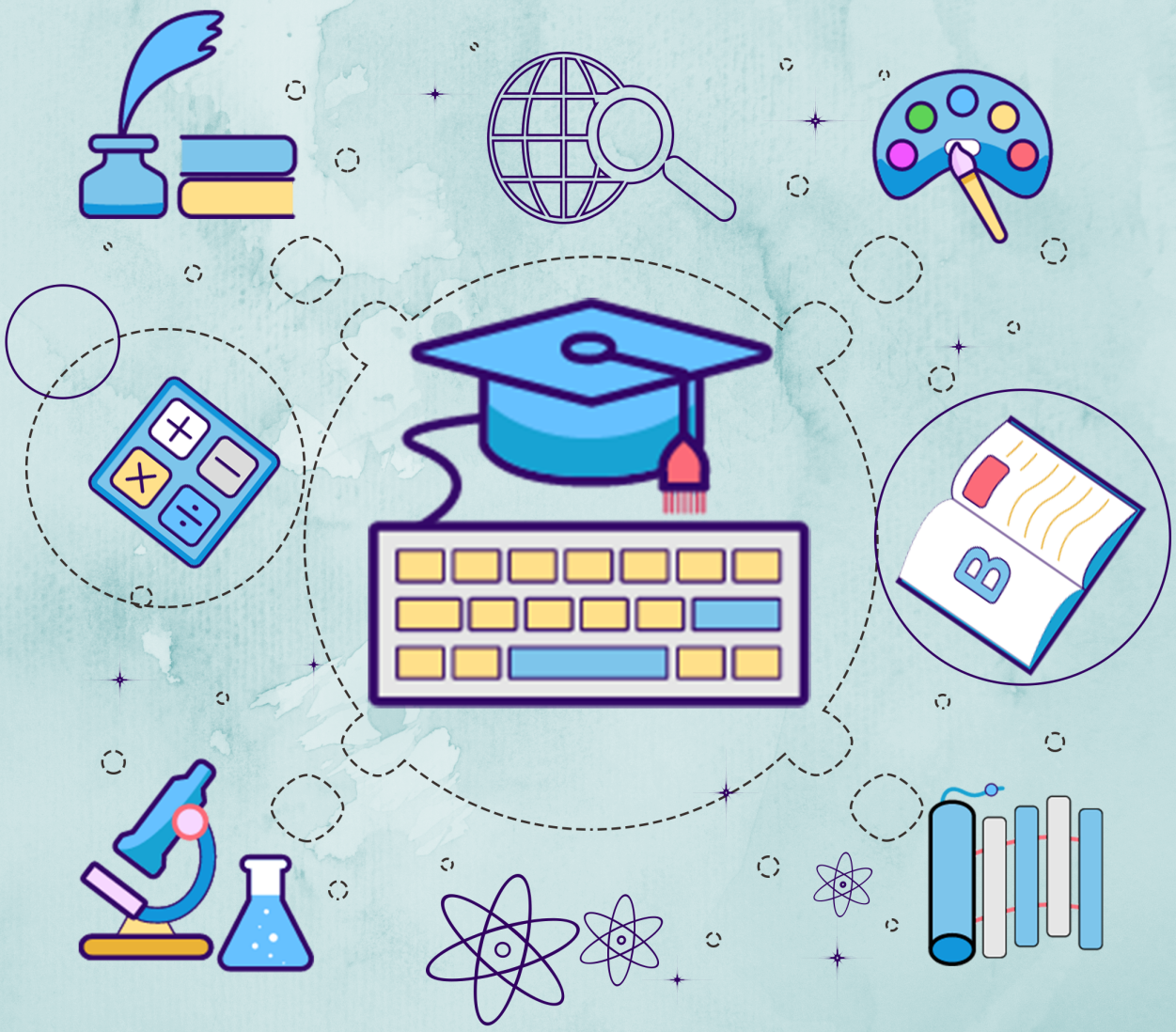
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

KeralaNotes



SYLLABUS | STUDY MATERIALS | TEXTBOOK PDF | SOLVED QUESTION PAPERS

www.keralanotes.com

KeralaNotes

KTU S4 CSE

COMPUTER ORGANIZATION AND ARCHITECTURE (CST 202)

QUESTION BANK

Related Link :

- KTU S4 CSE NOTES 2019 SCHEME
- KTU S4 SYLLABUS CSE COMPUTER SCIENCE
- KTU PREVIOUS QUESTION BANK S4 CSE SOLVED
- KTU CSE TEXTBOOKS S4 B.TECH PDF DOWNLOAD
- KTU S4 CSE NOTES | SYLLABUS | QBANK | TEXTBOOKS DOWNLOAD

www.keralanotes.com

MODULE 5

71. What are the different techniques for interfacing I/O device with CPU?

Memory mapped I/O and I/O mapped I/O

72. What is memory mapped I/O?

This is one of the techniques for interfacing I/O devices with CPU. In memory mapped I/O, the I/O devices assigned and identified by 16-bit addresses. To transfer the data between CPU and I/O devices memory related instructions (such as LDA, STA etc.) and memory control signals (MEMR, MEMW) are used. .

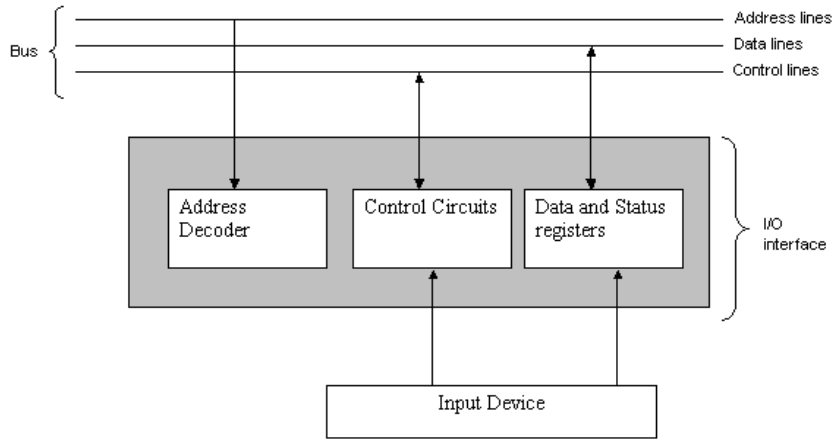
The CPU selects an I/O device by placing the address on the address lines; the corresponding I/O device recognizes its address and responds to the CPU's command. Any machine instruction used to access memory can be used to transfer data to and from I/O devices.

Example:

MOVE DATAIN, R_0

MOVE R_0, DATAOUT, Where DATAIN and DATAOUT are I/O buffer registers.

For More Study Materials :https://www.keralanotes.com/



73. What is I/O mapped I/O?

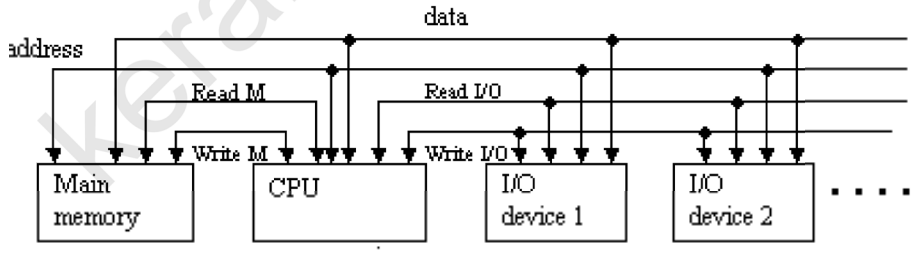
The memory and I/O address space are kept separate by Read/Write control lines.A memory reference instruction activates the Read M or Write M line and does not affect the I/O devices.

The I/O instruction such as for example IN or OUT activates the Read I/O and write I/O line and causes a word to be transferred between the addressed I/O and the CPU.

Therefore, the memory location and the I/O device may have the same address.

OUT data, device

IN data, device



74. What are the different data transfer modes between the CPU and the I/O devices.

- Programmed I/O
- Interrupt driven I/O
- Direct Memory Access (DMA)

For More Study Materials :https://www.keralanotes.com/

75. Explain Programmed I/O.

It is a mode of data transfer between the CPU and the logic within the peripherals. The transfer is initiated using an I/O transfer (IOT) instruction and controlled entirely by the CPU.

Programmed I/O can be:

- Synchronous (Or Unconditional) transfer: Here no check is made by the CPU to determine the state of the peripherals like Ready or Busy. The transfer takes place when ever the CPU directs it i.e. the I/O device must be ready to receive the output data from the CPU or to input data into the CPU when ever CPU initiates such a transfer. Hence the transfer is Unconditional. The data transfers are controlled by a clock of fixed frequency, independent of the CPU
- Asynchronous (Or Conditional) transfer: The peripheral is queried about its status to see if it is ready to perform the transfer. The actual transfer takes place only when the peripheral shows Ready state. This method accomplishes synchronization between the CPU and slower peripherals whose timing may not be fixed.

76. List the advantages and disadvantages of Programmed I/O.

Advantages

- The programmed I/O mode is very low cost interface
- Can meet speeds of most of the peripherals.

Disadvantages: CPU speed is brought down to that of the slow peripheral device

77. Explain Interrupt driven I/O.

Here, the CPU does not wait for the peripheral device to become ready for the data transfer instead, it initiates the device and then goes away to handle other tasks. The peripheral sends the Ready signal to the CPU whenever it is ready. An interrupt request from an I/O device may cause the processor to suspend the execution of one program and start the execution of another program. Interrupts may sometimes alter the sequence of events. The programmer should be able to enable and disable interrupts. This is performed by machine instructions such as Interrupt Enable and Interrupt Disable.

For More Study Materials :https://www.keralanotes.com/

78. **Explain the general sequence of events in interrupt I/O**

Interrupt request - The I/O requesting peripheral sends an interrupt request signal (INT REQ) to the CPU to indicate that it is ready for an I/O operation

Interrupt acknowledge - On receiving the interrupt request signal, the CPU completes the execution of the current instruction and sends an Interrupt acknowledgement signal (INT ACK) to the interrupt requesting peripheral

Status saving - Before servicing the interrupt request, the return address of the main program is stored in some pre-assigned memory locations, so that the CPU can resume its normal operation on completion of the subroutine.

Device identification - The interrupt request to the CPU is only through a single line bus, if the system has more than one device connected to it, the CPU must identify the interrupting peripheral, so that it can call the appropriate interrupt service routine (ISR) from the memory.

Peripheral service – having identified the peripheral which requested for the service, the program branches to the starting address of the specific subroutine in the memory and it services the peripheral.

Restore CPU status – after completion of the service subroutine, the data from the CPU registers that were held temporarily in memory are reloaded back to their respective locations and the CPU can take up its normal task

Resume main task – when all registers and program counter are restored with their original contents, the main program operation is resumed.

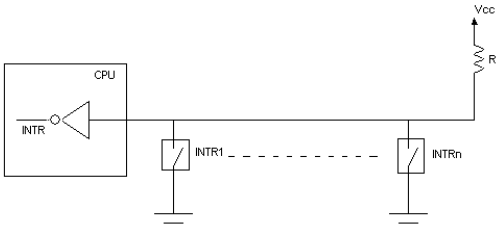
79. **Explain the methods for identifying the interrupting device.**

Polling and vectored interrupts are the methods to identifying the interrupting device.

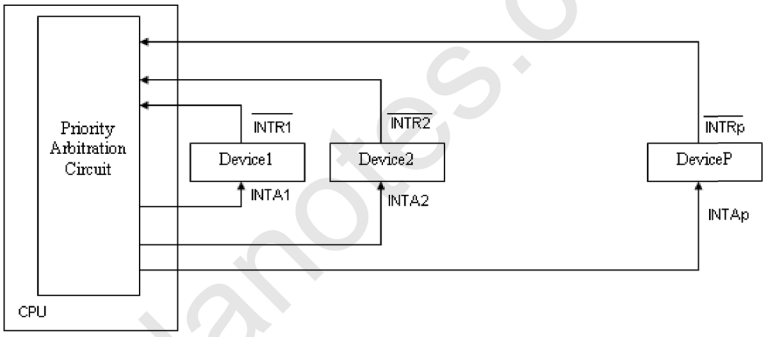
Polling

All the devices are connected to a common INTR (Interrupt Request) line via switches to ground. When all the interrupt requests signals (INTR1 to INTRn) are inactive, the INTR line is maintained in the high voltage state. When any device requests an interrupt it closes its switch and the voltage on the line drops to zero, causing INTR line to go to 1.

For More Study Materials :https://www.keralanotes.com/



A device that requests an interrupt identifies itself to the processor by sending a special code called vector to the processor. This code supplies the starting address of the ISR for that device. The processor reads this address, called the interrupt vector & loads into PC. This simplifies the circuit but it limits the number of devices that can be identified. One solution is to assign a code to a group of devices and the ISR identify the device by polling the devices in that group.



80. Explain interrupt nesting.

An interrupt request from a high priority device should be accepted while the processor is servicing another request from a low priority device. For this the priority level of the processor can be set to that of the device that is currently being serviced. Interrupt requests from the devices with priority greater than that of the processor will only be accepted.

Processor's priority is encoded in a few bits of the Program Status Word. Hardware is implemented by using a separate INTR and INTA line for each device. An interrupt request is sent to the priority arbitration logic and is accepted only if it is greater than the one that is currently assigned to the processor.

81. What is DMA?

Direct memory access refers to that data transfer, when large blocks of data are to be transferred at high speeds between I/O devices and main memory, without involving the

For More Study Materials :https://www.keralanotes.com/

CPU. In order to perform DMA transfer, an additional interface circuitry is required known as **DMA controller.**

82. Describe DMA Controller registers.

The DMA controller has the following four registers to perform transfer:

a. **Address register** – used to hold the address of the next word to be transferred
b. **Data buffer register** – used to store the data to be transferred
c. **Data count register or word counter** – used to store the number of words that are remaining to be transferred. It is automatically decremented after the transfer of each word and tested for zero. When the data count reaches zero, the DMA transfer halts.
d. **Control register and circuits** – used to control the DMA transfer

83. What is the need for DMA transfer?

During the actual transfer between peripheral and memory, CPU is free from this activity and can do some other useful work.

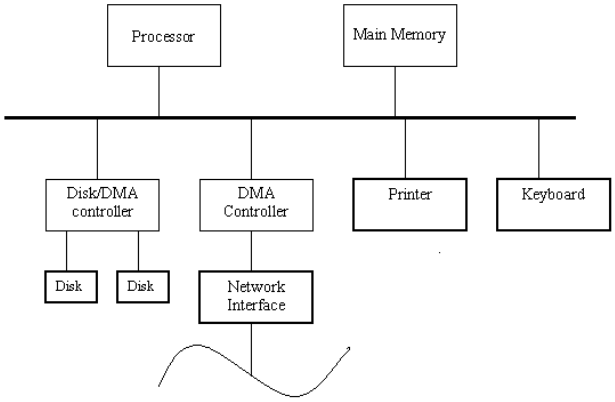
84. Explain the action carried out by the processor after occurrence of an interrupt.

An interrupt is a signal that refers to the transfer of program control from a currently running program to another service program as a result of an external or internal generated request. The control returns to the original program after the service program is executed. Interrupts are used to coordinate slow external devices with fast CPU. One of the bus control lines called the interrupt request line is dedicated for this purpose. Here, the CPU does not wait for the peripheral device to become ready for the data transfer instead, it initiates the device and then goes away to handle other tasks. The peripheral sends the Ready signal to the CPU whenever it is ready.

85. Explain with the block diagram the DMA transfer in a computer system

The peripheral devices initiate the DMA transfer by sending a DMA request signal to the CPU when it is ready to perform a transfer.

For More Study Materials :https://www.keralanotes.com/



There are two methods by which DMA transfer is initiated and operated:

Burst Mode

When the data is being transferred to the memory or from the memory, the CPU remains idle for relatively long periods of time or the CPU is in a kind of HOLD state. The CPU exits form this state only after the DMA request signal is withdrawn from the I/O device.The hold duration depends on the speed of the memory, I/O device, and number of bytes to be transferred.

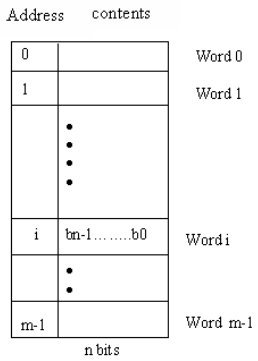
Cycle-Stealing Mode

The CPU after receiving the DMA request signal from the peripheral, will attend to the current instruction, it is doing and suspends its operation for the next time slice (machine cycle). During the next time slice, the DMA transfer takes place and one word is transferred between the peripheral device and the main memory. On completion of the transfer, the CPU can resume to its normal operation. DMA transfers are essentially time-sliced with the operations of the main memory. Here, the DMA controller can be regarded as "stealing" memory cycles from the CPU. Hence, this technique is known as cycle stealing.

86. Explain Memory system

The main memory consists of a large number of storage cells, each of which can store a binary digit 0 or 1. This 1-bit representation of information is too small to be handled by a computer. So, a group of n bits are used while storing or retrieving. Each group of n bits is referred to as its word length. The word length can be 8 bits, 16 bits, 32, 64 or 128 bits depending on the size of the computer.

For More Study Materials :https://www.keralanotes.com/



Each location in the main memory is given a distinct name or address. With this address it is possible to store or retrieve information from the memory location. The contents of memory locations can represent either instructions or operands.

87. Explain Big Endian and Little Endian Assignments.

Two ways that byte address can be assigned across words. Big Endian and Little Endian.

Big-endian is an order in which the "big end" (most significant value in the sequence) is stored first (at the lowest storage address). In big endian, store the most significant byte in the smallest address.

Word address	Byte address			
0	0	1	2	3
4	4	5	6	7
8	8	9	10	11

Little-endian is an order in which the "little end" (least significant value in the sequence) is stored first. In little endian, you store the least significant byte in the smallest address.

Word address	Byte address			
0	3	2	1	0
4	7	6	5	4
8				

88. Define Memory access time, memory cycle time and transfer rate.

- **Memory Access time:** It is the time that elapses between the initiation of an operation and the completion of that operation. E.g.: time between Read and MFC
- **Memory Cycle time :** It is the minimum time delay required between the initiations of two independent memory operations. E.g.: two successive Read operations.

For More Study Materials :https://www.keralanotes.com/

- **Transfer Rate:** It is the rate at which the data can be transferred to or out of memory unit.

TN=TA+N/R

TN= Average time to read / write n bits

TA=- Average access time

N= Number of bits

R= Transfer rate in bits/ sec.

89. Discuss the Classification of memory

Memory can be either primary memory or Secondary memory.

Two classifications of primary storage are read-only memory (ROM) and random-access memory (RAM).

a) READ-ONLY MEMORY (ROM)

In computers, it is useful to have instructions that are used often, permanently stored inside the computer. ROM enables us to do this without loosing the programs and data when the computer is powered down. Only the computer manufacturer can provide these programs in ROM; once done, we cannot change it. Consequently, we cannot put any of our own data or programs in ROM. Many complex functions, such as translators for high-level languages, and operating systems are placed in ROM memory. Since these instructions are hardwired, they can be performed quickly and accurately.

- **Programmable ROM (PROM):** This is a type of ROM that can be programmed using special equipment; it can be written to, but only once. This is useful for companies that make their own ROMs from software they write, because when they change their code they can create new PROMs without requiring expensive equipment. This is similar to the way a CD-ROM recorder works by letting you "burn" programs onto blanks once and then letting you read from them many times. In fact, programming a PROM is also called *burning*, just like burning a CD-R, and it is comparable in terms of its flexibility.
- **Erasable Programmable ROM (EPROM):** An *EPROM* is a ROM that can be erased and reprogrammed. A little glass window is installed in the top of the ROM package, through

For More Study Materials :https://www.keralanotes.com/

which you can actually see the chip that holds the memory. Ultraviolet light of a specific frequency can be shined through this window for a specified period of time, which will erase the EPROM and allow it to be reprogrammed again.

- **Electrically Erasable Programmable ROM (EEPROM):** The next level of erasability is the *EEPROM*, which can be erased under software control. This is the most flexible type of ROM, and is now commonly used for holding BIOS programs.

b) RANDOM-ACCESS MEMORY (RAM)

RAM is another type of memory found inside computers. It may be compared to a chalkboard on which we can scribble down notes, read them, and erase them when finished. In the computer, RAM is the working memory. Data can be read (retrieved) or written (stored) in RAM by providing the computer with an address location where the data is stored or where you want it to be stored. When the data is no longer required, we may simply write over it. Thus you can use the storage location again for something else.

The RAM family includes two important memory devices **static RAM** (SRAM) and **dynamic RAM** (DRAM). The primary difference between them is the lifetime of the data they store. *SRAM* retains its contents as long as electrical power is applied to the chip. If the power is turned off or lost temporarily, its contents will be lost forever. *DRAM*, on the other hand, has an extremely short data lifetime-typically about four milliseconds. This is true even when power is applied constantly.

A simple piece of hardware called a *DRAM controller* can be used to make DRAM behave more like SRAM. The job of the DRAM controller is to periodically refresh the data stored in the DRAM. By refreshing the data before it expires, the contents of memory can be kept alive for as long as they are needed. So DRAM is as useful as SRAM after all.

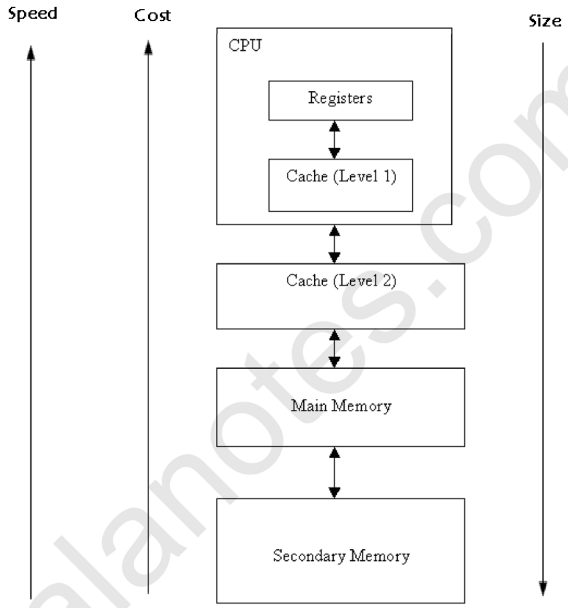
SRAM devices offer extremely fast access times (approximately four times faster than DRAM) but are much more expensive to produce. Generally, SRAM is used only where access speed is extremely important. A lower cost-per-byte makes DRAM attractive whenever large amounts of RAM are required.

Secondary storage, or auxiliary storage, is memory external to the main body of the computer (CPU) where programs and data can be stored for future use. When the computer is ready to

For More Study Materials :https://www.keralanotes.com/

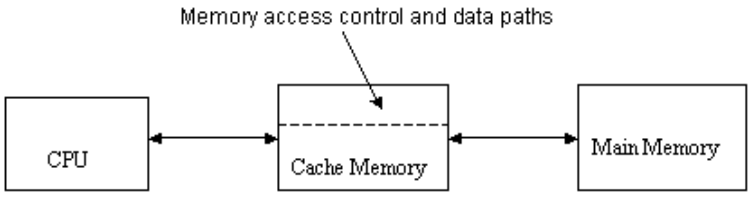
use these programs, the data is read into primary storage. Secondary storage media extends the storage capabilities of the computer system. Secondary storage is required for two reasons. First, the working memory of the CPU is limited in size and cannot always hold the amount of data required. Second, data and programs in secondary programs do not disappear when the power is turned off. Secondary storage is nonvolatile memory. This information is lost only when we erase it. Magnetic disks are the most common type of secondary storage. They may be either floppy disks or hard disks (hard drives).

90. Draw the picture of memory hierarchy.



91. What is the need of a cache memory?

Speed of main memory is very low compared to processor speed. The speed gap between processor and memory is resolved by using cache memories. The cache memory is a small and fast memory that is placed between the main memory and the CPU. It holds the currently active and more frequently used segments of program and its data.



For More Study Materials :https://www.keralanotes.com/

In this approach, the CPU need not always fetch the program and data from the main memory instead it can get it from the cache memory. Hence, the total time of execution is considerably reduced and it is more economical.

92. Explain the basic principle of cache.

The cache working is based on the phenomenon **Locality of Reference.**

Principle of Locality of reference: Program access a relatively small portion of the address space at any instant of time.

Example: 90% of time in 10% of the code

The locality of reference occur in two ways

- **Spatial Locality**

Instructions are closed to recently executed instruction (based on their address) are likely to be executed soon.

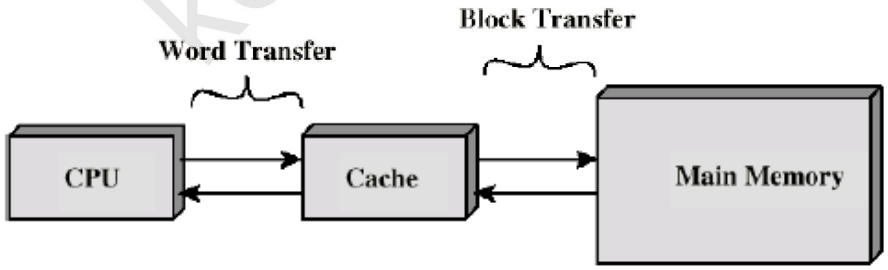
- **Temporal Locality**

Recently executed instructions are likely to be executed again very soon.

If the active segments are placed in the cache then execution time can be reduced then the total execution time is reduced. Due to temporal aspect, wherever information (data) are first needed this item should be brought into cache. Due to spatial aspect, instead of fetching an item fetch the assignment also.

93. Explain the basic operation of cache.

The cache memory is divided into number of blocks or cache lines.



When the CPU requests for an addressed word, it is transferred from the main memory to the cache. The memory-access control circuitry in the cache, checks whether the requested word currently exists in the cache. If the requested word exists in cache, then the **Read or Write hit** occurs and the required operation is performed on the appropriate location in the

For More Study Materials :https://www.keralanotes.com/

cache. When the operation is Read, the main memory is not involved. However, if the operation is Write, then there are two techniques followed. In the first technique known as **write through**, the main memory and cache locations are updated simultaneously. In the second technique known as **write back**, only the cache location is updated and it is marked as updated using **dirty** or **modified bit.** The main memory will be updated during swapping. If the requested word does not exist in cache, then **a read or write miss** (cache miss) occurs. If the operation is Read**,** then there are two techniques followed. In the first technique the block of words that contain the requested word is loaded from the main memory into the cache and the requested word is forwarded to the CPU. In the second technique the requested word is directly sent to the CPU as soon as it is read from main memory. This technique is known as **load-through or early restart.** This reduces the processor waiting time. When the operation is Write, and the requested word is not in cache, then it is written directly into the main memory.

94. Define Hit Ratio of cache memory.

$$\text{Hit Ratio} = \frac{\text{Number of hits}}{\text{Total CPU references}}$$

$$\text{Hit Ratio} = \frac{\text{Number of hits}}{\text{Number of Hits} + \text{Number of Misses}}$$

95. What are the mapping functions commonly used in cache memory?

Direct mapping technique

Associative mapping technique

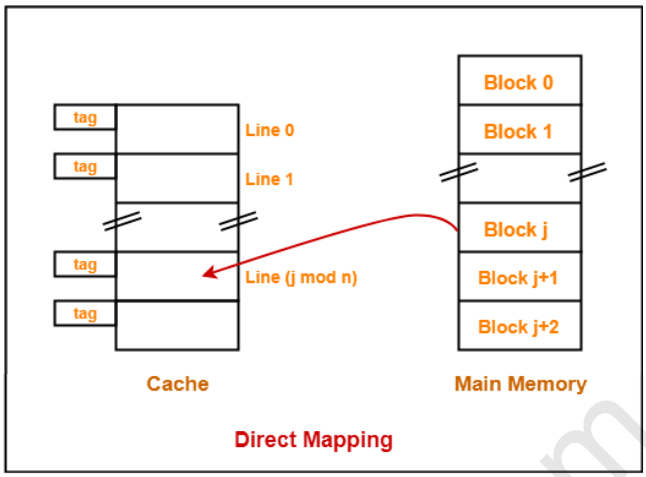
Set associative mapping technique

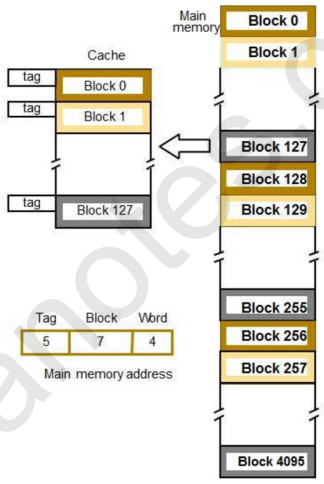
96. Explain direct mapping technique.

In direct mapped cache each main memory block is assigned to a specific line in the cache according to the formula i = j modulo N, where i is the cache line number assigned to main memory block j, and N is the number of lines in the cache. Direct mapping cache interprets a main memory address (comprising of s + w bits) as 3 distinct fields. Tag (the most significant s - r bits) is a unique identifier for each memory block. Line number (middle r bits) specifies which line will hold the referenced address. Line number identify one of the $N = 2^r$ lines of cache.

For More Study Materials :https://www.keralanotes.com/

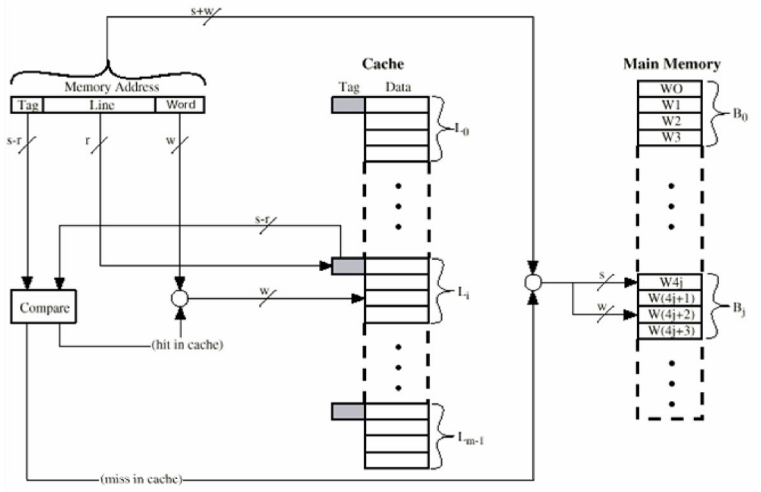
Word (least significant w bits) specifies the offset of a byte within a line or block. Word specifies the specific byte in a cache line that is to be accessed.





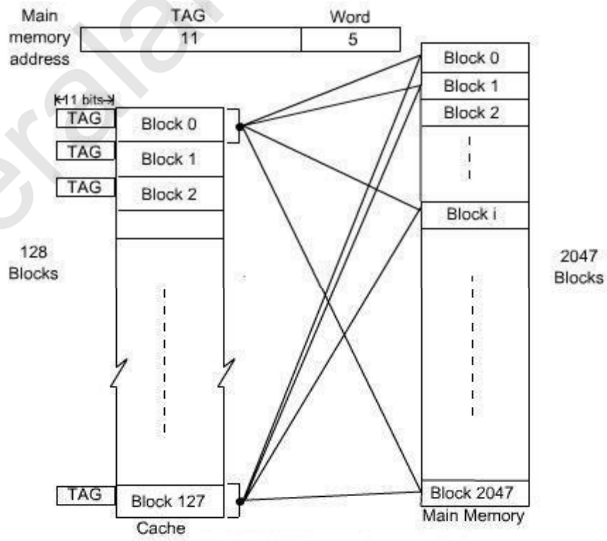
Block j of the main memory maps to j modulo 128 of the cache. (ie) Block 0, 128, 256 of main memory is maps to block 0 of cache memory. 1,129,257 maps to 1, & so on.

For More Study Materials :https://www.keralanotes.com/



97. Explain Associative mapping.

In associative mapping a block of the main memory can map to any available cache line. Associative mapping cache interprets a main memory address (comprising of s + w bits) as two distinct fields. Tag (the most significant s bits) is a unique identifier for each memory block. Word (least significant w bits) specifies the offset of a byte within a line or block. It might not be practical to use this complete flexibility of associative mapping technique due to searching overhead, because the TAG field of main memory address has to be compared with the TAG field of the entire cache block.

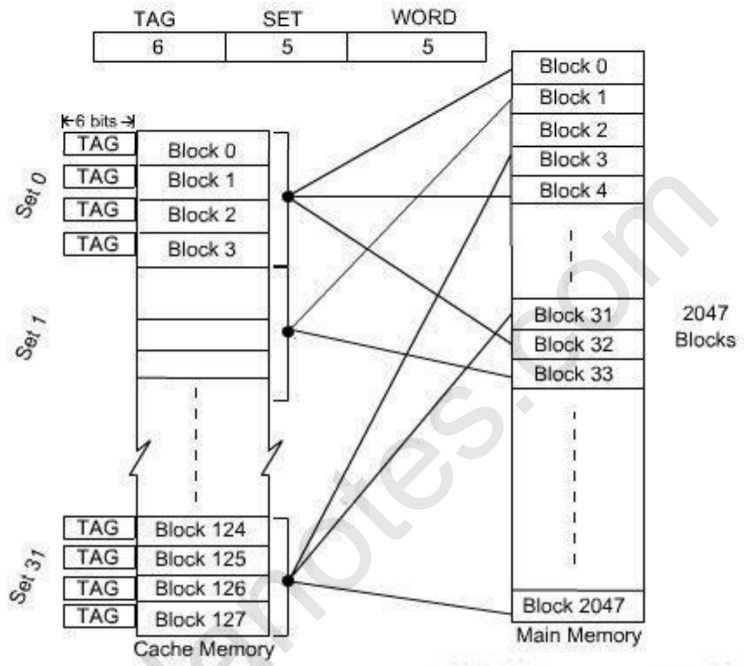


98. Explain Set associative mapping.

This mapping is compromise between direct mapping and fully associative mapping. It builds on the strengths of both: namely, the easy control of the direct mapped cache and

For More Study Materials :https://www.keralanotes.com/

the more flexible mapping of the fully associative cache. In set associative mapping, cache is divided into a number of smaller direct-mapped areas called sets (*v*), with each set holding a number of lines (*k*).The cache is then described in the number of lines each set contains. If a set can hold *X* lines, the cache is referred to as an X-way set associative cache. A main memory block can be stored in any one of the *k* lines in a set such that *cache set number = j modulo v (*where *j* is the main memory block number).



The TAG field of associative mapping technique is divided into two groups, one is termed as SET bit and the second one is termed as TAG bit. Each set contains 4 blocks, total number of set is 32. The main memory address is grouped into three parts: low-order 5 bits are used to identifies a word within a block. Since there are total 32 sets present, next 5 bits are used to identify the set. High-order 6 bits are used as TAG bits.

99. What is meant by block replacement?

When the cache is full, there is a need for replacement algorithm for replacing the cache block with a new block.

100. Discuss various block replacement algorithms in cache.

Three types of replacement algorithms

• **Random replacement policy.**

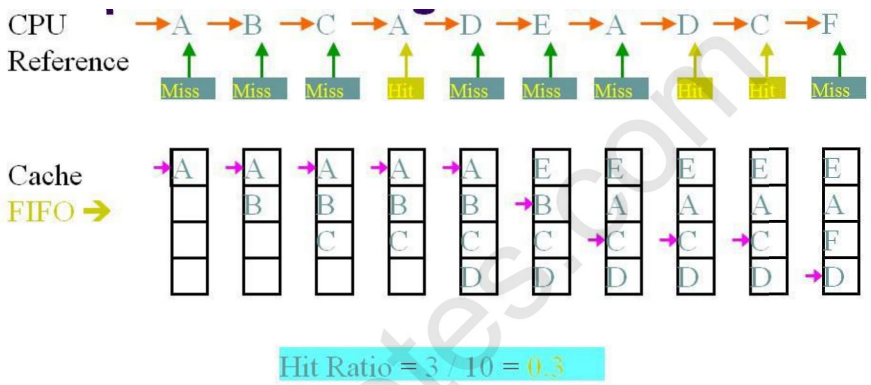
For More Study Materials :https://www.keralanotes.com/

Choose the block to be overwritten at random. In this algorithm replace any cache line by using random selection. Simple and very effective in practice.

• **First in first Out (FIFO) replacement policy**

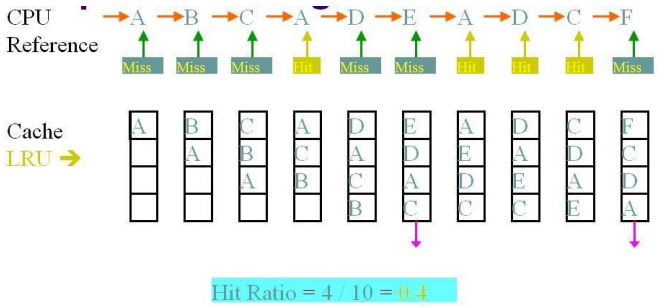
In this algorithm replace the cache block which is having the longest time stamp. While using this technique there is no need of updating when a hit occurs but when there is a miss occur then the block is put into an empty block and the counter values are incremented by one.

Example:



• **Least recently used (LRU) replacement policy.**

In the LRU, replace the cache block which is having the less reference with the longest time stamp. In this case also when a hit occurs when the counter value will be set to 0 but when the miss occurs there will be arising of two possibilities in which one possibility is that counter value is set as 0 and in another possibility, the counter value can be incremented as 1.

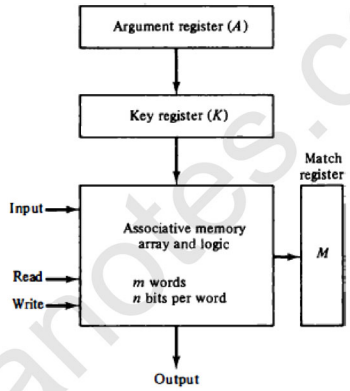


For More Study Materials :https://www.keralanotes.com/

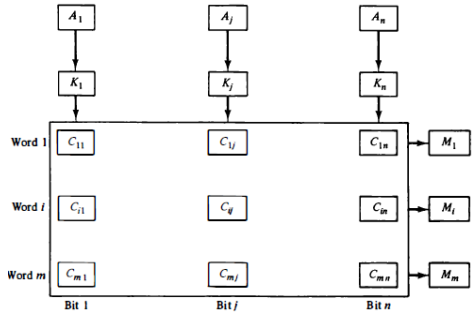
101. Discuss the concept of associative memory.

A memory unit accessed by content is called **an associative memory or Content Addressable Memory (CAM).**

This type of memory is accessed simultaneously and in parallel on the basis of data content rather than by specific address or location. When a word is written in an associative memory, no address is given. The memory is capable of finding an empty unused location to store the word. When a word is to be read from an associative memory, the content of the word, or part of the word, is specified. The memory locates all words, which match the specified content and marks them for reading. The organization of associative memory is as shown in figure 8. The comperand register is also known as argument register. The masking register is known as key register.



The match register M has m bits, one for each memory word. Each word in memory is compared in parallel with the content of the argument register. The words that match the bits of the argument register set a corresponding bit in the match register. After the matching process, those bits in the match register that have been set indicate the fact that their corresponding words have been matched.



For More Study Materials :https://www.keralanotes.com/

Example

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

keralanotes.com

For More Study Materials :https://www.keralanotes.com/