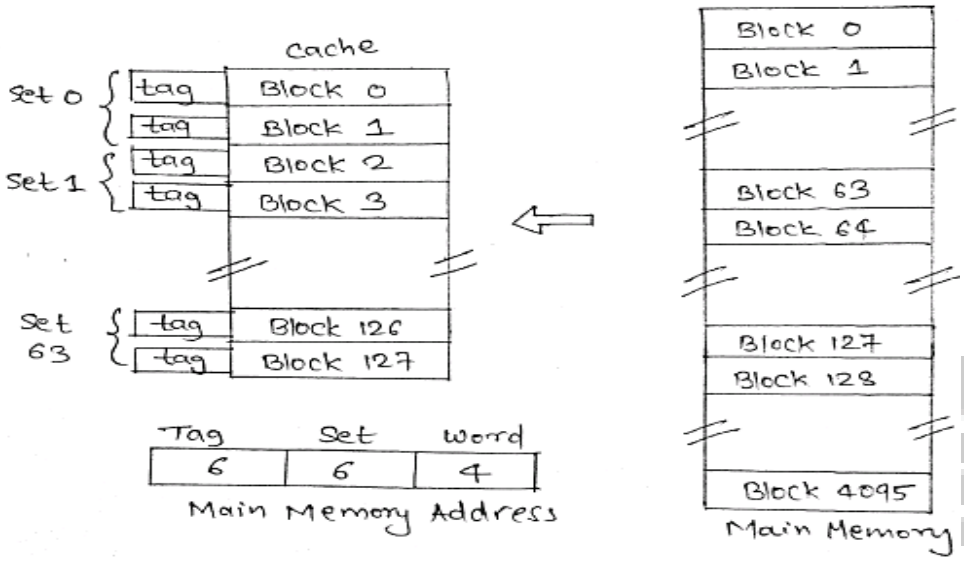
(2) Associative Mapping:-



1) This is more flexible mapping method, in which main memory block can be placed into any cache block position.
2) In this, 12 tag bits are required to identify a memory block when it is resident in the cache.
3) The tag bits of an address recevied from the processor are compared to the tag bits of each block of the cache to see, if the desired block is present. This is known as Associative Mapping technique.
4) Cost of an associated mapped cache is higher than the cost of direct-mapped because of the need to search all 128 tag patterns to determine whether a block is in cache. This is known as associative search.

(3) Set-Associated Mapping:-

1) It is the combination of direct and associative mapping technique.
2) Cache blocks are grouped into sets and mapping allow block of main memory reside into any block of a specific set. Hence contention problem of direct mapping is eased , at the same time , hardware cost is reduced by decreasing the size of associative search.
3) For a cache with two blocks per set. In this case, memory block 0, 64, 128,…..,4032 map into cache set 0 and they can occupy any two block within this set.
4) Having 64 sets means that the 6 bit set field of the address determines which set of the cache might contain the desired block. The tag bits of address must be associatively compared to the tags of the two blocks of the set to check if desired block is present. This is two way associative search.



15.Define virtual memory

Virtual memory is a feature of an operating system that enables a computer to be able to compensate shortages of physical **memory** by transferring pages of data from random access **memory** to disk storage. This process is done temporarily and is designed to work as a combination of **RAM** and space on the hard disk.

16.Expalin mapping techniques.

he process of transfer the data from main memory to cache memory is called as mapping. In the cache memory, there are three kinds of **mapping techniques** are used.

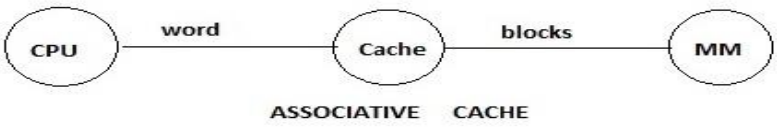
1. Associative mapping
2. Direct mapping
3. Set Associative mapping

Components present in each line are:

1. **Valid bit:** This gives the status of the data block. If 0 then the data block is not referenced and if 1 then the data block is referenced.
2. **Tag:** This is the main memory address part.
3. **Data:** This is the data block.

1) Associative mapping

In this technique, a number of mapping functions are used to transfer the data from main memory to cache memory. That means any main memory can be mapped into any cache line. Therefore, cache memory address is not in the use. Associative cache controller interprets the request by using the main memory address format. During the mapping process, the complete data block is transferred to cache memory along with the complete tags.



- Associative cache controller interprets the CPU generated request as:

- The existing tag in the cache controller compared with the CPU generated tags.
- If anyone of the tag in the matching operation becomes hit. So, based on the word offset the respective data is transfer to CPU.
- If none of the tags are matching operation become miss. So, the references will be forwarded to the main memory.
- According to the main memory address format, the respective main memory block is enabled then transferred to the cache memory by using the associative mapping. Later the data will be transfer to the CPU.
- In this mapping technique, replacement algorithms are used to replace the cache block when the cache is full.
- Tag memory size = number of lines * number of tag bits in the line.

Tag memory size = 4*3 bits

Tag memory size = 12 bits

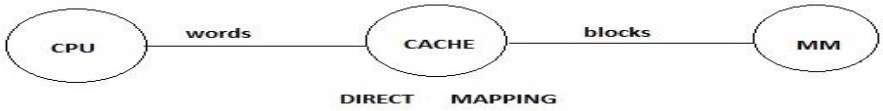
2) Direct mapping

In this mapping technique, the mapping function is used to transfer the data from main memory to cache memory. The mapping function is:

K mod N = i

Where,

- K is the main memory block number.
- N is the number of cache lines.
- And, i is the cache memory line number.



- Direct cache controller interprets the CPU generated a request as:

- Line offset is directly connected to the address logic of the cache memory. Therefore the corresponding cache line is enabled.
- Existing tag in the enabled cache line is compared with the CPU generated the tag.
- If both are matching operation then it becomes hit. So, the respective data is transfer to CPU based on the word offset.
- If both are not matching operation then it becomes a miss. So, the reference is forward into the main memory.

- According to the main memory address format, the corresponding block is enabled, then transferred to the cache memory by using the direct mapping function. Later the data is transfer to the CPU based on the word offset.
- In this mapping technique, a replacement algorithm is not required because the mapping function itself replaces the blocks.
- The disadvantage of direct mapping is each cache line is able to hold only one block at a time. Therefore. The number of conflicts misses will be increased.
- To avoid the disadvantage of the direct mapping, use the alternative cache organization in which each line is able to hold more than one tags at a time. This alternative organization is called as a set associative cache organization.
- Tag memory size = number of lines * number of tag bits in the line.

Tag memory size = 4*1 bits
Tag memory size =4 bits

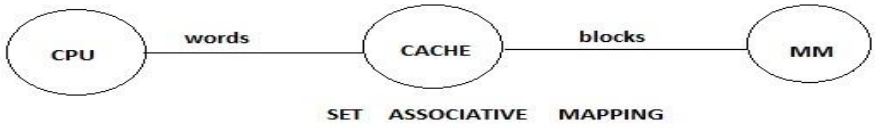
Set Associative Mapping

In this mapping technique, the mapping function is used to transfer the data from main memory to cache memory. The mapping function is:

K mod S = i

Where,

- K is the main memory block number,
- S is the number of cache sets,
- And, i is the cache memory set number.



- Set associative cache controller interprets the CPU generated a request as:

- A set offset is directly connected to the address logic of the cache memory. So, respective sets will be enabled.
- Set contain multiple blocks so to identify the hit block there is a need of multiplexer to compare existing tags in the enabled set one after the another based on the selection bit with the CPU generated a tag.
- If anyone is matching, the operation becomes hit. So, the data is transfer to the CPU. If none of them is matching, then operation becomes a miss. So, the reference is forward to the CPU.
- The main memory block is transferred to the cache memory by using a set associative mapping function. Later data is transfer to CPU.
- In this technique, replacement algorithms are used to replace the blocks in the cache line, when the set is full.
- Tag memory size = number of sets in cache * number of blocks in the set * tag bit.

Tag memory size = 2 * 2 * 2 bits.
Tag memory size = 8 bits.

17.DefineBurst Mode of DMA

It is the DMA data transfer technique in which no. of data words are transferred continuously until whole data is not transferred.

18.DefineCycle Stealing Mode

It is the data transfer technique in which one data word is transferred and then control is returned to CPU.