CS & IT ENGINEERING

COMPUTER ORGANIZATION AND ARCHITECTURE

IO Organization

Lecture No.- 03

By- Vishvadeep Gothi sir

Recap of Previous Lecture

- **Topic** Asynchronous Data Transfer
- **Topic** Modes of Transfer
- **Topic** Programmed IO
- **Topic** Interrupt IO

Topics to be Covered

- **Topic** Interrupt Mode
- **Topic** DMA
- **Topic** Modes of DMA

Topic : Modes of Transfer

1. Programmed I/O
2. Interrupt I/O
3. DMA

Topic : Programmed IO

- There is no any provision through which IO can inform to CPU about data transfer
- IO sets its own status and waits
- CPU runs program periodically and checks the status of each device one-by-one
- If any device has its status set then CPU performs data transfer for it.

Topic : Interrupt Initiated IO

- IO device has a provision (Interrupt Signal) to inform to CPU about communication.

Topic : Interrupt Initiated IO

- IO device has a provision (Interrupt Signal) to inform to CPU about communication.
- When CPU receives interrupt:
  - It completes execution of current instruction
  - Saves the status (PC, PSW etc.) of current process onto the stack
  - Branches to service the interrupt
  - Resumes the previous process by taking out the values from stack

Topic : Vectored vs Non-Vectored

Interrupt

Topic : Maskable vs Non-Maskable

Topic : Internal Vs External

↓

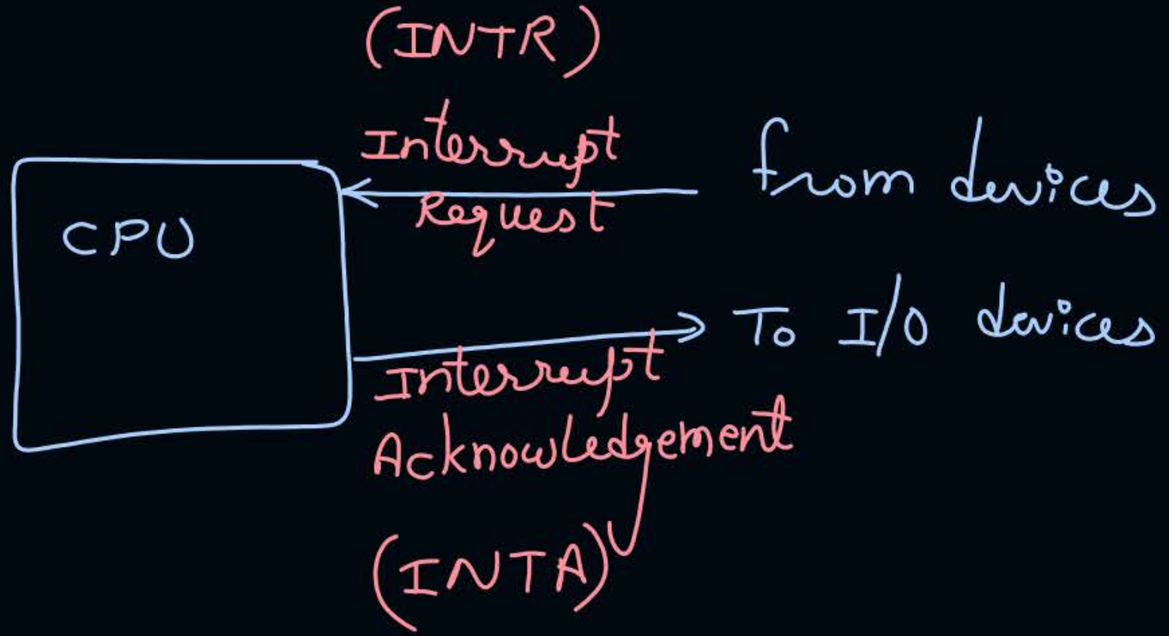
→ from devices

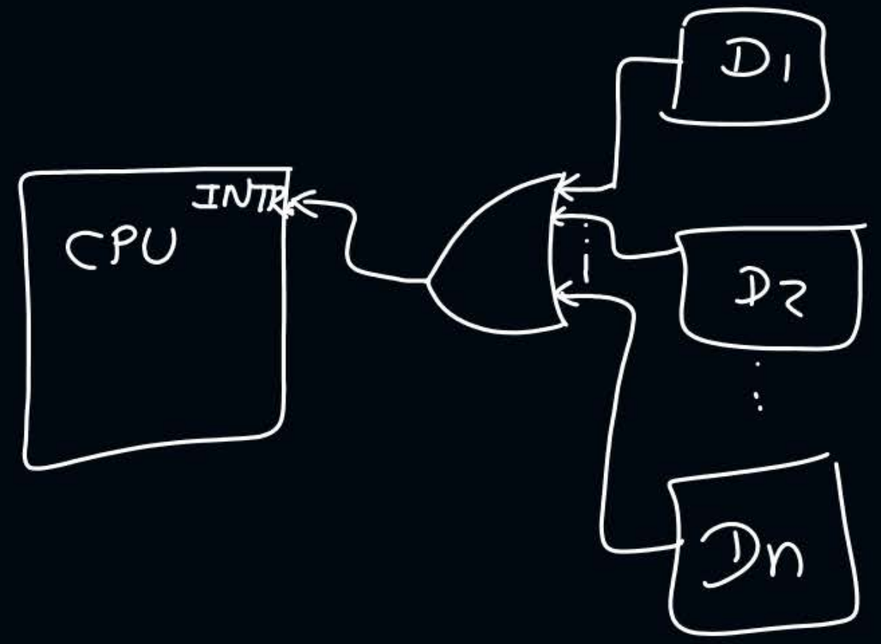
Topic : Simultaneous Interrupts

If multiple devices generate interrupt simultaneously, then interrupt from higher priority device will be serviced first.

Priority based interrupt handling

- S/w soln
- H/w soln ← Priority of devices are fixed.
  - Serial (daisy chaining)
  - Parallel





with single INTR line ⇒ multiple simultaneous interrupts possible

→ Device generates interrupt

→ CPU accepts it by sending Ack.

→ Device sends vector through data bus

Topic : Time Required in Interrupt IO

= Interrupt overhead time + Interrupt service time

↓

time from CPU receives interrupt till before CPU starts service

[NAT]

Ans = 0.65

#Q. Consider a CPU which takes 0.05 microseconds as interrupt overhead time when a device generates interrupt for CPU, and CPU accepts it. After that CPU takes 6 cycles to service the interrupt. If CPU runs on 10MHz clock rate then total time CPU spends for interrupt service is _____ microseconds?

$= 0.05\ \mu sec + 6\ cycles$

$= 0.05\ \mu sec + 6 * \frac{1}{10MHz}$

$= 0.05\ \mu sec + 0.6\ \mu sec$

$= 0.65\ \mu sec$

[NAT]

#Q. A device with data transfer rate 20 KB/sec is connected to a CPU. Data is transferred byte-wise. Let the interrupt overhead be 10 microsecond.

1. Total time required in programmed IO for 10 bytes data transfer? ⇒550 μs
2. Total time required in interrupt IO for 10 bytes data transfer? ⇒510 μs
3. What is the minimum performance gain of operating the device under interrupt mode over operating it under program controlled mode?

1. Programmed I/O time = status check time + Data transfer time

↳ 1 Byte time from I/O + 10 byte transfer time

$= 50\ \mu sec + 10 * 50\ \mu sec$

$= 550\ \mu sec$

---

20 kB, transfer time = 1 sec

1 B —"— $= \frac{1\ sec}{20\ kB} * 1B$

$= \frac{1}{20}\ msec$

$= \frac{1000}{20}\ \mu sec = 50\ \mu sec$

2.) Interrupt I/O time = interrupt overhead time + service time

= 10 μsec + 10 * 50 μs

= 510 μs

---

3. performance gain = $\frac{\text{older technique time}}{\text{newer technique time}} = \frac{\text{Programmed I/O time}}{\text{Interrupt I/O time}}$

or

speed up

= $\frac{550\ \mu s}{510\ \mu s}$

= 1.078

[NAT] GATE- PYQ

#Q. 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 microsecond. 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?

$$\text{Performance gain} = \frac{\text{Prog. controlled mode time}}{\text{Interrupt I/O time}} = \frac{\text{status check time} + \text{Data transfer time}}{\text{Interrupt overhead} + \text{Data transfer time}}$$

$$= \frac{100\,\mu s + 0}{4\,\mu sec + 0} = \frac{100\,\mu s}{4\,\mu s} = \underline{\underline{25}}$$

Ans.

for 10 kB, time = 1 sec

1 B, —11— = $\frac{1 \text{ sec}}{10 \text{ kB}} * 1\text{B}$

= 0.1 msec

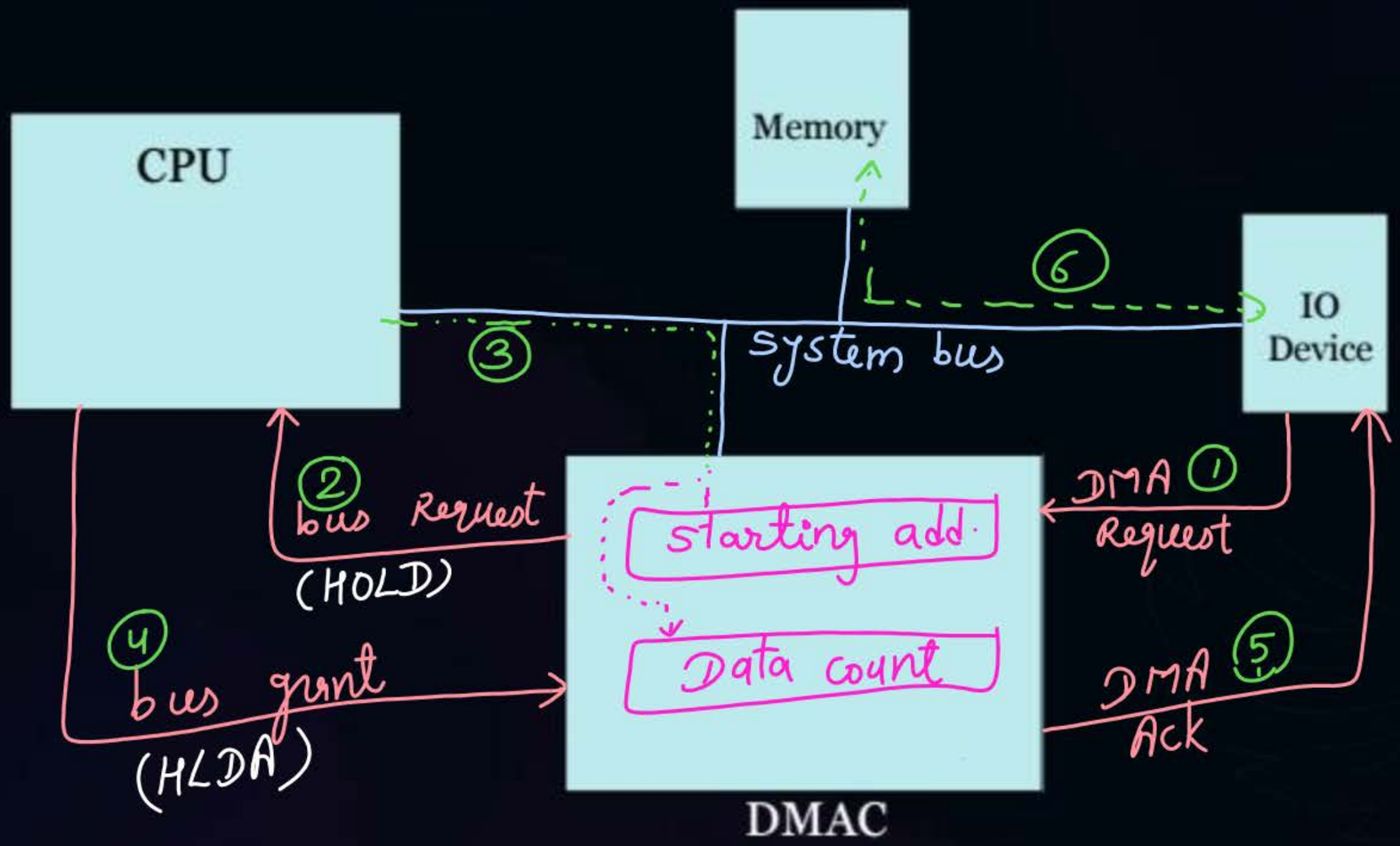
= 100 μsec

Topic : DMA (Direct Memory Access)

- Enables data transfer between I/O and memory without CPU intervention
- Need a hardware: DMAC
  - ↳ DMA Controller

→ DMAC is a special purpose processor which performs data transfer b/w mem. & I/O.
  - ↳ It can generate add. & control signals

Topic : DMA


Memory
CPU
IO Device
6
system bus
3
2
bus Request
(HOLD)
DMA
Request
1
starting add.
Data count
4
bus grant
(HLDA)
DMA
Ack
5
DMAC

Topic : DMA

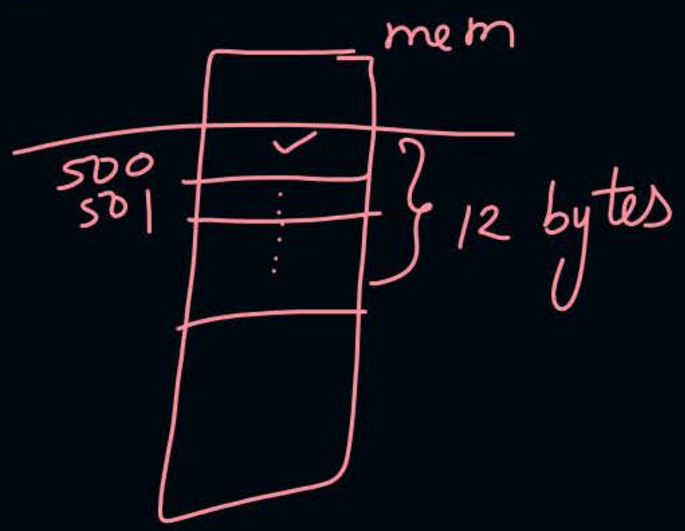
1. Starting Address :- add. of mem. starting from where the data transfer starts.

2. Data Count :- count of data (bytes or words) to be transferred.

- bytes → ✓ byte addressable memory
- words → word addressable memory

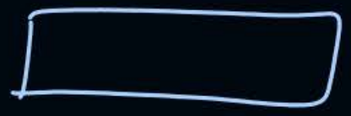
ex:-

	Initially	After 1B transferred	After 1 more byte		
starting add.	500	501	502	. . —	
Data count (in bytes)	12	11	10	. . . .	0



Data transfer with DMA stops when data count becomes zero.

ex:-

if data count Reg. 

and value $= (1001)_2$ $\Rightarrow$ no. of bytes transferred = 9

$\Downarrow$

$(9)_{10}$

---

if data count reg. $\Rightarrow$ 

$\Rightarrow$ max value $= (1111)_2 = (15)_{10}$

max no. of bytes DMA can transfer at time $\Rightarrow$ 15 bytes

⇒ while DMA transfer, CPU can perform only those operations which do not require buses.

⇓

mostly CPU will be blocked

2 mins Summary

Topic Interrupt Mode

Topic DMA

Topic Modes of DMA

Happy Learning

THANK - YOU