COMPUTER SCIENCE

Computer Organization and Architecture

Secondary Memory & IO Interface

Lecture_05

Vijay Agarwal sir

01 IO Organization

1. Disk

- Disk Structure
- Disk Capacity
- Disk Access time (S.T. Avg R.L. D.T.T & Data transfer Rate)
- Disk Addressing <C. h. S>
  - C → Cylinder
  - h → Surface
  - S → Sector.

II. I/O Interface

WHAT I/O Interface ?

WHY I/O Interface Used ?

WHEN I/O Interface Used ?

How I/O Interface ?

Type of Mode

1. Programmed I/o
2. Interrupt Driven I/o
3. DMA [Direct Memory Access]

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:

- 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.
- The data transfer rate of peripherals is usually slower than the transfer rate of the CPU, and consequently, a synchronization mechanism may be needed.

- Data codes and formats in peripheral differ from the word format in the CPU and memory.
- 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)

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.:

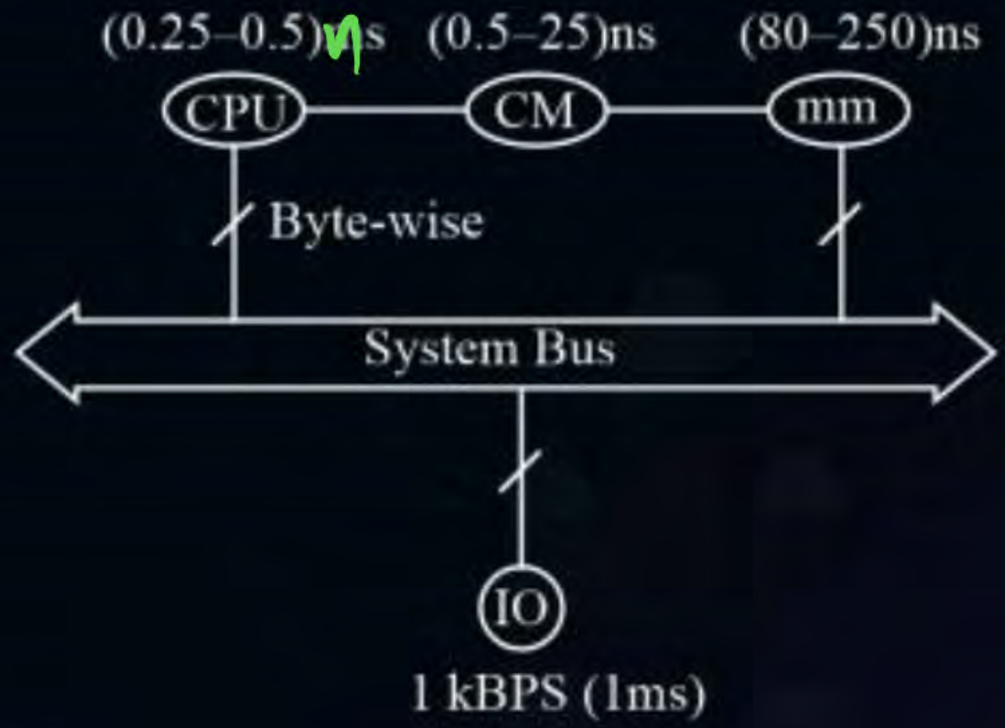
I/O ORGANIZATION

System without IO – Interface [Programmed- IO]

1 kB – 1Sec

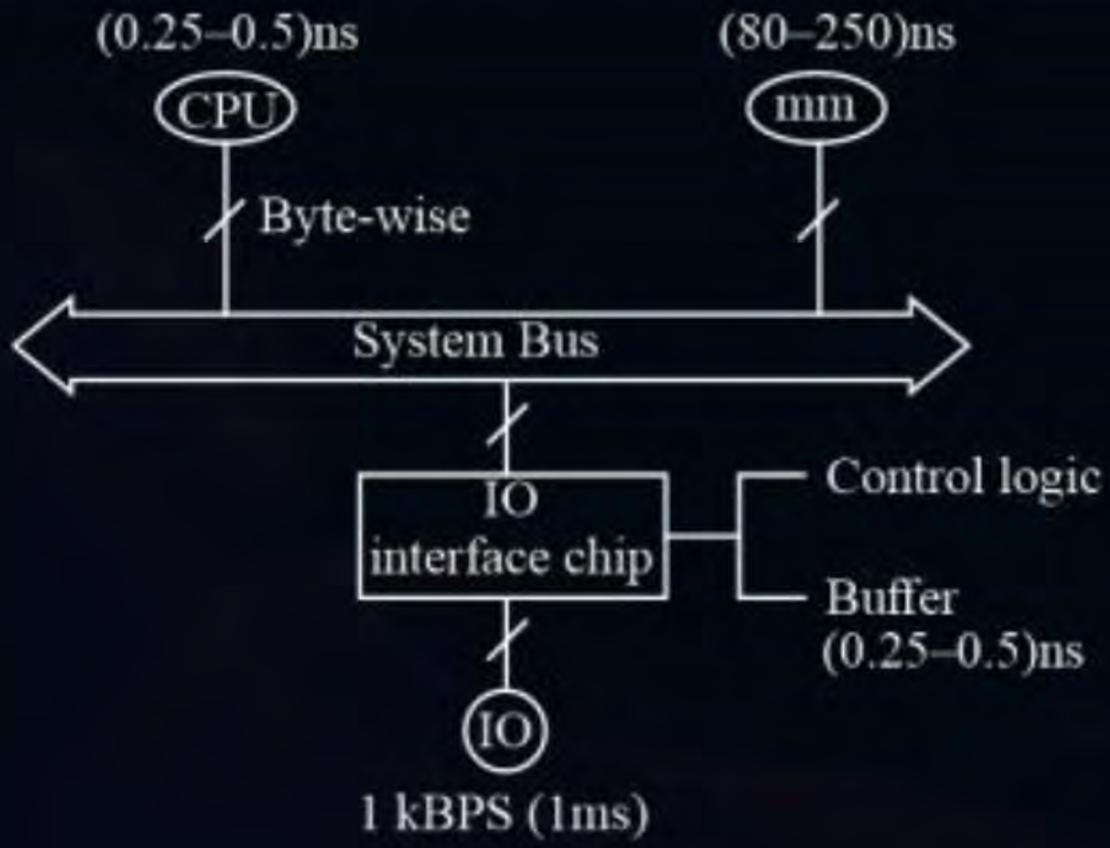
1 B – 2

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



I/O ORGANIZATION

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



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 Interface Logic is used b/w I/o Device & other Component of the Computer.

- CPU Time Depends on Speed of I/o Devices.

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 this High Speed I/O Interface chip(logic) is Used b/w I/O Devices & other Component of the Computer.

CPU Time = I/O Interface chip Latency

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 open.

(3) In this mode processor utilization is efficient, so CPU executing the other useful task during the IO Open.

(4) In this mode CPU time is depends on the Latency of a interface chip rather than the speed of a IO device.

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]

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

Programmed I/O :

10 KB/s

10 KB ——— 1 sec

1 Byte ——— $\frac{1}{10 \times 10^3}$ sec $\Rightarrow 10^{-4}$ sec $\times \frac{100}{100} \Rightarrow 100 \times 10^{-6}$ = 100 usec

$ET_{Prog\ I/O} = 100$ usec

$ET_{interrupt} = 4$ usec

$S = \frac{\text{Performance of Interrupt}}{\text{Performance of Prog I/O}}$

$\Rightarrow \frac{ET_{Prog\ I/O}}{ET_{interrupt\ I/O}}$

$= \frac{100}{4} = 25$

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 Driven I/O

- In these two we can not Access the memory Directly.

(we have to Access (go to) memory With the Involvement of CPU [Via CPU]

- These are Used for Small Amount of Data transfer.

③ DMA [Direct Memory Access]

- Highest Priority
- Cycle Stealing Mode.

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 a 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.

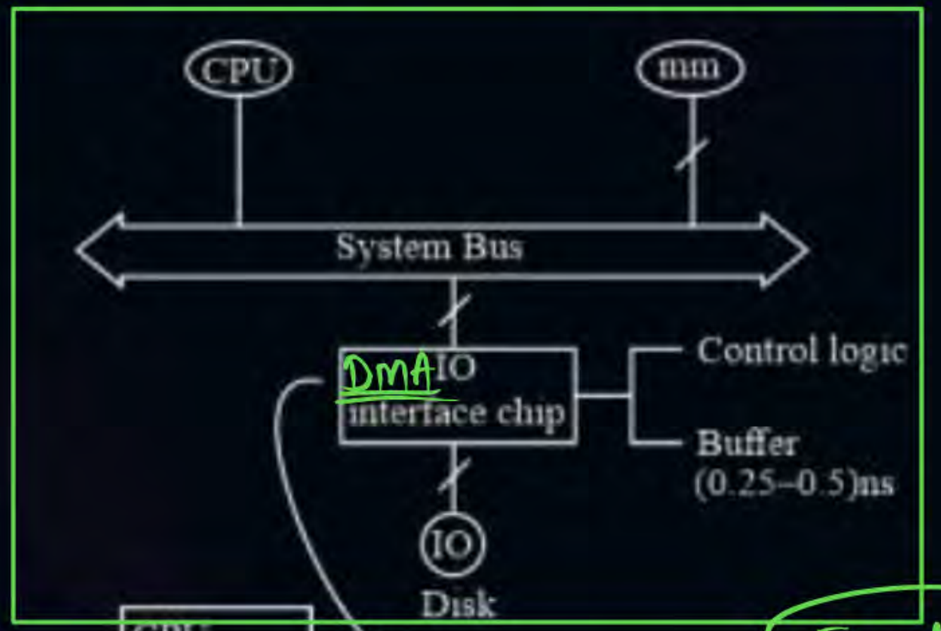
DMA

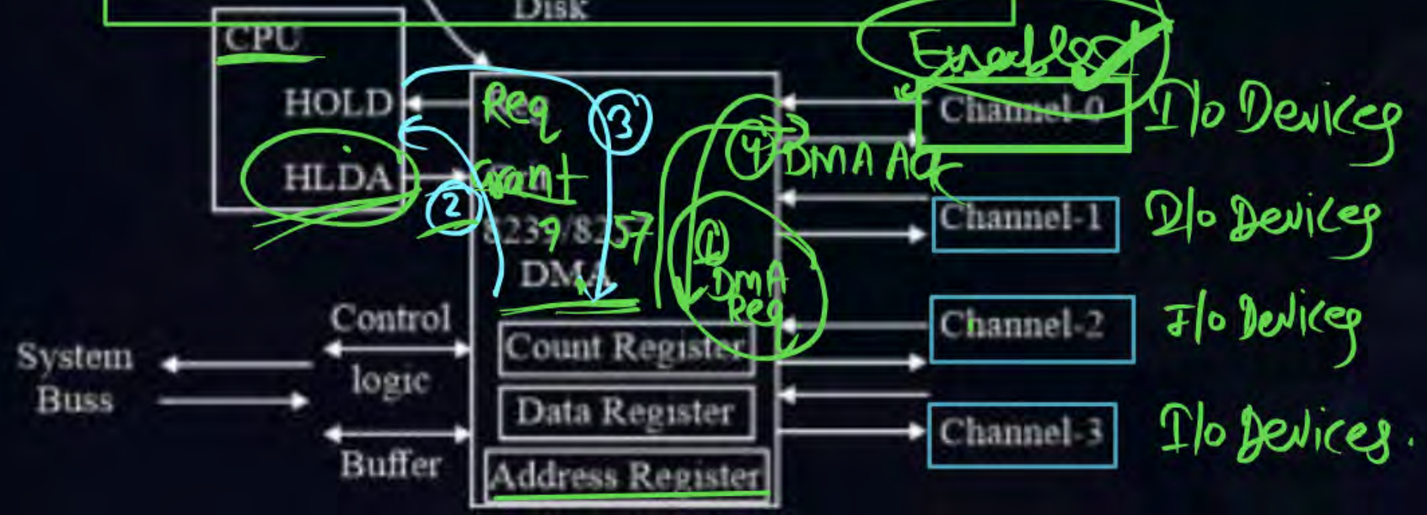
: ① Bulk Amount of Data transfer.
② Directly Memory Access
③ Without the Involvement of CPU.
④ Highest Poority.
⑤ Processor [CPU] & DMA Use the System Bus in Master-Slave.
⑥ Virtual Memory Concept is used.

DMA Module Working.

DMA (DIRECT MEMORY ACCESS)

Count Register





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.

e.g. 1KBPS & 40KB then Prepare time:

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.

(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 ([illegible]) 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

$\therefore$

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

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

(X) : Preparation Time

(eg) I/o Device (Disk)

1KBps

& 40Byte

1KB ——— 1sec

40Byte $\Rightarrow$ $\frac{40B}{1KB}$ sec $\Rightarrow 40 \times 10^{-3} =$ 40msec

ACCESS SEQUENCE

DMA module is operating in three mode :

(1) Burst mode :

(2) Cycle stealing mode :

(3) Interleaving mode

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____.

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

Count Register = 16 bit

Data transfer in one cycle = $2^{16} - 1 = 64\text{KB}$

Size of Data to be transferred = 29,154 KByte

Number of time DMA Need the control of System Bus = $\left\lceil \frac{29154\text{KB}}{64\text{KB}} \right\rceil = \lceil 455.5 \rceil = 456$ Ans

NAT

Consider a disk with data transfer rate 50MBPS. It is operated with cycle stealing mode of DMA. Here whenever 64bits information is available it is transferred in 40ns. What is the percentage(%) of time CPU blocked due to DMA____.

Disk Speed = 50MBps

64bit Data transfer [8Byte]

Data transfer time = 40 nsec [y]

X: [Preperation time]

50MB ——— in 1 Sec

8Byte ——— $\frac{8\text{Byte}}{50\text{MB}}$ sec = 160 nsec

$= \frac{8}{50} \times 10^{-6}$ sec $= 0.16$ usec

% time CPU Block $= \frac{y}{x+y} \times 100$

$\Rightarrow \frac{40}{160+40} \times 100 = \frac{40}{200} \times 100$

$= 20\%$ Ans

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.

Initialize the address register → 1 cycle
Initialize the count to 500 → 1 cycle
Initialized

LOOP: Load a byte from device → 2 cycle
Store in memory at address given by address register → 2 cycle
Increment the address register → 1 cycle
Decrement the count → 1 cycle
If count! = 0 go to LOOP → 1 cycle

500 times
500 Byte

In a Loop: Each Iteration takes : $2+2+1+1+1 = 7$ cycle

Loop Execute : 500 times (500 Iteration)

Total time in Loop $= 7 \times 500 = 3500$

Total Time $= 2 + 3500 = 3502$ cycle)

---

500 Byte Using DMA

Total time $=$ Initialize $20 + 500$ (2 cycle)

$= 1020$ cycle

Speed up $= \frac{3502}{1020} = 3.4$

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]

Ans (A)

(A) 3.4

(B) 4.4

(C) 5.1

(D) 6.7

COA :

1. Introduction of COA.
2. M/c Instⁿ & AM.
3. Floating Point Representation
4. ALU Data Path, uprog & Control Unit
5. Pipelining (V. Imp)
6. Cache Memory (V. Imp)
7. Secondary Memory (Disk) & IO Interface

Till Now 132 PYQ

121+ GATE PYQ's.

along with Classroom Question + D.P.P + Weekly Test.

THANK YOU!