

CS & IT ENGINEERING

Computer Organization and Architecture

Secondary Memory & IO Interface

Lecture No. - 07

By- Vijay Agarwal Sir



Recap of Previous Lecture



Flag

Shift & Rotate Operation

Interrupt Cycle

Amdhal's Law & Flynn Classification

System Bus

IO Interface & DMA

Topics to be Covered



Topic

IO Interface & DMA



I/O Interface

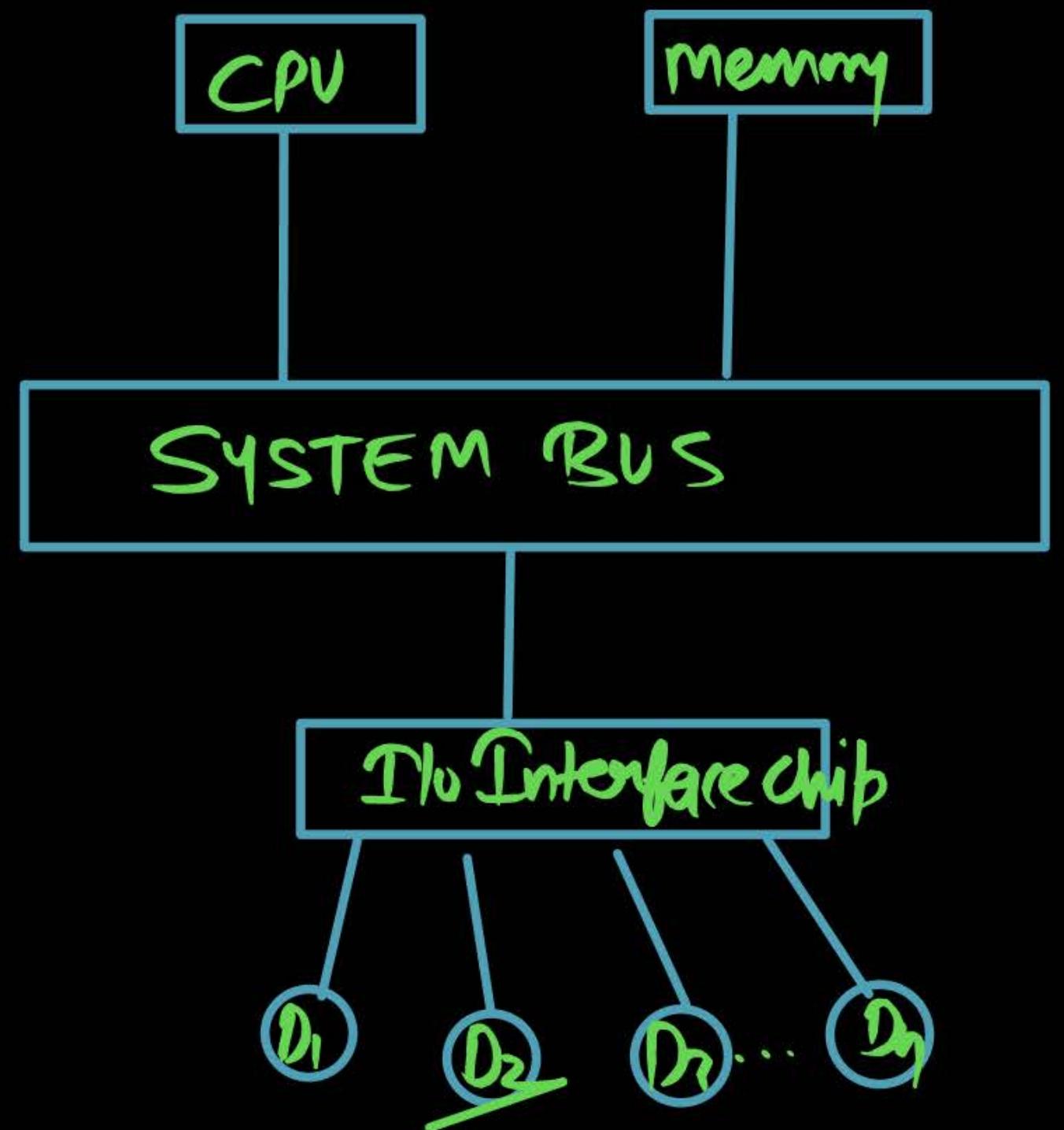
CPU is Electronic Component & I/O Device is Electro magnetic Grps.
(Fastest) (Slowest)

- In term of Speed
- operating Mode
- format
- Data transfer Rate.



I/O ORGANIZATION

- (1) I/O devices are electro-magnetic components and CPU is a electronic component. So, there is a difference exist in term of operating modes, data transfer rate and word formats.
- (2) To synchronize the I/O speed with a CPU, high speed interface chip is used named as I/O interface or I/O module.
- (3) I/O interface chip is responsible for I/O Operations so, in the computer design I/O devices are connected to system bus via I/O interface Chip.: .





Input Output Interface



Input-output interface provides a method for transferring information between internal storage and external I/O devices. Peripherals connected to a computer need special communication links for interfacing them with the central processing unit. The purpose of the communication link is to resolve the differences that exist between the central computer and each peripheral. The major differences are:

- 1 Peripherals are electromechanical and electromagnetic devices and their manner of operation is different from the operation of the CPU and memory, which are electronic devices. Therefore, a conversion of signal values may be required.
- 2 The data transfer rate of peripherals is usually slower than the transfer rate of the CPU, and consequently, a synchronization mechanism may be needed.

- ③ Data codes and formats in peripheral differ from the word format in the CPU and memory.
- ④ The operating modes of peripherals are different from each other and each must be controlled so as not to disturb the operation of other peripherals connected to the CPU.

Modes of Transfer

1. Programmed I/O
2. Interrupt-initiated I/O
3. Direct memory access (DMA)



Benefit of I/O Interface



I/O ORGANIZATION

System without I/O – Interface [Programmed- I/O]

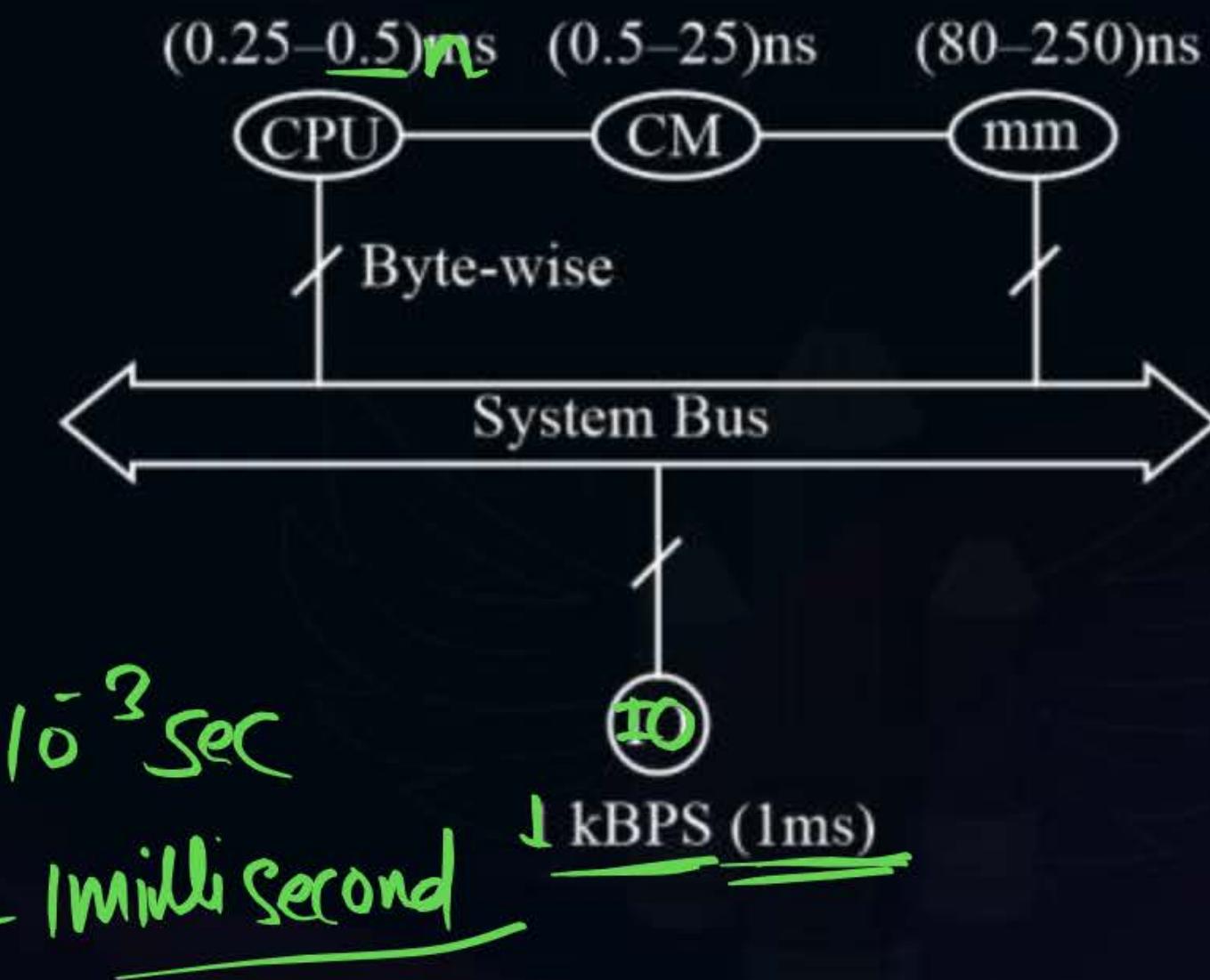
1 kBps

1 kB - 1 Sec
1 B - ?

$$ETIO = \frac{1B}{1kB} \text{ sec} = 10^{-3} \text{ sec} = 1 \text{ millisec}$$

1 kB — 1 Sec

$$1 \text{ Byte} = \frac{1 \text{ sec}}{1k} = \frac{1}{10^3} \text{ sec} = 10^{-3} \text{ sec} = 1 \text{ millisecond}$$

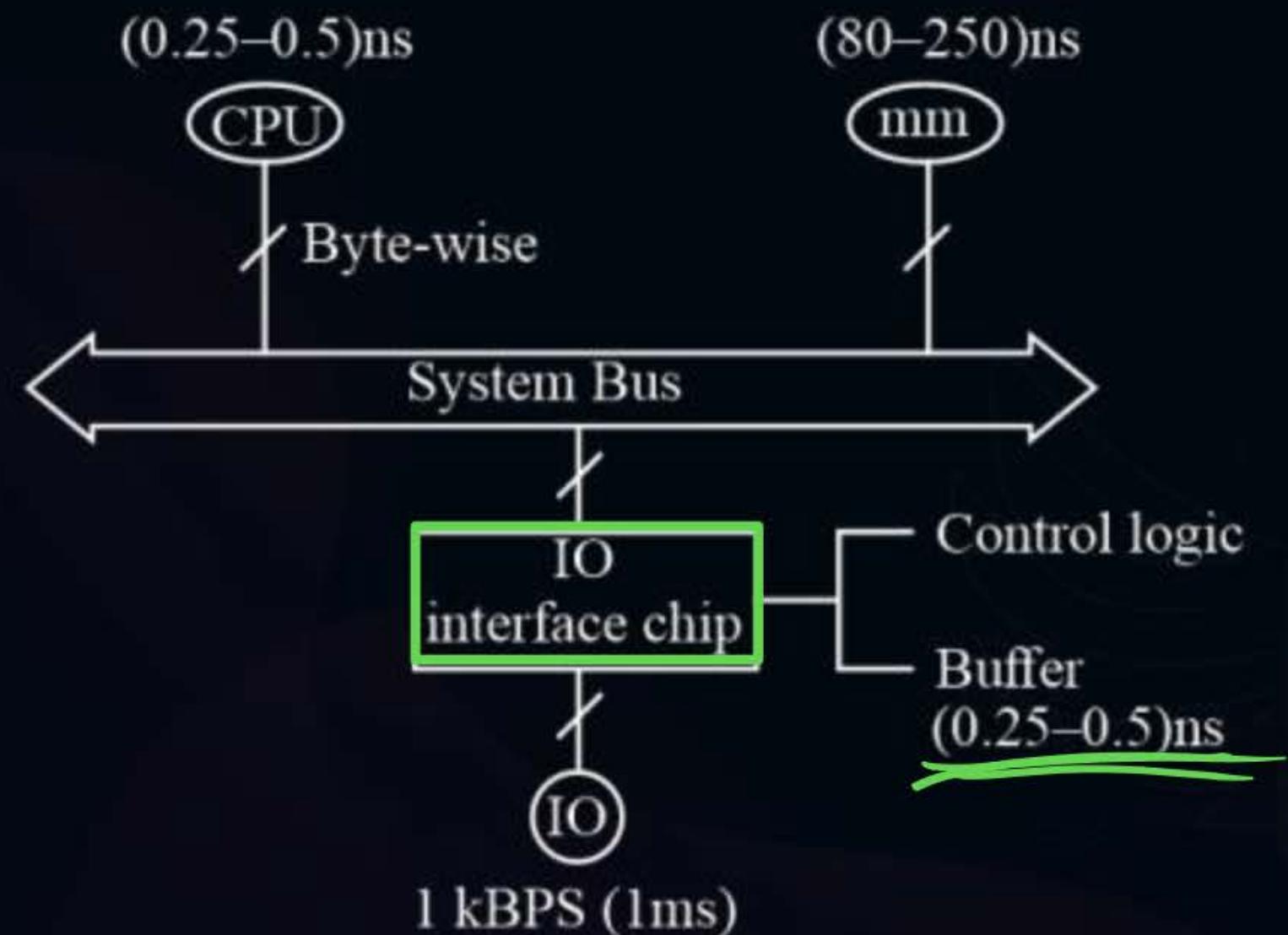




I/O ORGANIZATION



System with IO – Interface Chip [INT- Driven IO]





Working of I/O Interface



ACCESS SEQUENCE/Working Process

- (1) CPU initializes the I/O interface chip along with a I/O command(Operations), & then CPU will go & performing other useful task.
- (2) IO - interface control logic interprets the IO - Commands and Accordingly IO port will be enables for the IO operation.
- (3) Based on the speed of a IO device, & Amount of data to be transfer Consume the time to prepare the data(preparation time), then data is transferred from IO device to a interface buffer.
- (4) When the Data is available in the buffer IO interface generates the interrupt signals & send to the CPU and waiting for ack. Signal.



ACCESS SEQUENCE

- (4) After receiving the ack. Signal, buffer content will be transferred to CPU. In this process, CPU will be accessing the IO - data from interface buffer therefore speed gap is synchronized & Time saved.(Bcz IO interface fast).

Different IO – interface chip used in the computer design is.

- (1) 8255 PPI.
- (2) 8251 USART
- (3) 8259 A INT. Controller
- (4) 8237/8257 DMA etc.



IO - MODES

Three types of IO – transfer modes are present in the computer system , Used to transfer the data from the IO to other Component of a Computer. [CPU, memory]

Named as—

- (1) Programmed – IO .
- (2) Interrupt Driven IO.
- (3) DMA (Direct Memory Access)

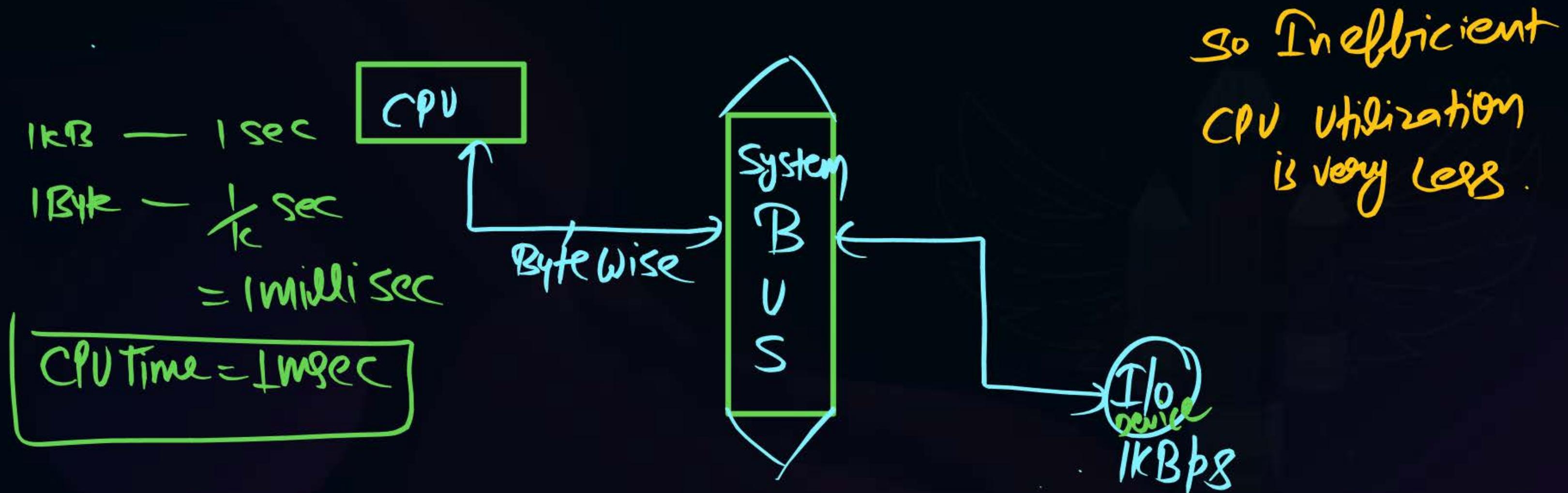
Modes of Transfer

1. Programmed I/O
2. Interrupt-initiated I/O
3. Direct memory access (DMA)



PROGRAMMED IO

- No High Speed IO Interface chip is used b/w DD Device & other Component of the Computer.





PROGRAMMED IO

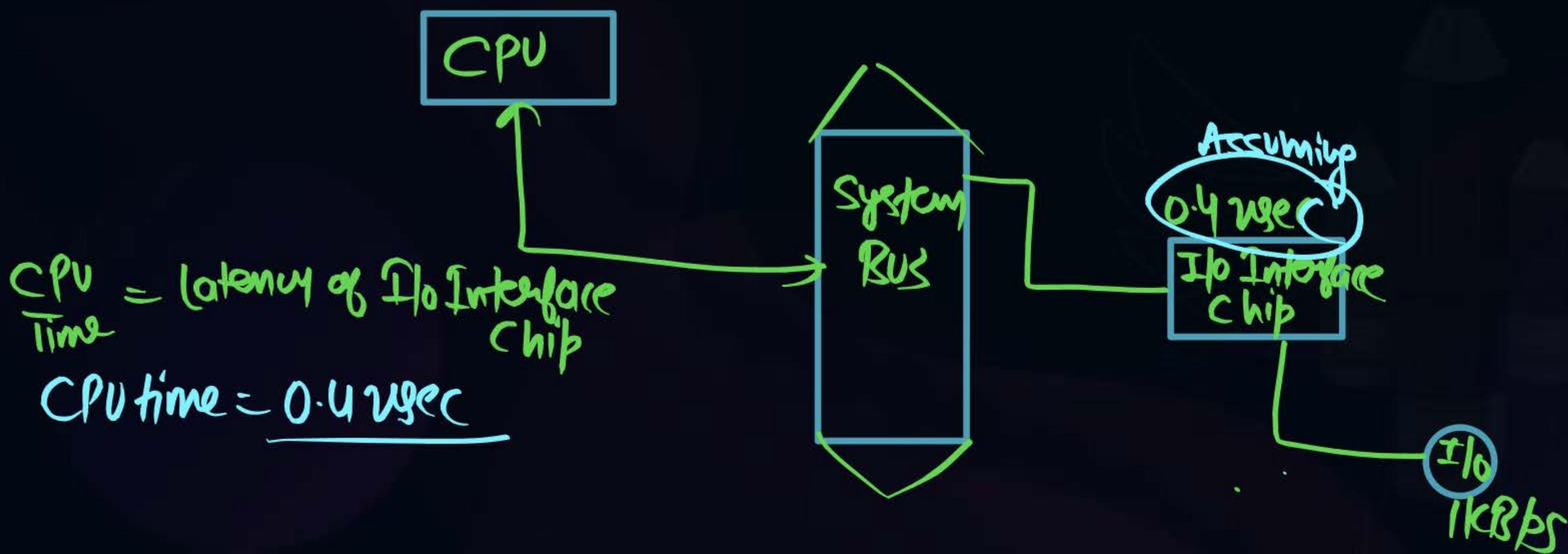


- (1) In this mode, IO- devices are directly Connected to CPU without IO- interface chip.
- (2) In this mode, CPU takes the responsibility to complete the IO operation, So CPU will be blocked [waiting] until the IO – operation is completed.
- (3) In this mode CPU Utilization is inefficient.
- (4) In this mode CPU time depends on the speed of a IO – device and the size of a data unit to be transferred.
- (5) This mode is suitable in the system centric application where the IO time is important than CPU time.



INTERRUPT DRIVEN IO

In Interrupt Driven I/O, High Speed Interface Chip is used between the I/O Devices & other Component of the Computer.





INTERRUPT DRIVEN IO



- (1) In this mode, IO- operation are controlled based on the interrupt signals.
- (2) In this mode, IO- devices are connected to a system bus via IO – interface chip
So, IO – interface takes the responsibility of a IO operation
- (3) In this mode processor utilization is efficient, so CPU executing the other useful task during the IO Operation. But in Programmed IO CPU is blocked.
- (4) In this mode CPU time is depends on the Latency of a interface chip rather than the speed of a IO device.

① Programmed I/O : No High speed Interface Chip is used.

② Interrupt I/O

- High speed Interface chip used
- CPU time Depends on Latency of I/O Interface
- CPU Utilization is efficient.

CPU takes Responsibility of I/O operation.

CPU Utilization Inefficient.

A device with data transfer rate 10 KB/sec is connected to a CPU. Data is transferred byte-wise. Let the interrupt overhead be 4 usec. The byte transfer time between the device interface register and CPU or memory is negligible. What is the minimum performance gain of operating the device under interrupt mode over operating it under program controlled mode?

(a) 15

~~(b) 25~~

(c) 35

(d) 45

[GATE-2005 : 2 Marks]

Performance $\propto \frac{1}{ET}$

10KBps

Interrupt (Interface) latency = 4 usec

Data transfer is Byte Wise.

$$\text{Performance} = \frac{\text{Perf. of Int.}}{\text{Gain}} = \frac{\text{Perf. of Prog Flo.}}{\text{Perf. of Int Flo.}}$$

$$= \frac{ET_{Prog\ Flo}}{ET_{Int\ Flo}} = \frac{ET_{Prog\ Flo}}{ET_{Prog\ Flo} + ET_{Int\ Flo}}$$

To Device

10 KBPS

10KB —— 1sec

$$1 \text{ Byte} = \frac{1}{10k} = 10^{-4} \text{ sec} = 100 \times 10^{-6} \text{ sec}$$

$$\begin{aligned} ET_{\text{Programmed Io}} &= 100 \mu\text{sec} \\ ET_{(\text{Interrupt Io})} &= 4 \mu\text{sec} \end{aligned}$$

$$\frac{\text{Performance Gain}}{Gain} = \frac{100}{4} = \cancel{25}$$



INTERRUPT DRIVEN IO



Drawback : More Complexity in interrupt implementation.

Solution : Use priority Serve Highest priority interrupt.

Who will Assign :

1. Daisy Chain Method(Static Approach) : Fixed : Starvation occur.
2. Polling Method(Dynamic Approach) : Changing : No Starvation.

- ① Programmed I/O
- ② Interrupt I/O.
- ③ DMA (Direct Memory Access).

Direct memory Access [DMA]

In DMA:

Bulk Amount of Data is transferred from I/O to Memory without the Involvement of CPU.

• CPU has the Highest Priority.

In Programmed I/O & Interrupt Driven I/O } Small Amount of Data transfer &
Can not Access Memory Directly ie we have to go
With the Help of (Via) CPU.



DMA (DIRECT MEMORY ACCESS)

(1) In this mode , bulk amount of the data will be transferred from the IO to main memory without involvement of a CPU.

Here processor & DMA controller use the system bus in Master-Slave mode.

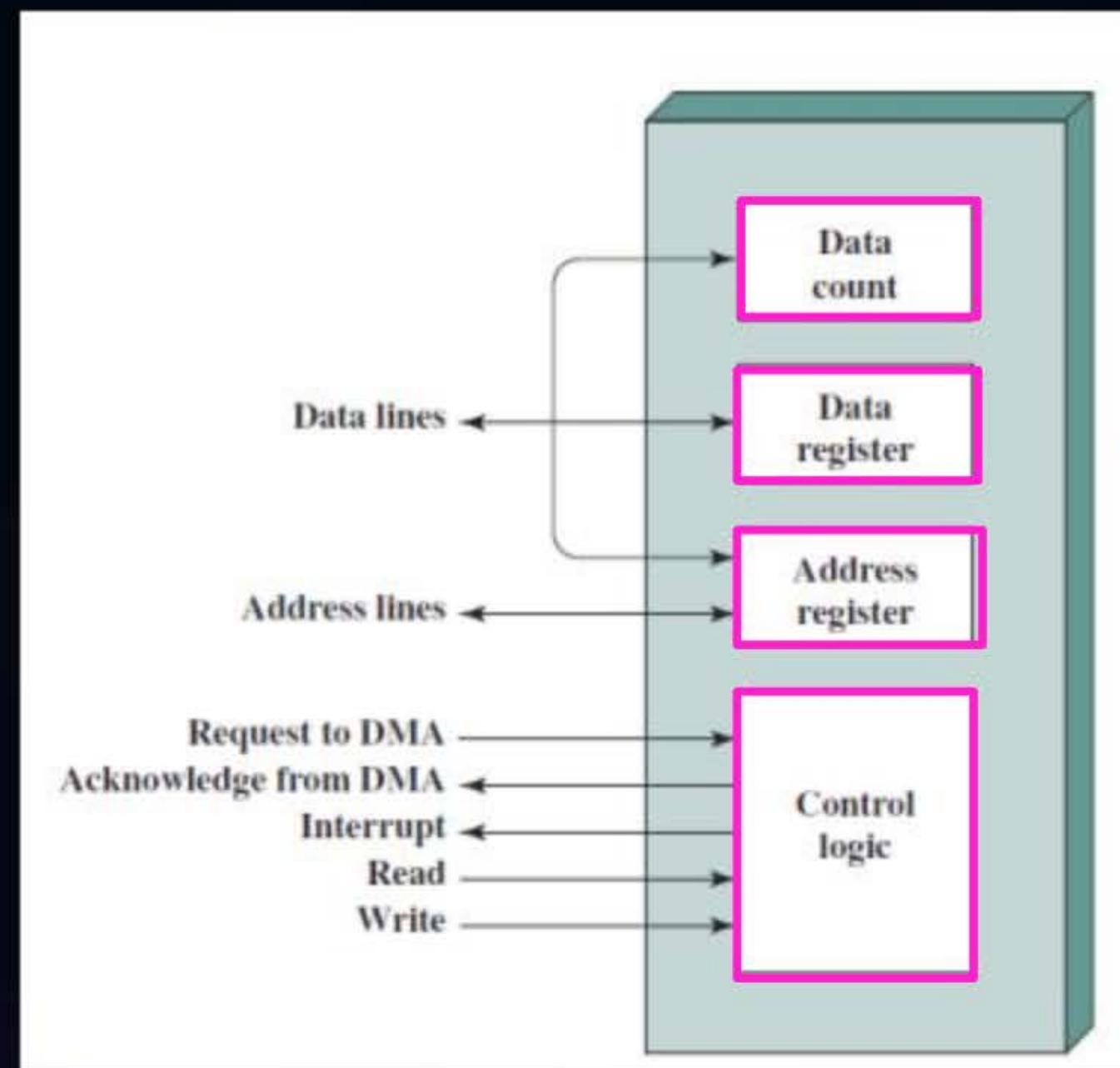
(2) When the user program size is greater than the main memory size than virtual memory concept is used to increase the address space.

(3) Virtual memory concept state that use the secondary memory to store the user program.

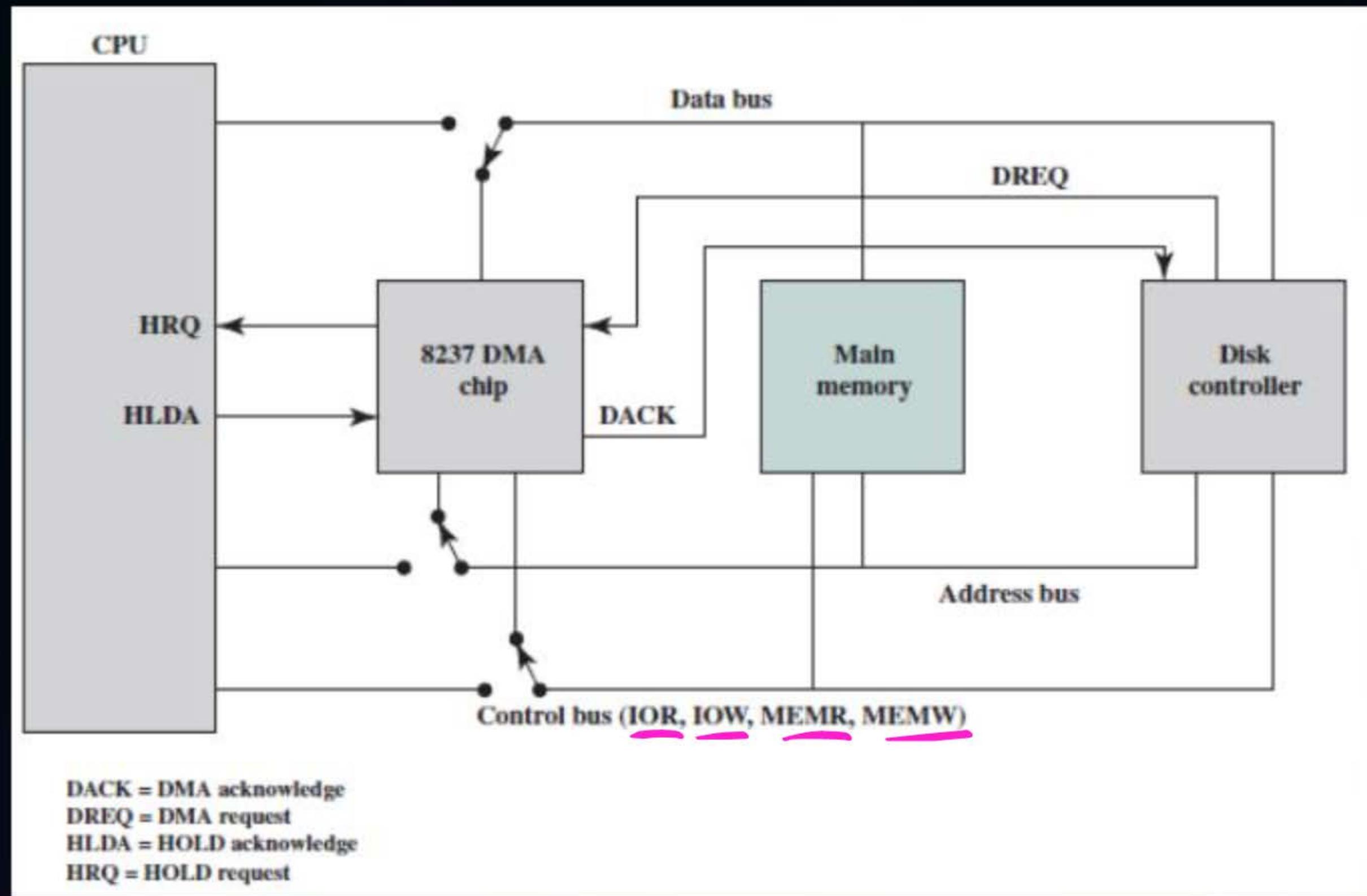
(4) Secondary memory is a kind of I/O device, which is interfaced to a DMA module, therefore during the program execution data will be transferred from I/O to main memory via DMA without involvement of a CPU.

DMA

- Direct Memory Access
- SYSTEM Bus in Master Slave Mode
- Bulk Amount of Data transfer .
- Virtual Memory Concept is Used
- Highest Priority .



Typical DMA Block Diagram



DACK = DMA acknowledge

DREQ = DMA request

HLDA = HOLD acknowledge

HRQ = HOLD request

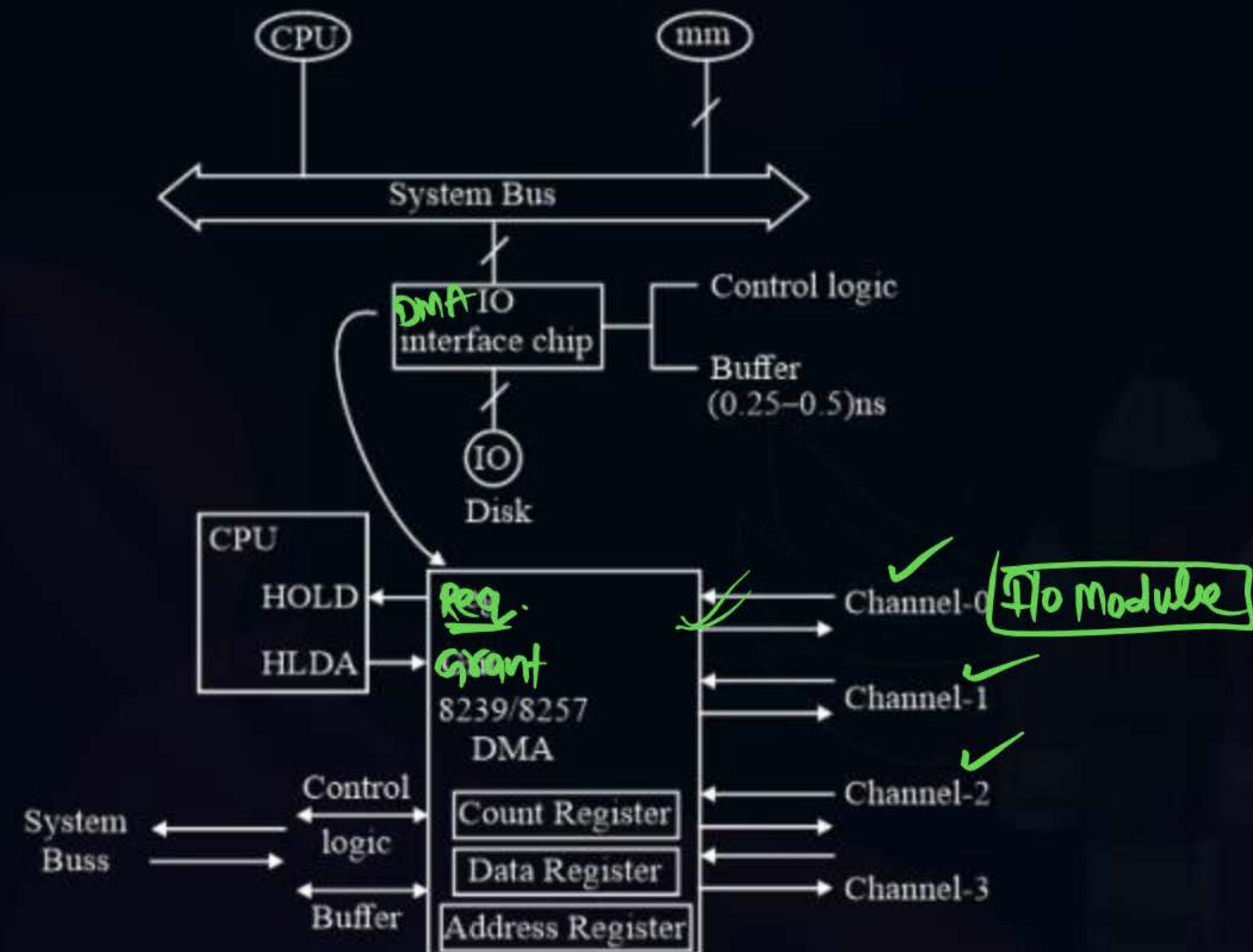


DMA Working.



DMA (DIRECT MEMORY ACCESS)

① CPU DMA I/O





ACCESS SEQUENCE

- ✓ (1) CPU initializes the DMA module along with a I/O command later busy with other useful task.
- ✓ (2) I/O command contain the information about port address, memory address, control signal and count value.
- ✓ (3) DMA module control logic interprets the command and enables the respective port for the operation.
- ✓ (4) Based on the speed of a device consumes the time to prepares the data Later enables the “DMA REQ” signal.



ACCESS SEQUENCE

- ✓(5) After receiving this signal, DMA module enables the Hold signal to CPU, to gain the control of a system Bus and waiting for HLDA signal. *from the CPU*
- (6) After receiving the HLDA signal ; DMA module enables the “DMAACK” signal.
- (7) After receiving this signal, IO device transfer the data to main memory via DMA to main memory via DMA until Count becomes “0”
- (8) After the DMA operation Bus Connection will be re-established to CPU.



Data Count



Data Count : How Many number of Bytes/words transfer from I/O to Memory.
Until the count value become '0'.



ACCESS SEQUENCE

NOTE : In the DMA Operation , CPU is in Two States—

- (1) Busy State : CPU is in Busy state Until prepares the data.so Busy state depends on preparation time. preparation time depends on Speed of I/O speed and Amount of Data(data Size).
- (2) Blocked (Hold) state :CPU is in Block state Until transferring the data.so Block time depends on transfer time. Transfer time depends on MM latency(MM Access time). [depends on mm speed]

Let
X is a Preparation time
Y is a transfer time

∴

$$\% \text{ time CPU Busy} = \left(\frac{X}{X+Y} \right) 100$$

$$\% \text{ time CPU Blocked} = \left(\frac{Y}{X+Y} \right) 100$$



ACCESS SEQUENCE



DMA module is operating in three mode :

- (1) Burst mode :
- (2) Cycle stealing mode(Single Cycle mode) :
- (3) Interleaving mode.



ACCESS SEQUENCE

(1) Burst mode: In the Burst Mode, the DMA Controller keeps control of the bus until all the data has been transferred to memory from the peripheral devices.

This mode of Transfer is needed for Fast Devices where data transfer can not be stopped until the entire transfer is done.

DMA having the Control on the Bus Until all Data transfer



ACCESS SEQUENCE

(2) **Cycle stealing mode** : In the Cycle stealing Mode , the DMA Controller relinquishes the bus after each transfer of one word. This minimize the amount of time that the DMA Controller keeps the CPU from controlling the bus, but it requires that the bus request/acknowledge sequence be performed for every single transfer.



ACCESS SEQUENCE

(3) Interleaving mode:

CPV is Busy in a Work which Not Require the Use of Buses that time DMA takes the Control of the Bus & perform the operation.

NAT

The size of the data count register of a DMA controller is 16 bits. The processor needs to transfer a file of 29, 154 kilobytes from disk to main memory. The memory is byte addressable. The minimum number of times the DMA Controller needs to get the control of the system bus from the processor to transfer the file from the disk to main memory is 456 Ans

[GATE-2016(Set-1)-CS: 2M]

~~DMA Count Register = 16 bit~~

Total Count Value = $2^6 - 1$

In One Time, Total # Byte Transferred = $2^{16} - 1 = 64KB$

Ans (456)

Total # Data transferred = 29, 154 kB.

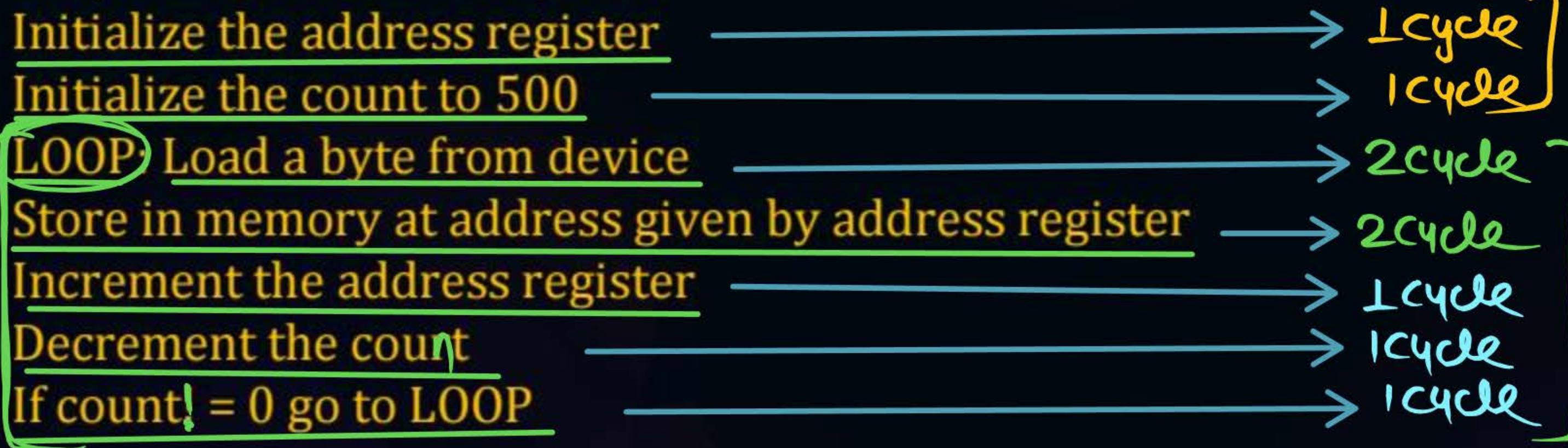
Time Need the Control
On The System Bus = $\left\lceil \frac{29,154KB}{64KB} \right\rceil = 456 Ans$

$$\text{Count} = 16 \text{ bit} = 2^{16}$$
$$\text{Total Data in One Cell} = 2^{16} \approx 64 \text{ KB}$$

$$\text{Total Data} = 29,154 \text{ KB}$$

MCQ

On a non-pipelined sequential processor, a program segment, which is a part of the interrupt service routine, is given to transfer ~~500 bytes~~ from an I/O device to memory.



Assume that each statement in this program is equivalent to a machine instruction which takes one clock cycle to execute if it is a non-load/store instruction. The load-store instructions take two clock cycles to execute.

The designer of the system also has an alternate approach of using the DMA controller to implement the same transfer. The DM controller requires 20 clock cycles for initialization and other overhead. Each DMA transfer cycle takes two clock cycles to transfer one byte of data from interrupt driven program-based input-output?

[GATE-2011-CS: 2M]

A 3.4

Ans (3.4)

B 4.4

C 5.1

D 6.7

(In each loop)

In Each Iteration takes = $2+2+1+1+1 = 7$ cycle per Iteration.

Loop will execute = 500 Time.

Total Time in 500 Iteration = $500 \times 7 = 3500$ cycle.

Total Time taken in Interrupt = $3500 + 2 = 3502$ cycle.

Using DMA transfer the 500 Byte.

(initialization)

In DMA, Time taken to transfer 500 Byte = $20 + 2 \times 500 = 1020$ cycle

$$\frac{3502}{1020} = 3.4 \text{ Avg}$$

∴



THANK - YOU