**Practice Set -2 for 5th Semester COA ETE Examination**

**Q.1 Discuss the various pipeline conflicts or pipeline hazards.**

A **pipeline hazard** is a potential discrepancy in the execution flow of a pipeline that manifests when several instructions overlap during execution inside the pipeline, leading to conditions that can interrupt the operation of the pipeline.

A hazard, in essence, prevents an instruction present in the pipe from being performed during the specified clock cycle. Since each of the instructions may be in a separate machine cycle, we use the term clock cycle.

**Types of Pipeline Hazards**

The three different types of hazards in computer architecture are:

1. *Structural:* Hardware resource conflicts among the instructions in the pipeline cause structural hazards. Memory, a GPR Register, or an ALU might all be used as resources here. When more than one instruction in the pipe requires access to the very same resource in the same clock cycle, a resource conflict is said to arise. In an overlapping pipelined execution, this is a circumstance where the hardware cannot handle all potential combinations.

2. *Data:* Data hazards in pipelining emerge when the execution of one instruction is dependent on the results of another instruction that is still being processed in the pipeline. The order of the READ or WRITE operations on the register is used to classify data threats into three groups.

3. *Control:* Branch hazards are caused by branch instructions and are known as control hazards in computer architecture. The flow of program/instruction execution is controlled by branch instructions. Remember that conditional statements are used in higher-level languages for iterative loops and condition testing (correlate with while, for, and if case statements). These are converted into one of the BRANCH instruction variations. As a result, when the decision to execute one instruction is reliant on the result of another instruction, such as a conditional branch, which examines the condition’s consequent value, a conditional hazard develops.

**Q.2 How many 128 X 8 RAM chips are needed to provide a memory capacity of 2048 bytes?**

According to question, 128 x 8 RAM = 1024 bits and 1 bit = 1/8Byte

Thus, 1024 bits = 1024 /8 = 128 Bytes

1 RAM chip = 128 Byte.

Let’s consider there are n number of Chips to provide 2048 Bytes Now, 1RAM Chip = 128 Byte

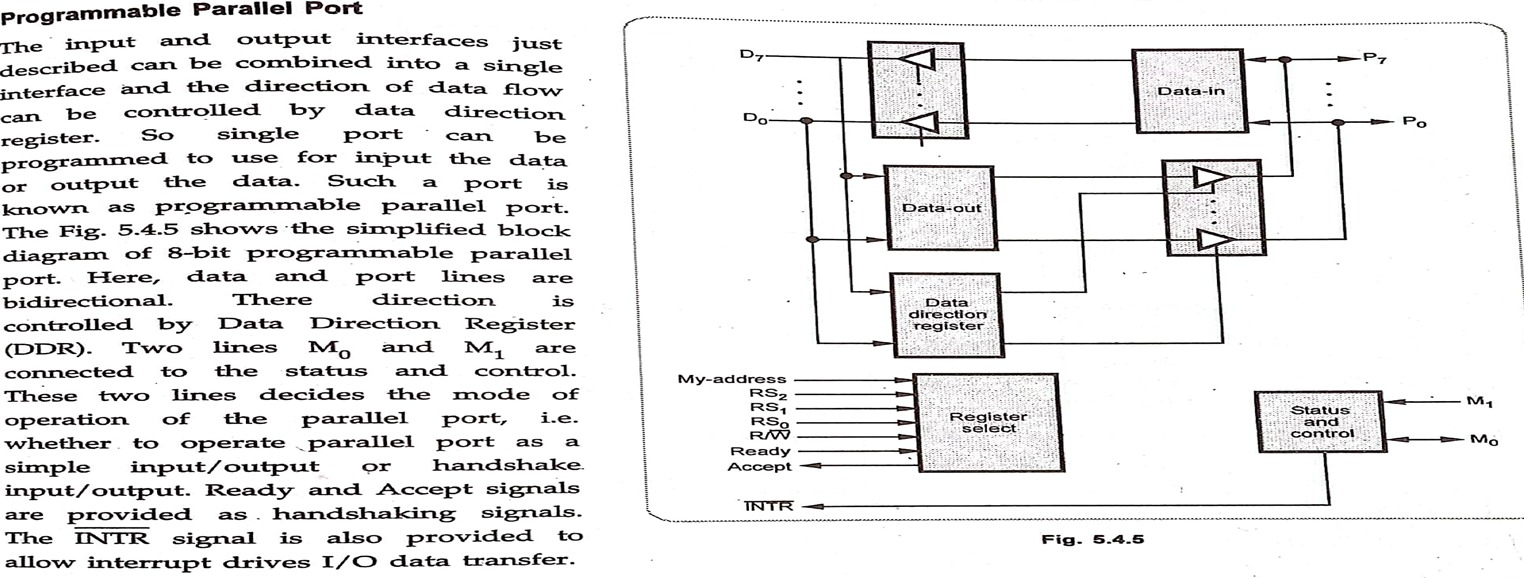
128n = 2048 => n = 2048/128= 16

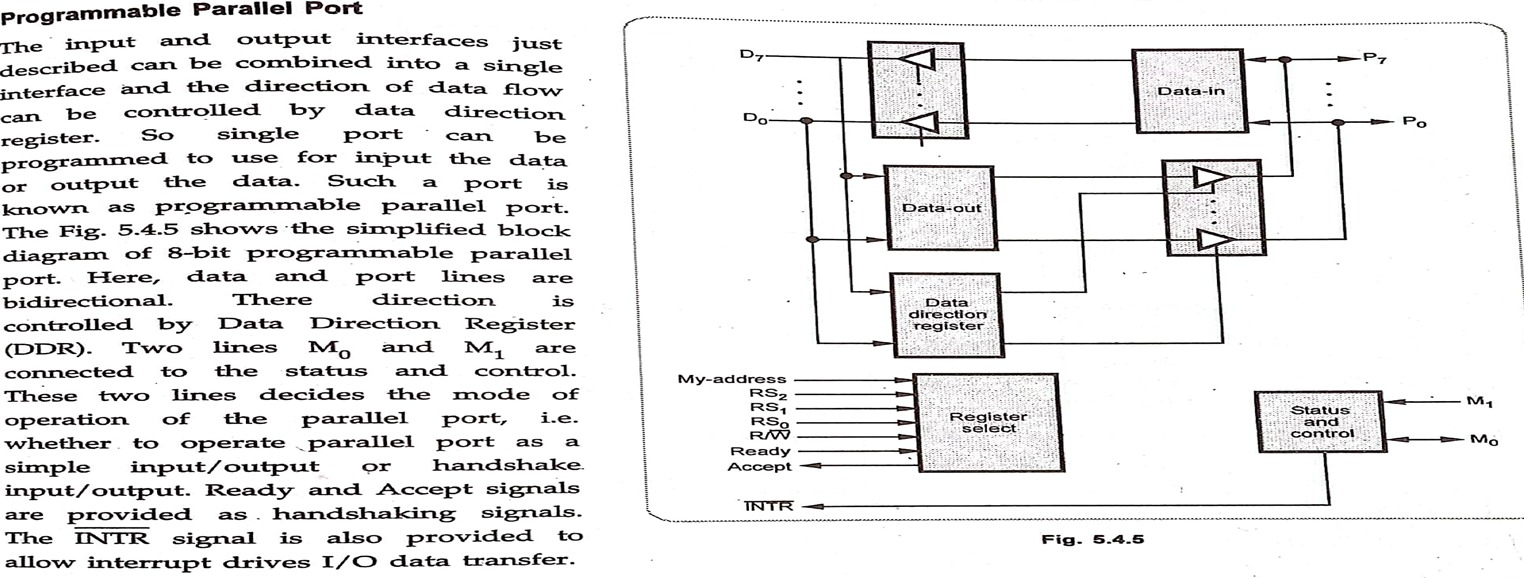
Therefore, 16 Chips will be needed to provide memory capacity of 2048 bytes.

Q.3 The main memory has 3-page frames (frame 0, frame 1 and frame 2). Pages from virtual memory are required in the order 2,3,2,1,5,2,4,5,3,2,5,2. Calculate the Hit ratio using FIFO replacement Algorithm

Q.4 What is the significance of page replacement? How many pages fault occurs in FIFO and LRU for the reference string 1,2,3,4,5,3,4,1,6,7,8,7,8,9,7,8,7,8,9,5,4,5,4,2 with 4-page frames?

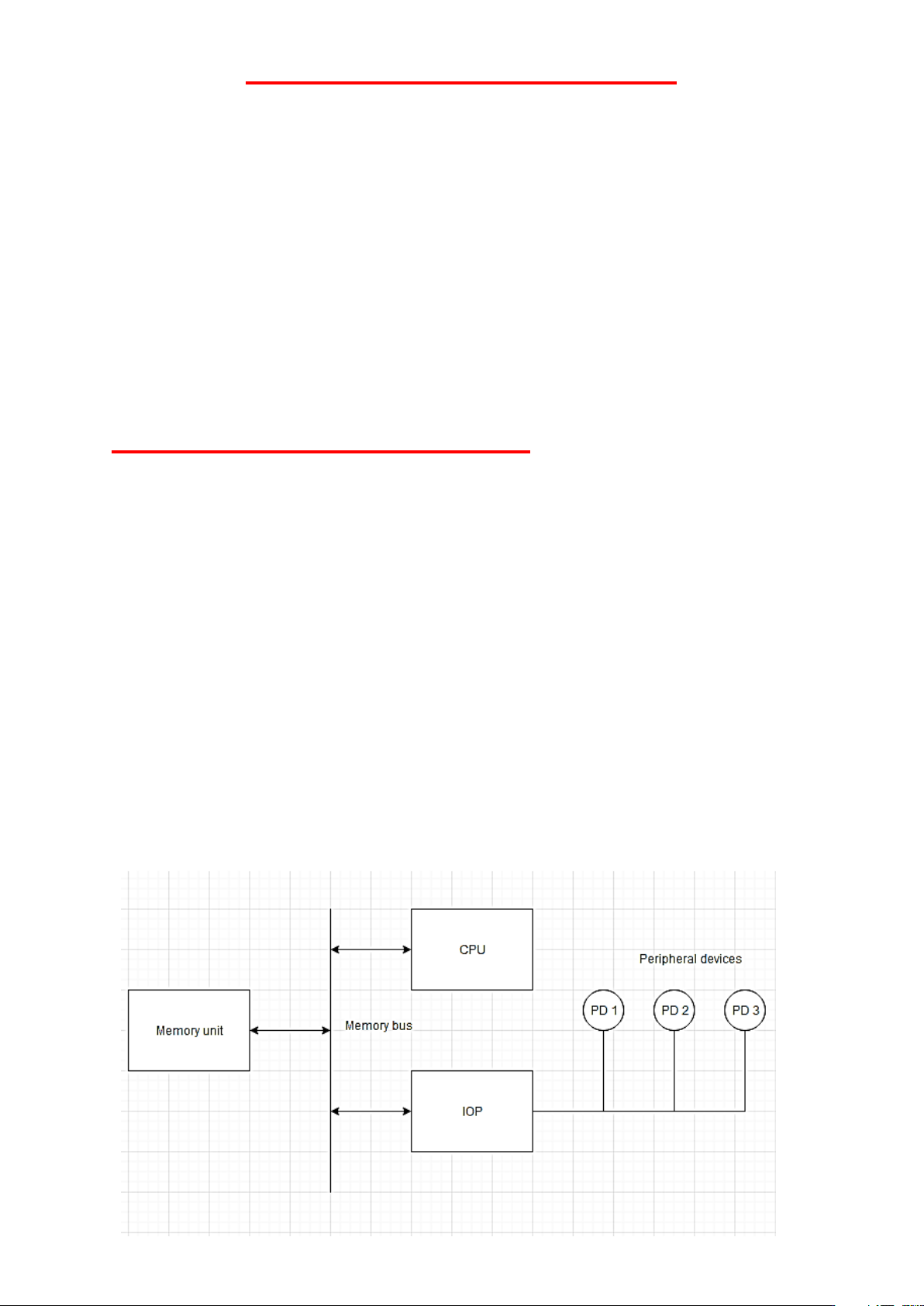
**Q.5 Examine the 8-bit programmable parallel port using a clear and well-organized diagram.**





**Q.6 Explain various types of I/O channels with the help of diagram.**

Input/Output Channels A channel is an independent hardware component that co-ordinate all I/O to a set of controllers. Computer systems that use I/O channel have special hardware components that handle all I/O operations. Channels use separate, independent and low cost processors for its functioning which are called Channel Processors. Channel processors are simple, but contains sufficient memory to handle all I/O tasks. When I/O transfer is complete or an error is detected, the channel controller communicates with the CPU using an interrupt, and informs CPU about the error or the task completion. Each channel supports one or more controllers or devices. Channel programs contain list of commands to the channel itself and for various connected controllers or devices. Once the operating system has prepared a list of I/O commands, it executes a single I/O machine instruction to initiate the channel program, the channel then assumes control of the I/O operations until they are completed.



**IBM 370 I/O Channel**

The I/O processor in the IBM 370 computer is called a Channel. A computer system configuration includes a number of channels which are connected to one or more I/O devices.

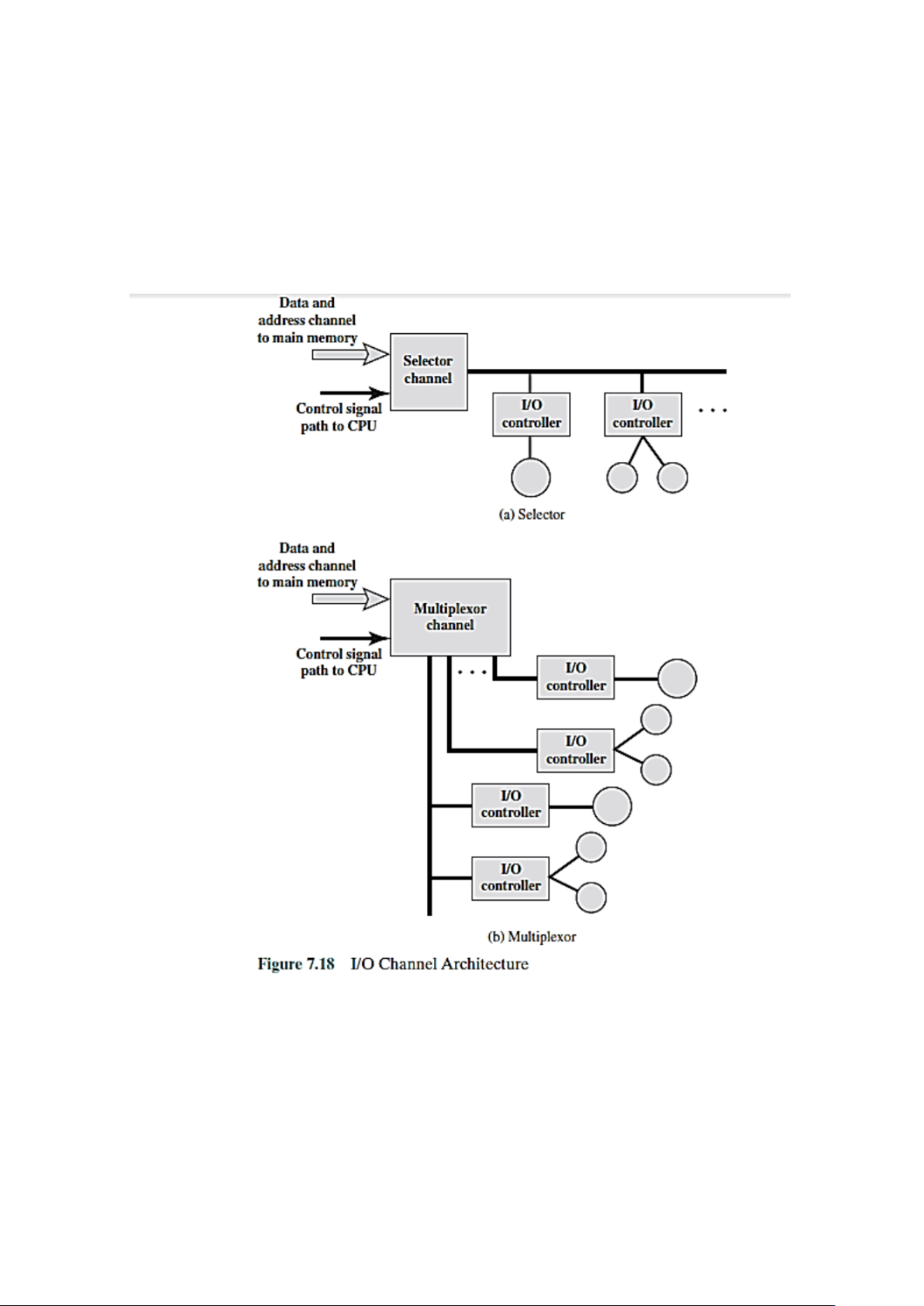
Categories of I/O Channels

Following are the different categories of I/O channels:

Multiplexer The Multiplexer channel can be connected to a number of slow and medium speed devices. It is capable of operating number of I/O devices simultaneously.

Selector This channel can handle only one I/O operation at a time and is used to control one high speed device at a time.

Block-Multiplexer It combines the features of both multiplexer and selector channels. The CPU directly can communicate with the channels through control lines.



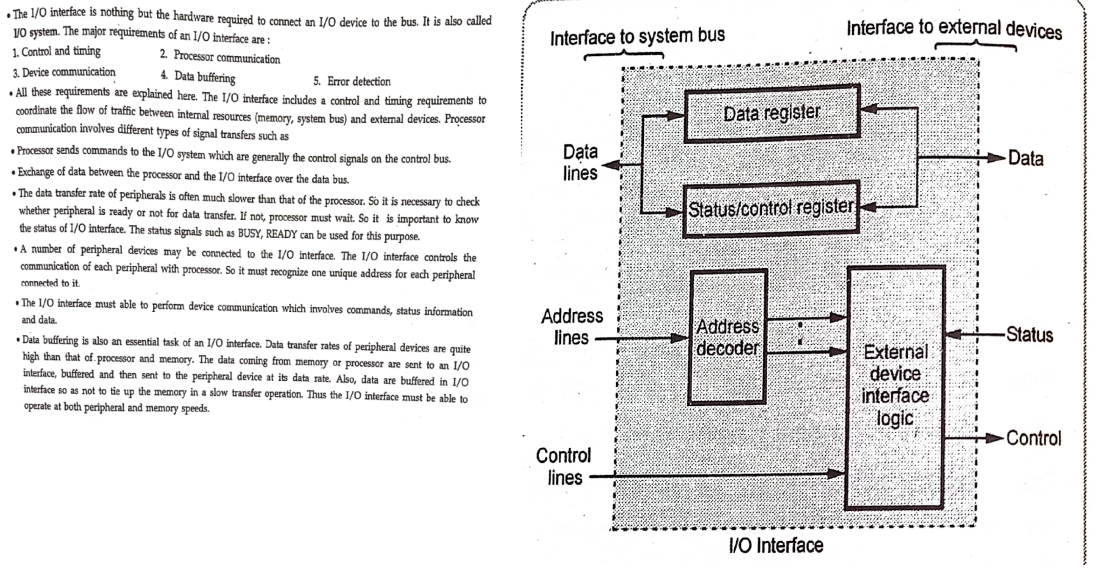
**Q.7 Determine the number of clock cycles that it takes to process 100 tasks in a six-segment pipeline.**

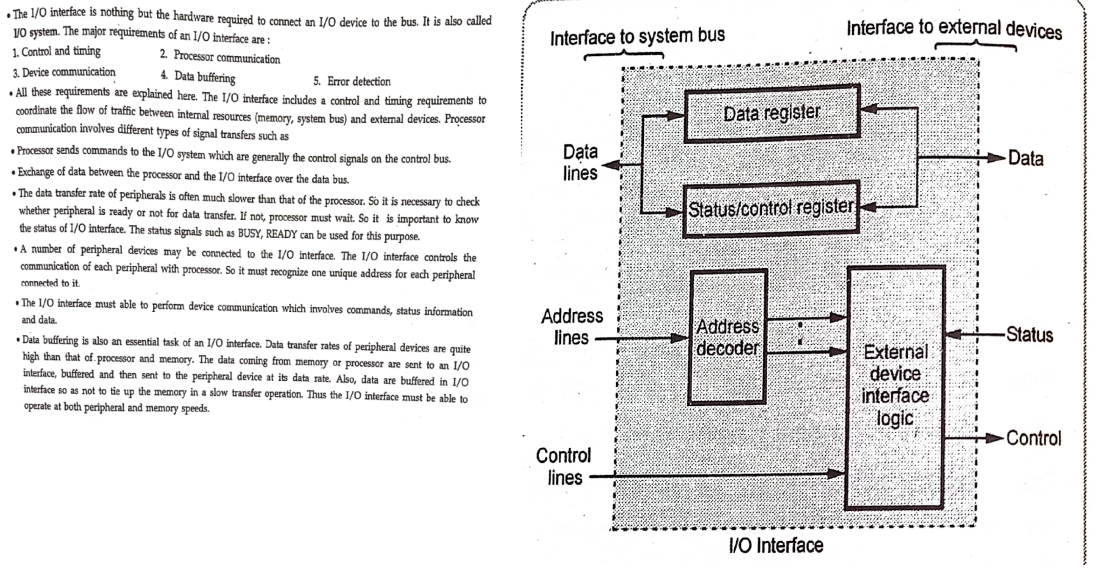
Given, No. of tasks (n) = 100

No. of the segments (k) = 6

No. of the clock cycles = **k + (n-1)** = 6+(100-1) = 6+99= **105 clock cycles.**

**Q.8 Discuss I/O interface with the help of its block diagram. What is the major requirement of an I/O interface?**





**Q.9 Describe Magnetic Disks and Magnetic Tapes in detail.**

Both Magnetic Tape and Magnetic Disk are the type of non-volatile magnetic memory and used to store the data. On the basis of architecture and features we can distinguish between both Magnetic Tape Memory and Magnetic Disk Memory. Following are the important differences between Magnetic Tape Memory and Magnetic Disk Memory.

| **Sr. No.** | **Key** | **Magnetic Tape Memory** | **Magnetic Disk Memory** |
| --- | --- | --- | --- |
| 1 | Definition | Magnetic tape is type of non-volatile memory uses thin plastic ribbon is used for storing data and as data use to be stored on ribbon so data read/write speed is slower due to which is mainly used for data backups. | On other hand Magnetic Disk is also type of non-volatile memory uses circular disk used for storing data. |
| 2 | Constituent | As mentioned above Magnetic tape memory is having plastic ribbon as main constituent. | On other hand Magnetic disk memory has metallic or plastic circular disk coated with magnetic oxide, as main constituent. |
| 3 | Cost Concern | Cost concern in case of Magnetic tape memory is less as plastic ribbon is much cheaper as compared to the circular disk as in case of magnetic disk memory. | While on other hand in case of Magnetic disk memory the cost concern is more as compared to that of Magnetic tape memory. |
| 4 | Reliability | As plastic ribbon is not much reliable and chances of its breakdown is more hence Magnetic tape memory is less reliable as compared to Magnetic Disk memory. | On other hand due to more breakdown prone circular disk Magnetic disk memory becomes more reliable than Magnetic tape memory. |
| 5 | Performance | Data store and retrieval is much slower in case of Magnetic tape memory due to which it is less efficient and has comparatively degrade performance as compared to Magnetic disk memory. | On other hand data storage and retrieval is faster in case of Magnetic disk memory and hence has better performance and more efficient as compared to Magnetic tape memory. |
| 6 | Usage | Due to low data transmission rate and less portability Magnetic tape memory mainly used for Data backup purposes. | On other hand Magnetic disk memory can be used as secondary storage. |

**Q.10 How many lines of the address bus must be used to access 2048 bytes of memory? How many of these lines will be common to all chips? How many lines must be decoded for chip select? Specify the size of the decoders if size of one RAM chip =128 x 8.**

Number of address line to access 2048 bytes = log**2**2048 = **11**  
Since there are 16 chips, so to distinguish among these chips bits required = log**2**16 = 4  
So, common lines to all chips = 11- 4 =**7**  
  
Since 7 lines are common to each chip, chip select will have = log**2**7 =**3 lines**

As there are 16 chips (2048X8/ 128X8=16), we will need **4 X 16 decoder**.

**Q.11 A non-pipeline system takes 50 ns to process a task. The same task can process in a six-segment  
pipeline with a clock cycle of 10 ns. Determine the speedup ratio of the pipeline for 100 tasks.  
What is the maximum speedup that can achieved?**

For a non- pipeline system, the total time taken to ‘n’ tasks is n t**n = 100 X 50 ns = 5000 ns.**

For a six- segment pipeline, the total time taken is [k+(n-1)] t**p = [6+(100-1)]X10 = 1050 ns.**

The speed up ratio of the pipeline = n t**n /** [k+(n-1)] t**p = 5000 ns. / 1050 ns. = 4.76**

maximum speedup = = t**n /** t**p = 50/10 = 5**

**Q.12. A cache memory has access time of 40 ns and main memory access time is 160 ns. Calculate the average access time of CPU if the hit ratio is 80%.**

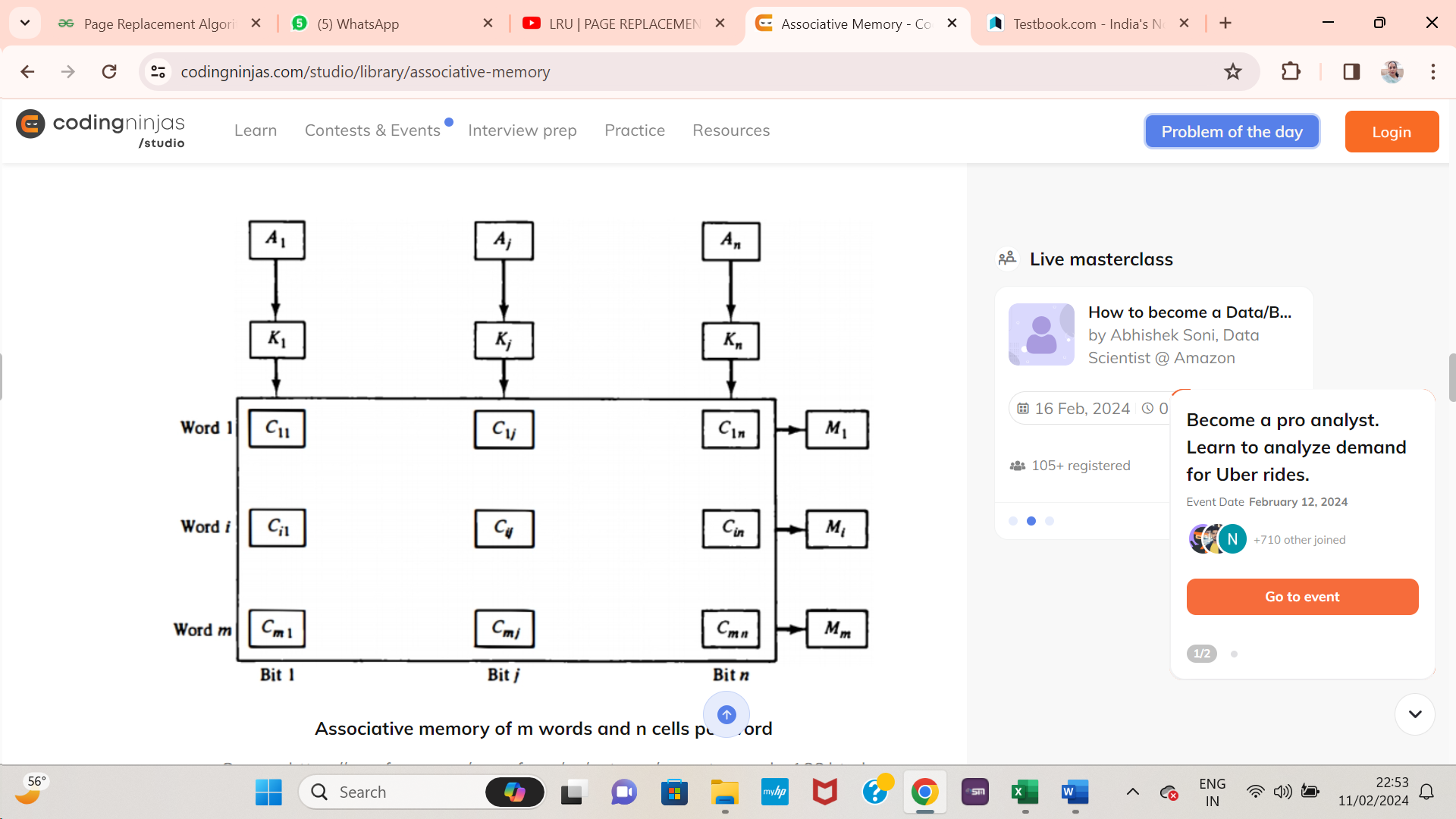
Effective Memory Access Time = Cache Hit Ratio \* Cache Access Time + Cache Miss \*(Cache Miss Penalty +Main Memory Access Time)

= 0.8 \* (40) + (1-0.8) \* (40+160) = **72 ns.**

**Q.13 Discuss hardware organization of associative memory with the help of its block diagram.**

Associative memory is a memory unit whose database can be accessed by the data or content instead of an address or memory location. It is also called content addressable memory or CAM. It helps optimize searching any data as the searching process does not involve addressing, but it works on data.

Following is the diagram showing the relationship between the memory array and external registers in associative memory:



Associative memory of m words and n cells per word

These are the cells marked by the letter C in the array with two subscripts. The first subscripts provide the word number, and the second one depicts the bit's position in the world. So, cell Cij is the cell for bit j in word i. Bit Ai in the argument register got compared with every bit in column j of the array provided that Ki = 1. This has been done for every column j = 1, 2, . . . , n. If one match occurs between every unmasked bit of the argument and the bits in word i, the corresponding bit M1 in the match register is set to 1. If one or more unmasked bits of the argument and the word do not match, M1 is cleared to 0.

**Read-Write operation**

When we read a word present in the associative memory, then the content of the word, or the part of the word, is specified. Words related to the specified content are memorized and marked for reading. No address or memory location is named when we write in associative memory. The memory itself can detect the space that will not store the word.

**Q.14 Consider a fully associative mapped cache of size 512 KB with block size 1 KB. There are 17 bits in the tag. Find-**

1. **Size of main memory**
2. **Tag directory size**

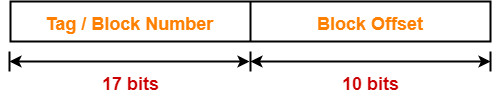
Given,

* Cache memory size = 512 KB
* Block size = Frame size = Line size = 1 KB
* Number of bits in tag = 17 bits

 We consider that the memory is byte addressable.

**Number of Bits in Block Offset-**

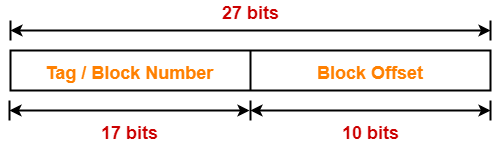
We have the Block size = 1 KB bytes = 210 bytes  
Thus, Number of bits in block offset = 10 bits



**Number of Bits in Physical Address-**

Number of bits in physical address = Number of bits in tag + Number of bits in block offset

= 17 bits +10 bits = 27 Bits



1. **Size of the Main Memory-**

We have Number of bits in physical address = 27 bits

So, the size of main memory = 227 bytes = **128 MB**

**Number of Lines in Cache-**

Total number of lines in cache = Cache size / Line size = 512 KB / 1KB  
= 512 lines = 29 lines  
Thus, Number of bits in line number = 9 bits

1. **Tag Directory Size-**

Tag directory size = Number of tags x Tag size

= Number of lines in cache x Number of bits in tag

= 29 x 17 bits

= 8704 bits

= 1088 bytes

**Thus, size of tag directory = 1088 bytes**

**\*\*One more extra question of similar type\*\***

**Extra Question: Consider a fully associative mapped cache of size 16 KB with block size 256 bytes. The size of main memory is 128 KB. Find-**

1. Number of bits in tag
2. Tag directory size

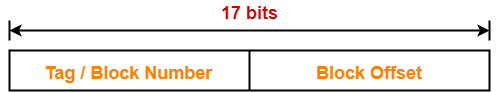
Solution: Given,

* Cache memory size = 16 KB
* Block size = Frame size = Line size = 256 bytes
* Main memory size = 128 KB

 We consider that the memory is byte addressable.

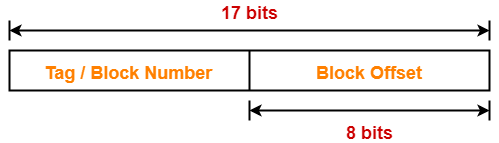
**Number of Bits in Physical Address-**

We have the size of main memory = 128 KB = 217 bytes  
Thus, Number of bits in physical address = 17 bits



**Number of Bits in Block Offset-**

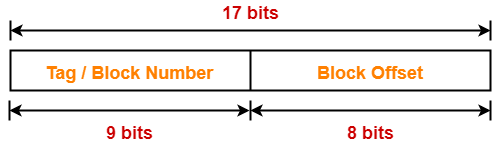
We have the Block size = 256 bytes = 28 bytes  
Thus, Number of bits in block offset = 8 bits



**Number of Bits in Tag-**

 Number of bits in tag = Number of bits in physical address – Number of bits in block offset

= 17-8 = **9-bits**



**Number of Lines in Cache-**

Total number of lines in cache = Cache size / Line size = 16 KB / 256 bytes  
= 214 bytes / 28 bytes = 26 lines  
Thus, Number of bits in line number = 6 bits

**Tag Directory Size-**

Tag directory size = Number of tags x Tag size

= Number of lines in cache x Number of bits in tag

= 26 x 9 bits

= 576 bits

= 72 bytes

**Thus, size of tag directory = 72 bytes**

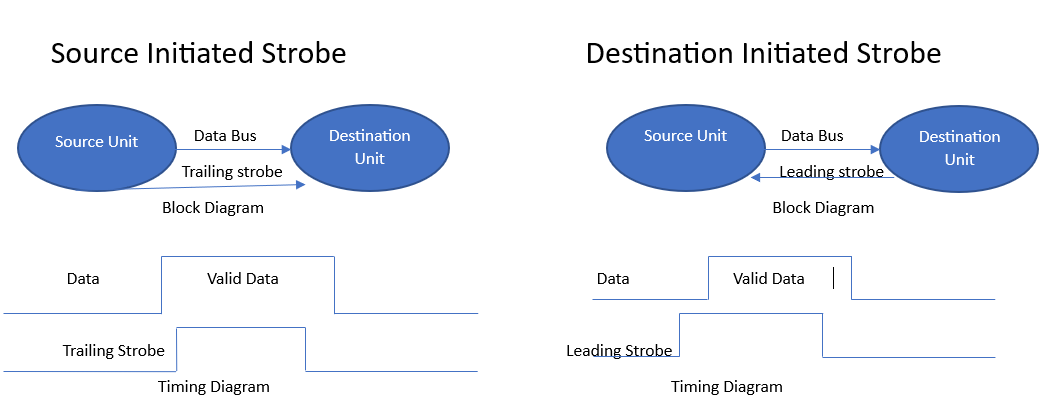
**Q.15 Distinguish between Strobe Control and Handshaking techniques for asynchronous data transfer.**

Asynchronous data transfer enables computers to send and receive data without having to wait for a real-time response. With this technique, data is conveyed in discrete units known as packets that may be handled separately.

Strobe Control Method for Data Transfer

Strobe control is a method used in [asynchronous](https://www.geeksforgeeks.org/difference-between-synchronous-and-asynchronous-transmission/) data transfer that synchronizes data flow between two devices. Bits are transmitted one at a time, independently of one another, and without the aid of a clock signal in asynchronous communication. To properly receive the data, the receiving equipment needs to be able to synchronize with the transmitting device.

Strobe control involves sending data along with a different signal known as the strobe signal. The strobe signal alerts the receiving device that the data is valid and ready to be read. The receiving device waits for the strobe signal before reading the data to ensure sure it is synchronized with its clock.



The strobe signal is usually generated by the transmitting device and is sent either before or after the data. If the strobe signal is sent before the data, it is called a leading strobe. If it is sent after the data, it is called a trailing strobe.

It is advantageous to utilize strobe control because it enables asynchronous data transfer, which is helpful when the participating devices have dissimilar clock rates or are not synchronized. The time of data transfer is also made more flexible by strobe control since the receiving device doesn’t have to synchronize with the transmitting device’s clock; instead, it can wait for the strobe signal before reading the data.

Overall, strobe control, which is frequently employed in a range of electronic devices and systems, is a helpful technique for assuring dependable data flow in asynchronous communication.

### Handshaking Method for Data Transfer

During an asynchronous data transfer, two devices manage their communication using **handshaking**. It is guaranteed that the transmitting and receiving devices are prepared to send and receive data. Handshakes are essential in asynchronous communication since there is no clock signal to synchronize the data transfer.

During handshaking, we use two types of signals mostly they are request-to-send (RTS) and clear-to-send (CTS). The receiving device is notified by an RTS signal when the transmitting equipment is ready to provide data. The receiving device responds with a CTS signal when it is ready to accept data.

once data is transmitted to the receiver end. the receiver generates a signal that it has done by sending an acknowledgment (ACK) signal. If the data is not successfully received, the receiving device will notify that a new transmission is necessary via a negative acknowledgment (NAK) signal.

The handshaking procedure guarantees synchronized and dependable data delivery. Additionally, it allows for flow management, preventing the transmitting device from sending the receiving device an excessive amount of data all at once. In order to offer flow control, handshaking signals are utilized to regulate the rate at which data is sent.

The Handshaking Method in asynchronous data transfer is used in different devices for the transfer of data to ensure reliable communication.

## Advantages of Asynchronous Data Transfer

* Because asynchronous data transfer sends data in discrete, independently processable pieces, it enables faster data transfer speeds.
* This method is more effective than synchronous data transfer because there is no need for the receiver to respond.
* Transmission is done by making large files or data sets into smaller packets and sending them in parallel cuts the duration time.

## Disadvantages of Asynchronous Data Transfer

* Asynchronous data transfer requires more complex programming and it may be possible that some data may get corrupted or lose data if packets are not received in the correct order or are lost during transmission.
* As we know there will be no real-time communication in asynchronous data transport can be more prone to errors than synchronous data transfer.

**Q.16 Demonstrate the block diagrams and functionalities of integrated circuit chips for RAM and ROM.**

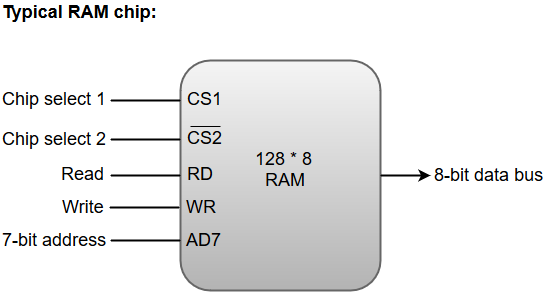
The main memory acts as the central storage unit in a computer system. It is a relatively large and fast memory which is used to store programs and data during the run time operations.The primary technology used for the main memory is based on semiconductor integrated circuits. The integrated circuits for the main memory are classified into two major units.

1. RAM (Random Access Memory) integrated circuit chips
2. ROM (Read Only Memory) integrated circuit chips

The RAM integrated circuit chips are further classified into two possible operating modes, **static** and **dynamic**.

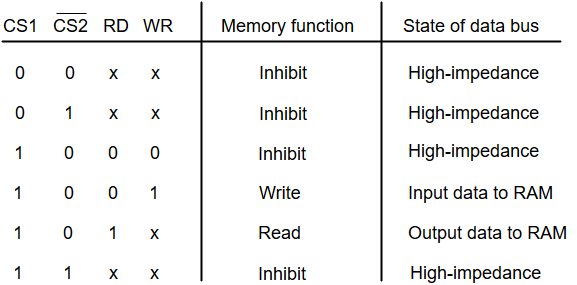
The primary compositions of a static RAM are flip-flops that store the binary information. The nature of the stored information is volatile, i.e. it remains valid as long as power is applied to the system. The static RAM is easy to use and takes less time performing read and write operations as compared to dynamic RAM.

The dynamic RAM exhibits the binary information in the form of electric charges that are applied to capacitors. The capacitors are integrated inside the chip by MOS transistors. The dynamic RAM consumes less power and provides large storage capacity in a single memory chip.RAM chips are available in a variety of sizes and are used as per the system requirement. The following block diagram demonstrates the chip interconnection in a 128 \* 8 RAM chip.



* A 128 \* 8 RAM chip has a memory capacity of 128 words of eight bits (one byte) per word. This requires a 7-bit address and an 8-bit bidirectional data bus.
* The 8-bit bidirectional data bus allows the transfer of data either from memory to CPU during a **read** operation or from CPU to memory during a **write** operation.
* The **read** and **write** inputs specify the memory operation, and the two chip select (CS) control inputs are for enabling the chip only when the microprocessor selects it.
* The bidirectional data bus is constructed using **three-state buffers**.
* The output generated by three-state buffers can be placed in one of the three possible states which include a signal equivalent to logic 1, a signal equal to logic 0, or a high-impedance state.

The following function table specifies the operations of a 128 \* 8 RAM chip.



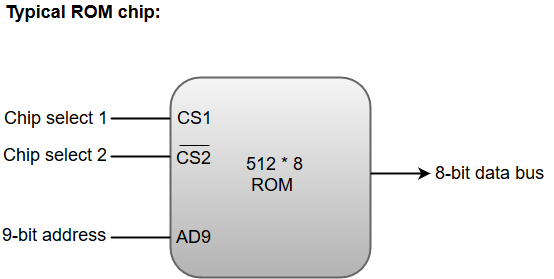
From the functional table, we can conclude that the unit is in operation only when CS1 = 1 and CS2 = 0. The bar on top of the second select variable indicates that this input is enabled when it is equal to 0.

The primary component of the main memory is RAM integrated circuit chips, but a portion of memory may be constructed with ROM chips.

A ROM memory is used for keeping programs and data that are permanently resident in the computer.

Apart from the permanent storage of data, the ROM portion of main memory is needed for storing an initial program called a **bootstrap loader**. The primary function of the **bootstrap loader** program is to start the computer software operating when power is turned on.

ROM chips are also available in a variety of sizes and are also used as per the system requirement. The following block diagram demonstrates the chip interconnection in a 512 \* 8 ROM chip.

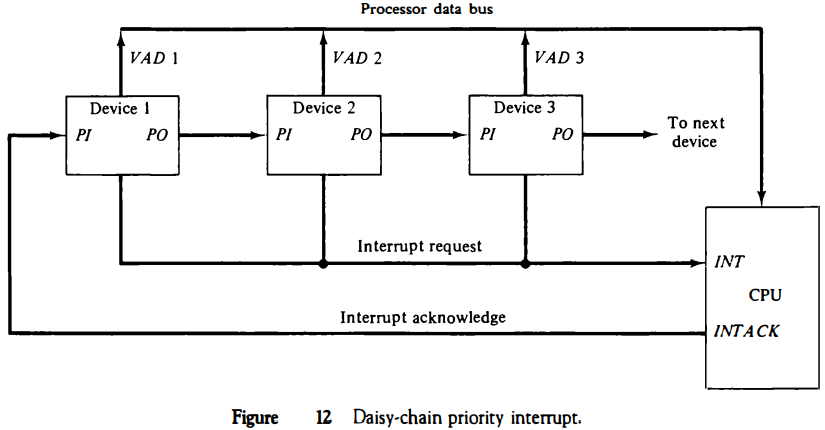


* A ROM chip has a similar organization as a RAM chip. However, a ROM can only perform read operation; the data bus can only operate in an output mode.
* The 9-bit address lines in the ROM chip specify any one of the 512 bytes stored in it.
* The value for chip select 1 and chip select 2 must be 1 and 0 for the unit to operate. Otherwise, the data bus is said to be in a high-impedance state.

**Q.17 Describe the concept of daisy-chaining. Devise a single-stage daisy-chaining priority logic circuit and elucidate its operation, detailing how it manages interrupt-initiated I/O requests.**

**The daisy-chaining** method of establishing priority consists of a serial connection of all devices that request an interrupt. **The device** with the highest priority is placed in the first position, followed by lower-priority devices up to the device with the lowest priority, which is placed last in the chain. **This method** of connection between three devices and the CPU is shown in Figure. The interrupt request line is common to all devices and forms a wired logic connection. **If any device** has its interrupt signal in the low-level state, the interrupt line goes to the low-level state and enables the interrupt input in the CPU. **When no interrupts** are pending, the interrupt line stays in the high-level state and no interrupts are recognized by the CPU. **This is equivalent** to a negative logic OR operation. The CPU responds to an interrupt request by enabling the interrupt acknowledge line. This signal is received by device 1 at its PI (priority in) input. **The acknowledge** signal passes on to the next device through the PO (priority out) output only if device 1 is not requesting an interrupt. **If device 1 has a pending interrupt**, it blocks the acknowledge signal from the next device by placing a 0 in the PO output. **It then proceeds** to insert its own interrupt vector address (VAD) into the data bus for the CPU to use during the interrupt cycle.

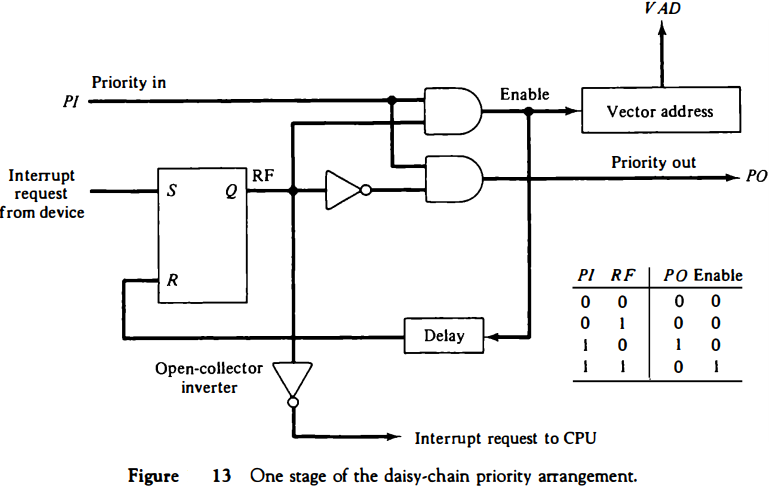
**A device** with a 0 in its PI input generates a 0 in its PO output to inform the next-lower-priority device that the acknowledge signal has been blocked. **A device**that is requesting an interrupt and has a 1 in its PI input will intercept the acknowledge signal by placing a 0 in its PO output. **If the device**does not have pending interrupts, it transmits the acknowledge signal to the next device by placing a 1 in its PO output.



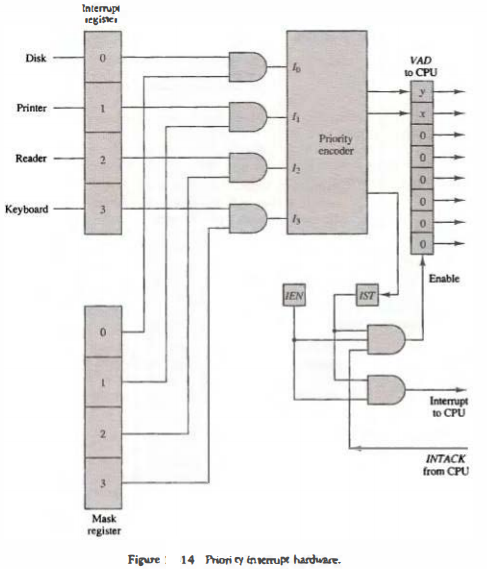
**Thus the device** with PI = 1 and PO = 0 is the one with the highest priority that is requesting an interrupt, and this device places its VAD on the data bus. **The daisy chain** arrangement gives the highest priority to the device that receives the interrupt acknowledge signal from the CPU. **The farther** the device is from the first position, the lower is its priority. **Figure** shows the internal logic that must be included within each device when connected in the daisy-chaining scheme. The device sets its RF flip-flop when it wants to interrupt the CPU. **The output of the RF flip-flop** goes through an open-collector inverter, a circuit that provides the wired logic for the common interrupt line. **If PI = 0, both PO**and the enable line to VAD are equal to 0, irrespective of the value of RF. If PI = 1 and RF = 0, then PO = 1 and the vector address is disabled. **This condition** passes the acknowledge signal to the next device through PO. **The device** is active when PI = 1 and RF = 1. This condition places a 0 in PO and enables the vector address for the data bus. **It is assumed** that each device has its own distinct vector address. The RF flip-flop is reset after a sufficient delay to ensure that the CPU has received the vector address.

**Q.18 Design parallel priority interrupt hardware for a system with four interrupt sources using priority encoder.**

**The parallel priority** interrupt method uses a register whose bits are set separately by the interrupt signal from each device. **Priority** is established according to the position of the bits in the register. **In addition** to the interrupt register, the circuit may include a mask register whose purpose is to control the status of each interrupt request.



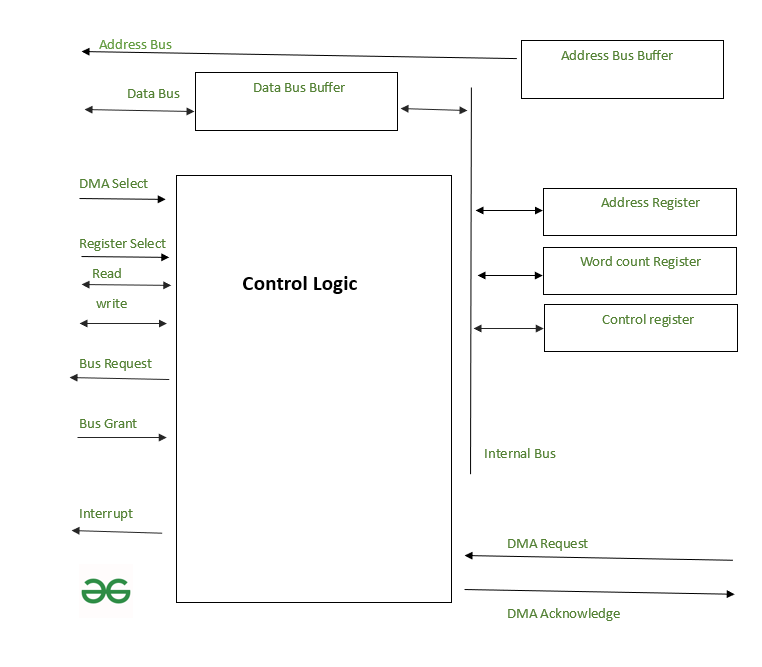
**The mask register** can be programmed to disable lower-priority interrupts while a higher-priority device is being serviced. **It can also provide** a facility that allows a high-priority device to interrupt the CPU while a lower-priority device is being serviced. **The priority logic**for a system of four interrupt sources is shown in Figure. **It consists** of an interrupt register whose individual bits are set by external conditions and cleared by program instructions. **The magnetic disk,** being a high-speed device, is given the highest priority. The printer has the next priority, followed by a character reader and a keyboard. **The mask register** has the same number of bits as the interrupt register. By means of program instructions, it is possible to set or reset any bit in the mask register. **Each interrupt bit** and its corresponding mask bit are applied to an AND gate to produce the four inputs to a priority encoder. **In this way** an interrupt is recognized only if its corresponding mask bit is set to 1 by the program. The priority encoder generates two bits of the vector address, which is transferred to the CPU. **Another output** from the encoder sets an interrupt status flip-flop lST when an interrupt that is not masked occurs. The interrupt enable flip-flop lEN can be set or cleared by the program to provide an overall control over the interrupt system. **The outputs** of IST ANDed with IEN provide a common interrupt signal for the CPU. **The interrupt** acknowledge INTACK signal from the CPU enables the bus buffers in the output register and a vector address VAD is placed into the data bus.



Q.19 **Explain the Direct Memory Transfer (DMA) techniques used for data transfer in a computer system in detail.**

DMA (Direct memory access) is the special feature within the computer system that transfers the data between memory and peripheral devices (like hard drives) without the intervention of the CPU. In other words, for large data transfer like disk drives, it will be wasteful to use expensive general-purpose processors in which status bits are to be watched and fed the data into the controller register one byte at a time which is termed **Programmed I/O**. Computers avoid burdening the CPU so, they shift the work to a Direct Memory Access controller. Let’s see the workings of this in detail.

To initiate the DMA transfer the host writes a DMA command block into the memory. This block contains the pointer to the source of the transfer, the pointer to the destination of the transfer, and the count of the number of bytes to be transferred. This command block can be more complex which includes the list of sources and destination addresses that are not contiguous. CPU writes the address of this command block and goes to other work. DMA controller proceeds to operate the memory bus directly, placing the address on it without the intervention of the main CPU.



The device controller places a signal on the DMA request wire when a word of data is available for transfer. This cause DMA controller to seize the memory bus of CPU and place the desired address on the DMA acknowledge wire. Up on successful data transfer the device controller receives the DMA acknowledge and then it removes the DMA request signal.

When the entire transfer is finished, DMA controller interrupts the CPU. This entire process is depicted in the above diagram. DMA controller seizes the memory bus and CPU momentarily prevented from accessing main memory. Although it can access the data items in its cache. This **cycle stealing (**Seizing the memory bus temporarily and preventing the CPU from accessing it**)**slows down the CPU computation, shifting the data transfer to DMA controller generally improves the total system performance. Some of the computer architecture used physical memory address for DMA, but other uses virtual addresses (DVMA). Direct virtual memory access performs data transfer between memory mapped I/O without the use of main memory.

Q.20 Differentiate between the following:

1. RISC vs CISC Processor
2. Hardwired Control Unit vs Micro-programmed Control Unit
3. Vectored and Non- vectored Interrupts
4. Write -Through and Write Back policy of Cache

Q.21 Compare and contrast the following page replacement policies.

1. LRU page replacement
2. FIFO page replacement
3. Optimal page replacement

Assuming three frames, determine how many page faults would occur for the following page reference string: 1,2,3,4,2,1,5,6,2,1,2,1,5,6 in each of the mentioned replacement algorithm.

**Q.22 Explain the concept of Virtual memory. Also, elaborate how memory management is done by using paging and segmentation.**

Virtual memory is a method that computers use to manage storage space to keep systems running quickly and efficiently. Using the technique, operating systems can transfer data between different types of storage, such as random-access memory (RAM), also known as main memory, and hard drive or solid-state disk storage. At any particular time, the computer only needs enough active memory to support active processes. The system can move those that are dormant into virtual memory until needed.

Virtual memory uses both the computer's software and hardware to work. It transfers processes between the computer's RAM and hard disk by copying any files from the computer's RAM that aren't currently in use and moving them to the hard disk. By moving unused files to the hard disk, a computer frees up space in its RAM to perform current tasks, such as opening a new application. If the computer later needs to use its RAM for a more urgent task, it can again swap files to make the most of the available RAM. RAM is a limited resource stored on chips in the computer’s CPU. Installing more RAM chips can be expensive, so virtual memory allows the computer to move files between systems as needed to optimize its use of the available RAM.

The two ways computers handle virtual memory are through paging and segmenting.

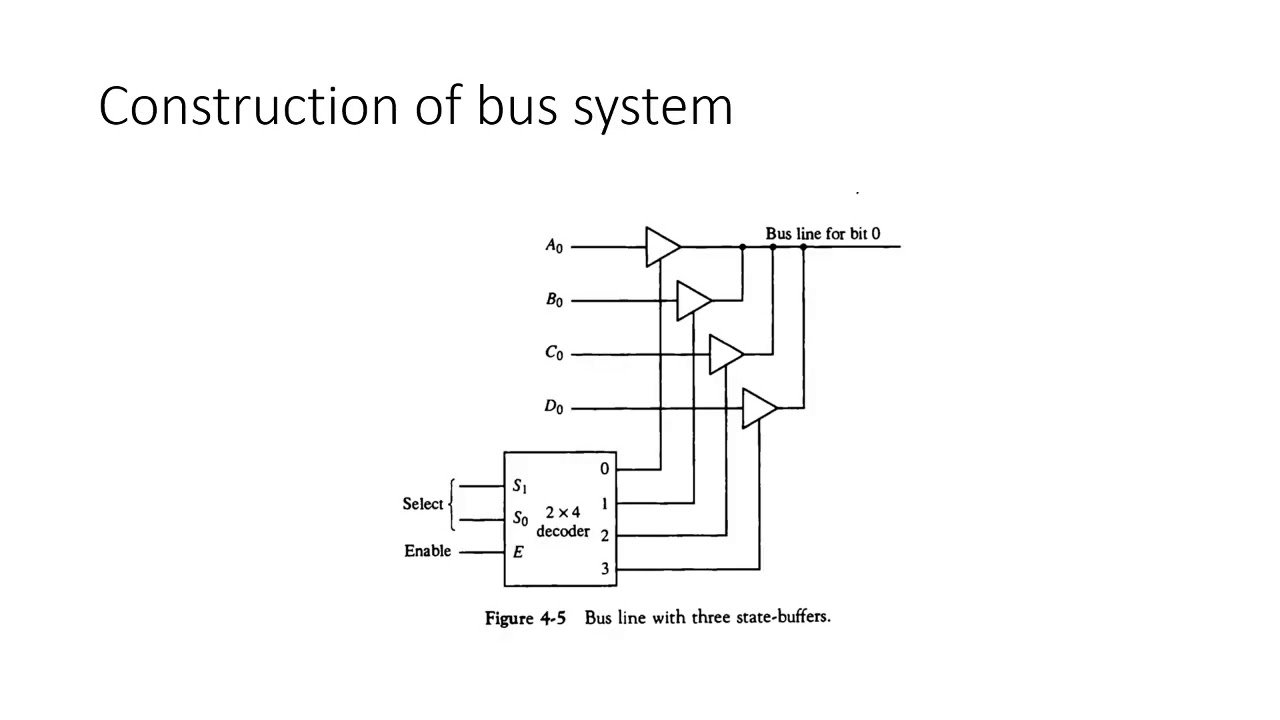
1. **Paging** is a virtual memory technique that separates memory into sections called paging files. When a computer reaches its RAM limits, it transfers any currently unused pages into the part of its hard drive used for virtual memory. The computer performs this process using a swap file, a designated space within its hard drive for extending the virtual memory of the computer's RAM. By moving unused files into its hard drive, the computer frees its RAM space for other memory tasks and ensures that it doesn't run out of real memory.

As part of this process, the computer uses page tables, which translate virtual addresses into the physical addresses that the computer's memory management unit (MMU) uses to process instructions. The MMU communicates between the computer's OS and its page tables. When the user performs a task, the OS searches its RAM for the processes to conduct the task. If it can't find the processes to complete the task in RAM, the MMU prompts the OS to move the required pages into RAM and uses a page table to note the new storage location of the pages.

1. **Segmentation** is another method of managing virtual memory. A segmentation system divides virtual memory into varying lengths and moves any segments that aren't in use from the computer's virtual memory space to its hard drive. Like page tables, segment tables track whether the computer stores the segment in memory or a physical address. Segmentation differs from paging because it divides memory into sections of varying lengths, while paging divides memory into units of equal size. With paging, the hardware determines the size of a section, but the user can select the length of a segment in a segmentation system.

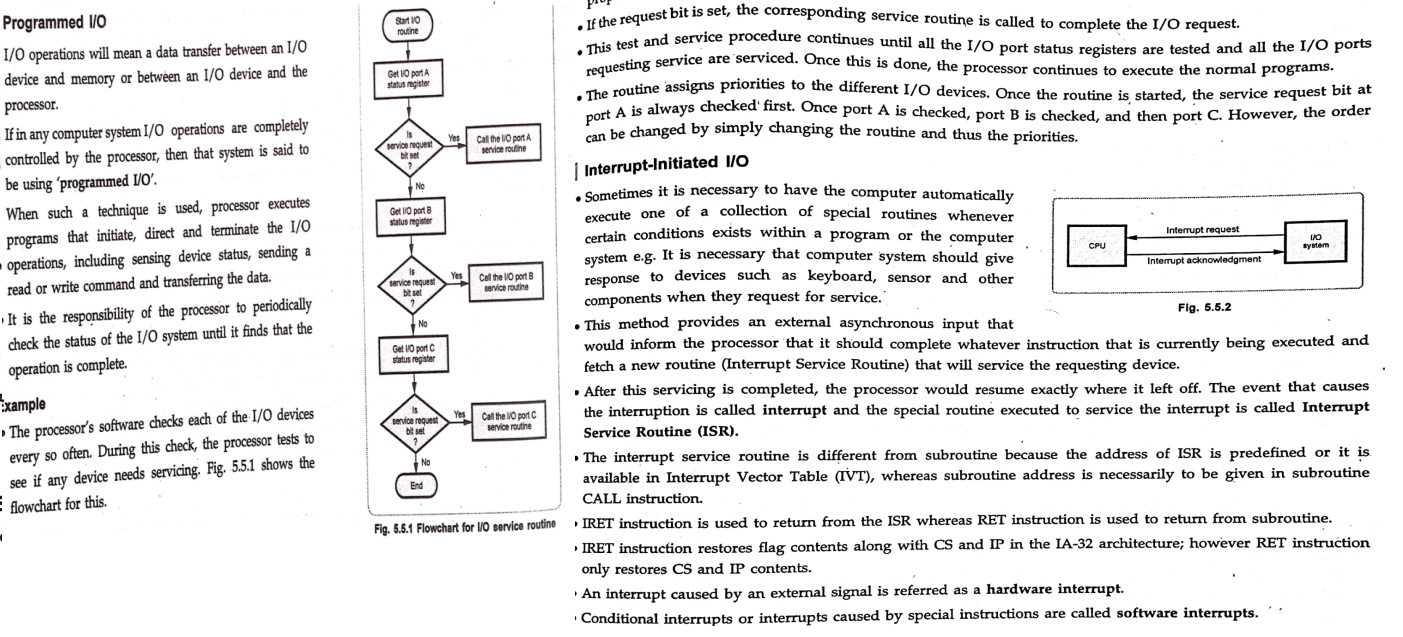
Segmentation is often slower than paging, but it offers the user more control over how to divide memory and may make it easier to share data between processes. You can customize the segments based on the machine's purpose and usage. Casual computer users may prefer a paging system because it automatically handles memory divisions.

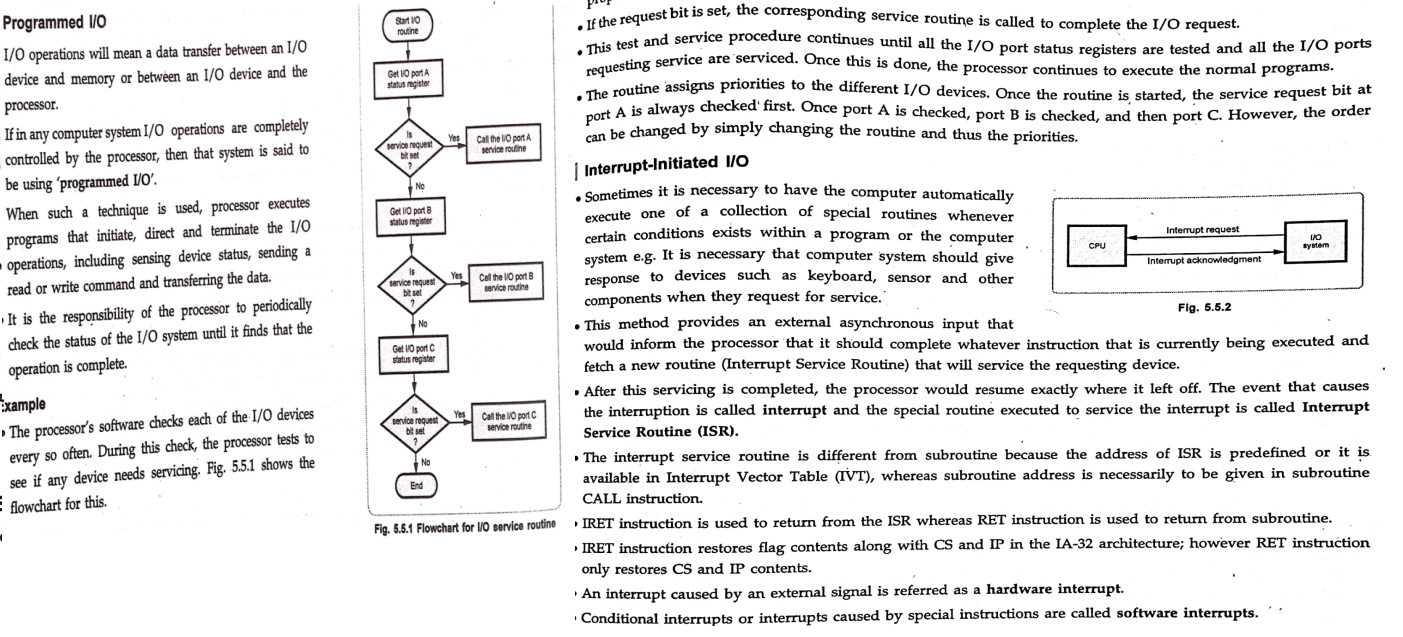
Q.23 **Design a common bus system for a digital computer system having four registers of 4-bit using 3-state buffers.**



\*\*\* Repeat this gate buffer set diagram three more times for bit ‘1’, bit’2’, and bit’3’ and extend the output lines of the decoder to remaining gate buffers in same manner.

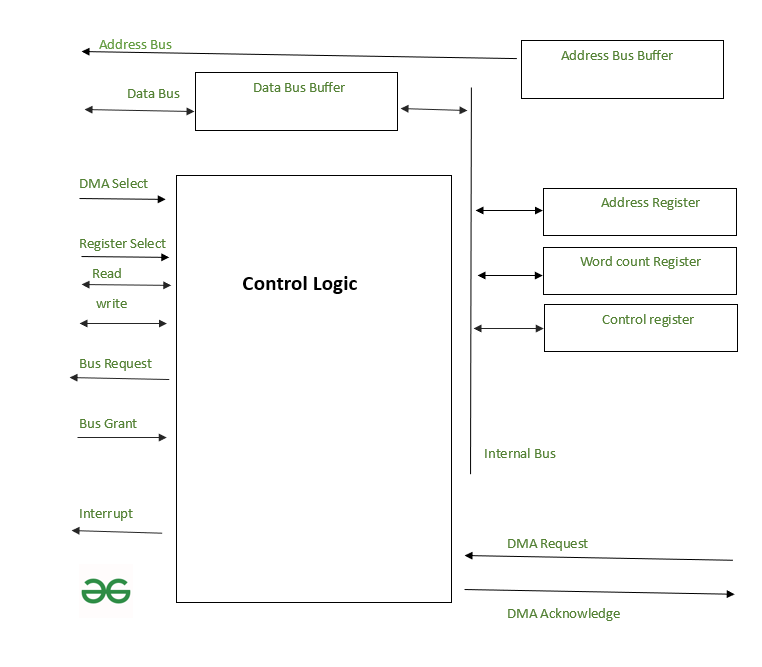
Q.24 **Explain the three different modes of transfer: Programmed I/O, Interrupt- Initiated, and Direct Memory Transfer. Show the flowchart for CPU program flow in Programmed I/O.**





DMA (Direct memory access) is the special feature within the computer system that transfers the data between memory and peripheral devices (like hard drives) without the intervention of the CPU. In other words, for large data transfer like disk drives, it will be wasteful to use expensive general-purpose processors in which status bits are to be watched and fed the data into the controller register one byte at a time which is termed **Programmed I/O**. Computers avoid burdening the CPU so, they shift the work to a Direct Memory Access controller. Let’s see the workings of this in detail.

To initiate the DMA transfer the host writes a DMA command block into the memory. This block contains the pointer to the source of the transfer, the pointer to the destination of the transfer, and the count of the number of bytes to be transferred. This command block can be more complex which includes the list of sources and destination addresses that are not contiguous. CPU writes the address of this command block and goes to other work. DMA controller proceeds to operate the memory bus directly, placing the address on it without the intervention of the main CPU.



The device controller places a signal on the DMA request wire when a word of data is available for transfer. This cause DMA controller to seize the memory bus of CPU and place the desired address on the DMA acknowledge wire. Up on successful data transfer the device controller receives the DMA acknowledge and then it removes the DMA request signal.

When the entire transfer is finished, DMA controller interrupts the CPU. This entire process is depicted in the above diagram. DMA controller seizes the memory bus and CPU momentarily prevented from accessing main memory. Although it can access the data items in its cache. This **cycle stealing (**Seizing the memory bus temporarily and preventing the CPU from accessing it**)**slows down the CPU computation, shifting the data transfer to DMA controller generally improves the total system performance. Some of the computer architecture used physical memory address for DMA, but other uses virtual addresses (DVMA). Direct virtual memory access performs data transfer between memory mapped I/O without the use of main memory.

**Q.25 Distinguish between the following:**

1. **Isolated I/O vs Memory Mapped**

As a CPU needs to communicate with the various memory and input-output devices (I/O) as we know data between the processor and these devices flow with the help of the system bus. There are three ways in which system bus can be allotted to them:

1. Separate set of address, control and data bus to I/O and memory.
2. Have common bus (data and address) for I/O and memory but separate control lines.
3. Have common bus (data, address, and control) for I/O and memory.

In first case it is simple because both have different set of address space and instruction but require more buses.

#### Isolated I/O –

Then we have Isolated I/O in which we Have common bus(data and address) for I/O and memory but separate read and write control lines for I/O. So when CPU decode instruction then if data is for I/O then it places the address on the address line and set I/O read or write control line on due to which data transfer occurs between CPU and I/O. As the address space of memory and I/O is isolated and the name is so. The address for I/O here is called ports. Here we have different read-write instruction for both I/O and memory.

#### Memory Mapped I/O –

In this case every bus in common due to which the same set of instructions work for memory and I/O. Hence, we manipulate I/O same as memory and both have same address space, due to which addressing capability of memory become less because some part is occupied by the I/O.

**Differences between memory mapped I/O and isolated I/O –**

| **Isolated I/O** | **Memory Mapped I/O** |
| --- | --- |
| Memory and I/O have separate address space | Both have same address space |
| All address can be used by the memory | Due to addition of I/O addressable memory become less for memory |
| Separate instruction control read and write operation in I/O and Memory | Same instructions can control both I/O and Memory |
| In this I/O address are called ports. | Normal memory address are for both |
| More efficient due to separate buses | Lesser efficient |
| Larger in size due to more buses | Smaller in size |
| It is complex due to separate logic is used to control both. | Simpler logic is used as I/O is also treated as memory only. |

### Advantages of Memory-Mapped I/O:

**Faster I/O Operations:**Memory-mapped I/O allows the CPU to access I/O devices at the same speed as it accesses memory. This means that I/O operations can be performed much faster compared to isolated I/O.

**Simplified Programming:** Memory-mapped I/O simplifies programming as the same instructions can be used to access memory and I/O devices. This means that software developers do not have to use specialized I/O instructions, which can reduce programming complexity.

**Efficient Use of Memory Space:**Memory-mapped I/O is more memory-efficient as I/O devices share the same address space as the memory. This means that the same memory address space can be used to access both memory and I/O devices.

### Disadvantages of Memory-Mapped I/O:

**Limited I/O Address Space:**Memory-mapped I/O limits the I/O address space as I/O devices share the same address space as the memory. This means that there may not be enough address space available to address all I/O devices.

**Slower Response Time:**If an I/O device is slow to respond, it can delay the CPU’s access to memory. This can lead to slower overall system performance.

### Advantages of Isolated I/O:

**Large I/O Address Space:**Isolated I/O allows for a larger I/O address space compared to memory-mapped I/O as I/O devices have their own separate address space.

**Greater Flexibility:**Isolated I/O provides greater flexibility as I/O devices can be added or removed from the system without affecting the memory address space.

**Improved Reliability:**Isolated I/O provides better reliability as I/O devices do not share the same address space as the memory. This means that if an I/O device fails, it does not affect the memory or other I/O devices.

### Disadvantages of Isolated I/O:

**Slower I/O Operations:**Isolated I/O can result in slower I/O operations compared to memory-mapped I/O as it requires the use of specialized I/O instructions.

**More Complex Programming:**Isolated I/O requires specialized I/O instructions, which can lead to more complex programming.

1. **SRAM vs DRAM**

When the power to the PC or laptop is turned off, the information saved in the Random Access Memory (RAM) is gone. The BIOS may be used to examine the information saved in RAM. It is also known as the computer system’s primary memory, temporary memory, cache memory, or volatile memory. The two types of RAM are SRAM and DRAM.

SRAM stands for static random-access memory. SRAM is a form of random-access memory (RAM) that stores each bit using latching circuitry (flip-flops). It is a volatile memory, which means that data is lost when the power is turned off. SRAM must be updated on a regular basis. It is quicker and more costly than DRAM, and it is often used for a CPU’s cache and internal registers.

DRAM stands for dynamic random-access memory. It is a form of RAM that allows each bit of data to be stored in its own capacitor within an integrated circuit. It is a kind of computer memory that may be found in any current desktop computer. DRAM is named dynamic because it requires regular modification or activity, such as refreshing, to keep the data intact. It is employed in the implementation of main memory.

|  |  |
| --- | --- |
| **SRAM** | **DRAM** |
| It can store data as long as electricity is available. | It saves data for as long as the power is on or for a few moments if the power is turned off. |
| Because capacitors aren’t utilized, there’s no need to refresh. | The contents of the capacitor must be updated on a regular basis in order to store information for a longer amount of time. |
| SRAM has a storage capacity of 1 MB to 16 MB in most cases. | DRAM, which is often found in tablets and smartphones, has a capacity of 1 GB to 2 GB. |
| The storage capacity of SRAM is low. | The storage capacity of DRAM is higher than SRAM. |
| SRAM is more expensive than DRAM. | DRAM is less expensive than SRAM. |
| It is comparatively faster. | It is comparatively slower. |
| The power consumption is minimal, and the access speed is quick. | The cost of production is low, and the memory capacity is higher. |
| SRAM is used in cache memories. | DRAM is used in main memories. |

1. **Serial Communication vs Parallel Communication**

To transfer the data between various devices, like laptops and computers, two methods come in handy, namely parallel transmission and serial transmission. But there is a primary difference between serial and parallel transmission, although they are similar in some aspects. In the case of serial transmission of data, the data gets transferred bit by bit. 8-bits are conveyed at a time in serial transmission, with a start bit and a stop bit.

* All long-distance communication and most computer networks employ serial communication.
* Serial computer buses are becoming more common, even across shorter distances, since newer serial technologies' greater signal integrity and transmission speeds have begun to outperform the parallel bus's simplicity advantage.
* The majority of communication systems use serial mode. Serial networks may be extended over vast distances for far less money since fewer physical wires are required.

On the other hand, in parallel transmission, only a character or a single byte (or 8 bits) is transmitted at any given time.

* parallel interface comprises parallel wires that individually contain data and other cables that allow the transmitter and receiver to communicate. Therefore, the wires for a similar transmission system are put in a single physical thread to simplify installation and troubleshooting.
* A large amount of data must be delivered across connection lines at high speeds that match the underlying hardware.
* The data stream must be transmitted through "n" communication lines, which necessitates using many wires. This is an expensive mode of transportation; hence it is usually limited to shorter distances.

1. **Primary memory vs Secondary memory**

Primary memory is used for temporarily storing data that is actively being used by the computer's CPU, while secondary memory (storage) is used for long-term data storage, such as files and programs that are not currently in use. Listed below are some other primary and secondary memory differences.

|  |  |  |
| --- | --- | --- |
| **Comparison Parameters** | **Primary Memory** | **Secondary Memory** |
| Storage validity | Primary memory is the main memory and stores data temporarily. | Secondary memory is the external memory and stores data permanently. |
| Access | The CPU can directly access the data. | The CPU cannot directly access the data. |
| Volatility | Primary memory is volatile. It loses data in case of a power outage. | Secondary memory is non-volatile; data is stored even during a power failure. |
| Storage | Data is stored inside costly semiconductor chips. | Data is stored on external hardware devices like hard drives, floppy disks, etc. |
| Division | It can be divided into [RAM and ROM](https://www.shiksha.com/online-courses/articles/difference-between-ram-and-rom/) | They do not have such a classification. Secondary memories are permanent storage devices like CDs, DVDs, etc. |
| Speed | Faster | Slower |
| Stored data | It saves the data that the computer is currently using. | It can save various types of data in various formats and huge sizes. |
|  |  |  |