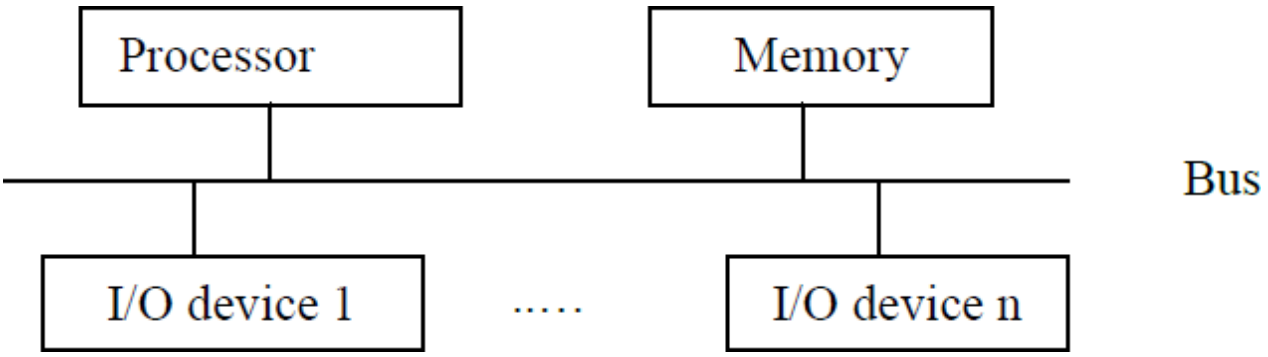
UNIT – III

INPUT/OUTPUT ORGANIZATION: Accessing I/O Devices, Interrupts, Interrupt Hardware, Enabling and Disabling Interrupts, Handling Multiple Devices, Controlling Device Requests, Exceptions, Use Of Interrupts in Operating Systems, Pentium Interrupt Structure, Direct Memory Access, Busses, Interface Circuits, Standard I/O Interfaces.

---

2 Marks

1. Give the organization of single bus structure?



2. What is memory mapped I/O?

With Memory mapped I/O, any machine instruction that can access memory can be used for transfer data to or from an I/O device.

3. What is program controlled I/O?

In program controlled I/O, the processor repeatedly checks a status flag to achieve the required synchronization between the processor and an input and output device.

4. What are the various mechanisms for implementing I/O operations?

1. Program controlled I/O
2. Interrupts
3. Memory mapped I/O
4. DMA

5. Define ISR

ISR is nothing but interrupt service routine. It can handle the execution of the interrupts and it responses to an interrupt request.

6. What constitute the device's interface circuit?

The address decoder, the data and status register and the control circuitry required to co-ordinate I/O transfers constitute the device‟s interface circuit.

7. What is interrupt service routine?

The routine executed in response to an interrupt request is called as interrupt service routine. In short , it is called as ISR.

8. What is the purpose of interrupt acknowledgement signal?

The interrupt acknowledgement signal is used by the processor to inform the device that is request has been recognized so that it may remove its interrupt request signal.

9. Define interrupt latency?

The delay between the time an interrupt request is received and the start of execution of the interrupt service routine is called interrupt latency.

10. What is real time processing?

The concept of interrupts is used in many control applications where processing of certain routines must be accurately timed relative to external events. This type of application is called as real time processing?

11. What are Special gates used for driving INTR line?

The special gates used for driving INTR line are,

1.Open collector

2.Open drain

12. When is an interrupt line said to be edge – triggered?

An interrupt line is said to be edge triggered if the interrupt handling circuit responds only to the leading edge of the signal.

13. Give a typical scenario assuming that interrupts are enabled?

- The device raises an interrupt request.
- The processor interrupts the program currently being executed.
- Interrupts are disabled by changing the control bits in the bus.
- The device is informed that its request has been recognized and in response, it activates the interrupt request signal.
- The action requested by the Interrupt is performed by the ISR.
- Interrupts are enabled and execution of the interrupted program is resumed.

14. What are vectored interrupts?

To reduce the time involved in the polling process, a device requesting an interrupt may identify itself directly to the processor. Then the processor can immediately start executing the corresponding ISR. The term vectored interrupts refer to all interrupt handling schemes based on this approach.

15. What is interrupt vector?

Interrupt vector is the starting address of the interrupt service routine stored in the location pointed by the interrupting device.

16. What are privileged instructions?

Privileged instructions are the instructions which are executed only while the processor is running in the supervisor mode.

17. What is privilege exception?

An attempt to execute a privileged instruction while in the user mode leads to a special type of interrupt called a privilege exception.

18. What are the two independent mechanisms for controlling interrupt requests?

To control interrupt requests ,the mechanisms used are ,

1. At the device end , an interrupt enable bit in a control register determines whether the device is allowed to generate an interrupt request.

2. At the processor end, either an interrupt enable bit in the PS register or a priority structure determines whether a given interrupt request will be accepted.

19. What are exceptions? Give an Example?

An except is a term often used to refer to any event that cause an interruption.

Eg. I/O interrupts.

20. What is a debugger?

A debugger is a program used by system software which helps the programmer finds errors in a program.

21. What are the two facilities provided by a debugger?

The facilities provided by a debugger are.

- 1.Trace
- 2.Breakpoints.

22. What does an exception occur when the processor is in trace mode?

When the processor is operating in the trace mode, an exception occurs after execution of every instruction, using the debugging program as the exception service routine. The trace exception is disabled during the execution of debugging program.

23. What are the uses of interrupts in OS?

The uses of interrupts in OS are,

1. To assign priorities
2. Switch from one user program to another.
3. Implementing security.
4. Protection features.
5. Co-ordinate I/O activities.

24. What is the process?

A program together with any information that describes current state of execution is regarded by the OS as an entity called process.

25. Define Multitasking?

Multitasking is a mode of operation in which a processor executes several user programs at the same time.

26. What is time slicing?

Time slicing is a common OS technique that makes multitasking possible. With this technique, each program runs for a short time period called as a time slice, then another program runs for its time slice and so on. The period is determined by continuously running hardware clock, which generates an interrupt every second.

27. What are the three states of a process?

A process can be in one of the three possible states.

1.Running,

2.Runnable.

3.Blocked.

28. Differentiate a process in running and runnable state?

The running state means that the program is currently being executed .The process is runnable if the program is ready for execution but is waiting to be selected by the scheduler.

29. What is program state?

A program state is state which includes register contents, program counter and the program status word.

30. What is DMA? Write the advantages of DMA.

A special control unit that may be provided to allow transfer of a block of data directly between an external device and the main memory without continuous intervention by the processor. This approach is called Direct memory access.

- Computer system performance is improved by direct transfer of data between memory and I/O devices, bypassing the CPU.
- CPU is free to perform operations that do not use system buses.

31. What is the purpose of a DMA controller?

The DMA controller performs the functions that would normally be carried out by the processor when accessing the main memory.

32. What is cycle stealing?

Cycle stealing is an interweaving technique used by DMA controller to steal the memory cycles from the processor.

33. What is a block or burst mode?

The DMA controller may be given exclusive to the main memory to transfer a block of data without interruption. This is known as block or burst mode.

34. What is bus master?

The device that is allowed to initiate data transfers on the bus at any given time is called bus master.

35. What is bus arbitration?

Bus arbitration is the process by which the next device to become the bus master is selected and bus mastership is transferred to it.

36. Name the two approaches to bus arbitration.

The approaches to bus arbitration are,

1. Centralized arbitration
2. Distributed arbitration.

37. What do you mean by distributed arbitration?

Distributed arbitration means that all devices waiting to use the bus have equal responsibility in carrying out the arbitration process, without using a central arbiter.

38. What is the purpose of a bus protocol?

A bus protocol is the set of rules that governs the behavior of various devices connected to the bus.

39. Define master?

Master is a device that initiates data transfer by issuing read or write commands on the bus. Master is also called as initiator.

39. What is a slave?

The device addressed by the master is called as slave. Slave can also called as target.

40. What is a synchronous bus?

In synchronous bus, all devices derive timing information from the common clock line. Equally spaced pulses on this define equal time intervals.

41. What is an asynchronous bus?

In asynchronous bus, controlling data transfer on the bus is based on the use of handshake between the master and slave.

42. Define full handshake. List the two advantages of a full handshake.

An asynchronous logic uses hardware handshaking to exchange information on the bus. Here the clock line is eliminated by two control signals master ready and slave ready.

The advantages of a full handshake are,

1. Highest degree of flexibility is provided.

2. Highest degree of reliability is provided.

43. What are the main advantages of asynchronous bus?

The main advantage of an asynchronous bus is that the handshake process eliminates the need for synchronization of the sender and the receiver blocks, thus simplifying timing design.

44. What is a port?

The side opposite to bus signals in an I/O interface consists of data path with its associated controls to transfer data between the interface and the I/O device. This side called a port.

45. What is the difference between serial port and parallel port?

1. A parallel port transfers data in the form of a number of bits typically 8 or 16 simultaneously to or from the device. A serial port transmits and receives data one bit at a time.

2. Parallel format is suitable for devices that are physically close to the computer. Serial format is much more convenient and cost-effective where longer cables are needed.

46. What is a bridge?

A bridge is an interconnection circuit between two buses. It translates the signals and protocols of one bus into those of the other.

47. What is transaction?

A complete transfer operation on the bus , involving an address and a burst of data is called a transaction.

48. Define SCSI? Or. What is SCSI?

SCSI stands for **Small Computer System Interface**. SCSI refers to the standard bus which is defined by ANSI (American National Standard Institute). under designation X3.131.

SCSI bus the several options. It may be,

- **Narrow bus** (It has 8 data lines transfer 1 byte at a time.)
- **Wide bus (i**t has 16 data lines & transfer 2 byte at a time.)
- **Single-Ended Transmission (**Each signal uses separate wire.)
- **HVD (High Voltage Differential) (** it was a 5v (TTL Cells))
- **LVD (Low Voltage Differential) (**it uses 3.3v)

49. What are the different categories of SCSI bus signals?

SCSI bus signals are classified as,

1.Data signal

2.Phase signal

3.Information signal

4.Handshake.

5.Direction of transfer.

50. What are the objectives of USB?

The objectives of USB are as follows,

1.Provide a simple ,low cost and easy to use interconnection system.

2.Enhance user convenience through a „plug and play" mode operation.

51. What is isochronous? (or) What does isochronous data stream means?

An isochronous data stream means that the successive events are separated by equal periods of time.

52. What is hub?

A hub is the intermediate control point between the host and the I/O device.

53. Define serial port.

A serial port transmits and receives data one word bit at a time.

54. What is meant by bus arbitration?

It is the process by which the next device to become the bus master is selected and the bus mastership is transferred to it.

Types:

There are 2 approaches to bus arbitration. They are,

- Centralized arbitration
- Distributed arbitration

55. Name and give purpose of widely used bus standards?

- **PCI** defines an expansion bus on the motherboard.
- **SCSI** and **USB** are used for connecting additional devices both inside and outside the computer box.
- **SCSI** bus is a high speed parallel bus intended for devices such as disk and video display.
- **USB** uses a serial transmission to suit the needs of equipment ranging from keyboard to game control to internal connection

56. What is distributed arbitration?

Distributed Arbitration:

It means that all devices waiting to use the bus have equal responsibility in carrying out the arbitration process.

57. What is interrupt?

- When a program enters a wait loop, it will repeatedly check the device status. During this period, the processor will not perform any function.
- The Interrupt request line will send a hardware signal called the interrupt signal to the processor.
- On receiving this signal, the processor will perform the useful function during the waiting period.

58. Define DMA? (or) why do we need DMA?

To allow transfer of a block of data directly between an external device and the main memory, without continuous intervention by the processor, a special control unit is provided. This approach is called Direct access memory or DMA.

59. How interrupt request from multiple devices can be handled?

An I/O Device Request An Interrupt By Activating A Bus Line Called Interrupt Request

- SINGLE LEVEL
- MULTI LEVEL

60. What are the components of I/O interface?

. An I/O interface consists of the circuitry required to connect an I/O device to a computer bus. On one side of the interface are the bus signals for address, data and control. On the other side is a data path with its associated controls to transfer data between the interface and the I/O device. This side is called a port. This may be either a serial port or parallel port.

61. Mention the advantage of USB bus.

USB helps to add many devices to a computer system at any time without opening the computer box.

62. What are the i/o data transfer method using memory busses?

Three methods used for data transfer between I/O devices and CPU

- program i/o or polling
- interrupt driven i/o
- direct memory access

63. Distinguish between isolated and memory mapped I/O? (Nov 12)

The isolated I/O method

- Isolates memory and I/O addresses so that memory address values are not affected by interface address assignment since each has its own address space.
- This is the case in computers that employ only one set of read and write signals and do not distinguish between memory and I/O addresses.
- The assigned addresses for interface registers cannot be used for memory words, which reduce the memory address (range available).

The memory mapped I/O

- uses the same address space for both memory and I/O.
- The computer treats an interface register as being part of the memory system.
- In memory mapped I/O organization, there are no specific inputs or output instructions. The CPU can manipulate I/O data residing in interface registers with the same instructions that are used to manipulate memory words.

64. Explain handshaking with respect to data transfer.

Some signals are exchanged between the processor and processor and peripheral prior to the data transfer. This is called handshake signals. Data transfer on the on the bus is based on the use of handshake between the master and the slave

65. What are the functions of a typical I/O interface?

When the CPU issues an I/O command. The command, the command contains address of the device, according to the address, the device is selected. The CPU reads data through buffer from I/P device or writes data through latch to output device addressed

66. How does the processor handle an interrupt request?

The processor instruct an I/O device interface to activate the interrupt-request line as soon as it is ready for a data transfer, then it performs other useful work. It transfer the request to interrupt service routine, which handles the task.

67. Specify the different I/O transfer mechanisms available.

They are two different I/O transfer

- Direct access memory or DMA.
- Bus arbitration

To allow transfer of a block of data directly between an external device and the main memory, without continuous intervention by the processor, a special control unit is used called DMA.

Bus arbitration is the process by which the next device to become the bus master is selected and the bus mastership is transferred to it.

Types:

There are 2 approaches to bus arbitration. They are,

- Centralized arbitration
- Distributed arbitration

68. What are priority groups?

When interrupt requests arrive from two (or) more devices simultaneously, the processor has to decide which request should be serviced first, and which one should be delayed. Devices are organized in groups and each group is connected at priority level

69. What are the operating system routines of a keyboard driver?

Clear keybuf: clears the keyboard buffer

Install keyboard: installs the allegro keyboards interrupt controller

70. How is the interrupt request from a device handled?

When the device interrupt the processor, CPU suspends the current program execution and branches into an **Interrupt Service Routine** (ISR).After execution of ISR, the CPU return back to the interrupted program

71. Why does the DMA gets priority over CPU when both request memory Transfer? (or) state the important of DMA in networks

- A DMA controller connects high speed network to the computer bus
- The disk controller which controls two disks, also has DMA capability and provides two DMA channels

72. Why does DMA have priority over the CPU when both request a memory transfer?

The data transfer is monitored by DMA controller which is known as DMA channel. The CPU is involved only at the beginning and end of the transfer.

When the CPU wishes to read or write a block a data, it issues a command to the DMA channel by sending read/write operation, address of I/O, number of words to be read or written. Hence DMA have priority over the CPU when both request a memory transfer.

73. What is the advantage of using interrupt initiated data transfer over transfer under program control without interrupt?

In the interrupt initiated data transfer, the processor identifies the request and transfer the control Interrupt Service Routine(ISR) to perform the task and its resumes back with the useful task where as, the processor has to waste its time by performing all the task.

74. What is the difference between subroutine and interrupt service routine.

Subroutine: subroutine or the subprogram is the routine which could be called by another subroutine or main routine under program control

Interrupt Service Routine (ISR): ISR is called automatically on the occurrence of an interrupt which is predefined

75. What factors influences the bus design decision?

A bus is a set of lines (wires) designed to transfer all bits of a word from a specified source to a specified destination. Protocol defines certain set of rules that give on the behavior of various devices connected to the bus.

76. Write the advantages of USB.

1. Simple connectivity:
2. Simple cables:
3. One interface for many devices:
4. Automatic configuration:
5. No user setting:

11 Marks

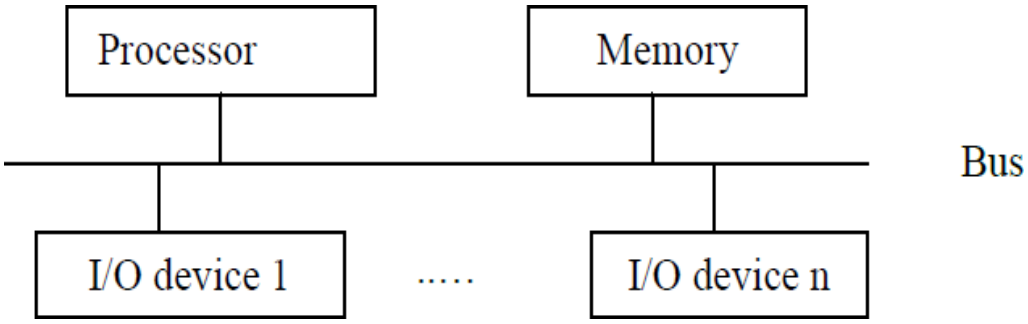
1. Explain about Programmed I/O And Memory Mapped I/O. (Nov 12)

A simple arrangement to connect I/O devices to a computer is to use a single bus structure. It consists of three sets of lines to carry

- Address
- Data
- Control Signals.

- When the processor places a particular address on address lines, the devices that recognize this address responds to the command issued on the control lines.
- The processor request either a read or write operation and the requested data are transferred over the data lines.
- When I/O devices & memory share the same address space, the arrangement is called memory mapped I/O.

Single Bus Structure



Eg:-
Move DATAIN, Ro - Reads the data from DATAIN then into processor register Ro.
Move Ro, DATAOUT - Send the contents of register Ro to location DATAOUT.
DATAIN - Input buffer associated with keyboard.
DATAOUT - Output data buffer of a display unit / printer.

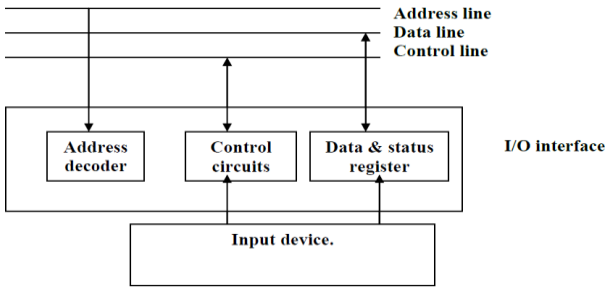


Fig: I/O Interface for an Input Device

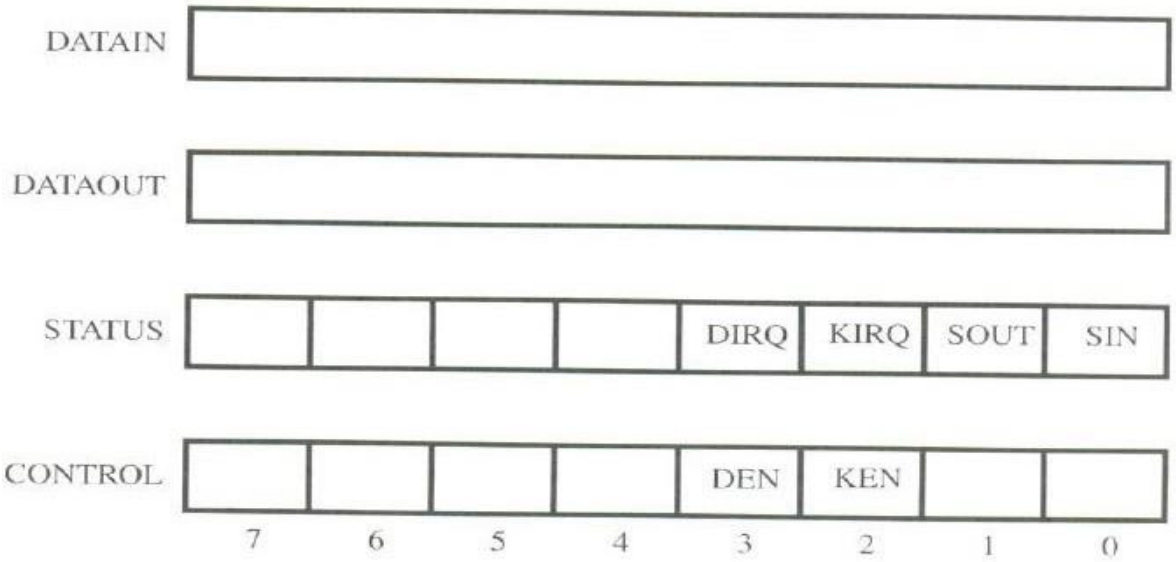
Address Decoder:
It enables the device to recognize its address when the address appears on address lines.

Data register - It holds the data being transferred to or from the processor.
Status register - It contains infn/. Relevant to the operation of the I/O devices.
The address decoder, data & status registers and the control circuitry required to co-ordinate I/O transfers constitute the device‟ s I/F circuit.

For an input device, SIN status flag in used SIN = 1, when a character is entered at the keyboard.

For an output device, SOUT status flag is used SIN = 0, once the char is read by processor.

Eg



DIR Q - Interrupt Request for display.
KIR Q -Interrupt Request for keyboard.
KEN - keyboard enable.
DEN - Display Enable.
SIN, SOUT - status flags.
The data from the keyboard are made available in the DATAIN register & the data sent to the display are stored in DATAOUT register.

Program:
WAIT K Move # Line, Ro
Test Bit #0, STATUS
Branch = 0 WAIT K
Move DATAIN, R1
WAIT D Test Bit #1, STATUS
Branch = 0 WAIT D
Move R1, DATAOUT
Move R1, (Ro)+

Compare #OD, R1
Branch = 0 WAIT K
Move #DOA, DATAOUT
Call PROCESS

Explanation:

- This program, reads a line of characters from the keyboard & stores it in a memory buffer starting at locations LINE.
- Then it calls the subroutine "PROCESS" to process the input line.
- As each character is read, it is echoed back to the display.
- Register Ro is used as a updated using Auto – increment mode so that successive characters are stored in successive memory location.
- Each character is checked to see if there is carriage return (CR), char, which has the ASCII code 0D(hex).
- If it is, a line feed character (on) is sent to more the cursor one line down on the display & subroutine PROCESS is called. Otherwise, the program loops back to wait for another character from the keyboard.

Program Controlled I/O

- Here the processor repeatedly checks a status flag to achieve the required synchronization between Processor & I/O device.(ie) the processor polls the device.
- There are 2 mechanisms to handle I/o operations. They are,
- □ Interrupt,-
- □ DMA (Synchronization is achieved by having I/O device send special over the bus where is ready for data transfer operation)

DMA:

- Synchronization is achieved by having I/O device send special over the bus where is ready for data transfer operation)
- It is a technique used for high speed I/O device.
- Here, the input device transfers data directly to or from the memory without continuous involvement by the processor.

2. Explain about Interrupts With Neat Diagram.(Nov 12)

OVERVIEW:

- Interrupt Hardware
- Enabling and Disabling Interrupts
- Handling Multiple Devices
  - ❖ Vectored Interrupts
  - ❖ Interrupts Nesting
  - ❖ Interrupt Priority
- Controlling Device Requests

- Exceptions

INTERRUPTS:

- An interrupt is an event inside a computer system requiring some urgent action by the CPU. In response to the interrupt, the CPU suspends the current program execution and branches in to an Interrupt Service Routine(ISR).

- The ISR is a program that services the interrupt by taking appropriate actions. After the execution of ISR, the CPU returns back to the interrupted program.

- On returning from ISR, the CPU resumes the old interrupted program as if nothing has taken place.

- This means the CPU should continue from the place when interruption has occurred and the CPU status at that time should be the same.

- For this purpose, the condition of the CPU should be saved before taking up ISR. Before returning from ISR, this saved status should be loaded back into the CPU.

- The processor can perform some useful task while waiting for I/O device to become ready.

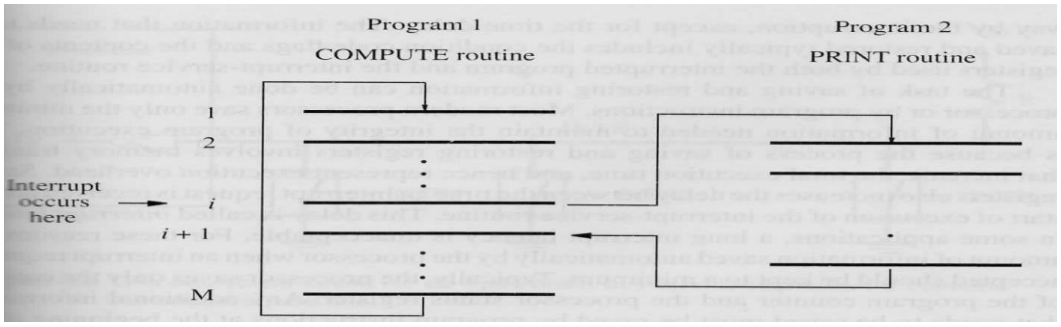


Fig: Interrupt Service Routine

- To allow this to happen, the I/O devices can be allowed to alert the processor when it becomes ready.
- If can be done by sending a hardware signal called interrupt to the processor. Interrupt Request Line Is used for this purpose.
- The process is no longer to continuously check the status of external devices, it can use the waiting period other useful functions.
- Interrupts eliminates waiting period. The routine serviced in response to an interrupt request is called interrupt service routine. The processor acknowledges the interrupt by interrupt acknowledgement signal.
- The interrupt service routine is very similar to sub routine.
- The task of saving and restoring information can be done automatically by processor or by program instruction.
- Saving and restoring registers requires memory transfers are ISR overhead.
- The time between when an interrupt request is received and the start of execution of ISR is called interrupt latency.

- The routine executed in response to an interrupt request is called the interrupt service routine, which is the PRINT routine is our example.
- Interrupts bear considerable resemblance to subroutine calls.
- Assume that an interrupt request arrives during execution of instruction I in figure.
- The processor first completes execution of instruction I, and then it leads the program counter with address of the first instruction of the interrupt service routine.

INTERRUPT HARDWARE:

- An I/O device requests can interrupt by activating a bus line called interrupt request. Computers can request an interrupt. The interrupts are classified as:
-
    - **Single level interrupt**
    - **Multi Level interrupt**

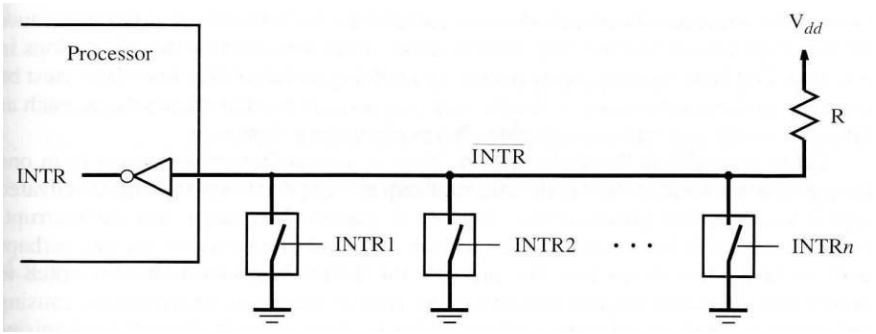


Fig: Common Interrupt Request Line

- A single interrupt request line may be used to serve an I/O device as shown in figure
- All devices are connected to interrupt request line via switches to ground. The device to raise interrupt closes the switch.
- If all interrupt request lines of I/O devices are inactive, then interrupt request line will be equal to V_{dd} . when a device attempts to raise an interrupt by closing the switch, the voltage drops to 0, causing the interrupt request signal INTR, received by the processor to go 1, INTR is logical OR (or) the interrupt request from individual devices.

$INTR = INTR_1 + INTR_2 + ….. + INTR_n$

ENABLING AND DISABLING INTERRUPTS:

- The arrival of an interrupt request from an external device cause the processor to suspend the execution of one program and start the execution of another.
- A simple way is to provide machine instructions, such as interrupt-enable and interrupt-disable that perform this function.
- Let us consider in detail the interrupt handling by single interrupt request line. When a device activates the interrupt request signal, it keeps this signal activate until it is serviced by processor.

- This ensures that this active request signal does not lead to successive interruptions, causing to enter an infinite loop.

This can be handled by the following ways.

1. The first possibility is to have the processor ignore the interrupt request line until the ISR is completed. This can be done using interrupt disable as the first instruction in ISR and interrupt enable as the last instruction.

2. The second option is processor while saving the contents of program counter (pc) and programs status Register(PS) on the stack, it also performs an equivalent of executing an interrupt disable instruction (interrupt enable bit =0), while after completing ISR H is enabled.

The sequence of events in handling interrupts can be summarized as follows.

1. The device raises an interrupt request.
2. The processor interrupts the program currently being executed.
3. Interrupts are disabled by interrupt disable instruction or by setting interrupt enable bit to 0.
4. The device is informed that is interrupt request has been recognized by interrupt acknowledgement.
5. The ISR Is serviced.
6. Interrupts are enabled and execution of the interrupted program is resumed.

HANDLING MULTIPLE DEVICES:

Consider the situation where number of devices capable of initiating interrupts is connected to processor. The devices are operationally independent.

- Polling Scheme
- Vectored interrupts
- Interrupt nesting
- Interrupt priority
- Daisy chain
- Priority Groups

Polling Scheme:

- It is a simplest scheme to identify the interrupting device. The device that raises the interrupt will set one of the bit d (IRQ) in status register t01.
- The processor will poll the devices to find which raised an interrupt.

Disadvantage

- In this technique is ti9me spent in interrogating the IRQ bits of all devices.

Vectored interrupts:

- To reduce time involved in polling process, a device requesting an interrupt may identify itself directly to the processor.
- The code is used to identify the device and it may represent the starting address of the interrupt-service routine for that device.
- If the interrupt produces a call to a predetermined memory location, which is starting address of ISR, then that address is called vectored address and such interrupt are called vectored interrupts.

Interrupt Nesting:

- For some device, a long delay in responding to an interrupts request may lead to error in the operation of computer.
- Such interrupts are acknowledged and serviced even though processor is executing an interrupt service routine (ISR) for another device.
- A system of interrupts that allow an interrupt service routine to be interrupted is known as nested interrupts.

Interrupt Priority:

- When interrupt request arrives from two or more devices simultaneously, the processor has to decide which request should be serviced first. And which one one should be delayed.

- The processor takes this decision with the help of interrupt priorities.

- The processor accepts interrupt request having highest priority.

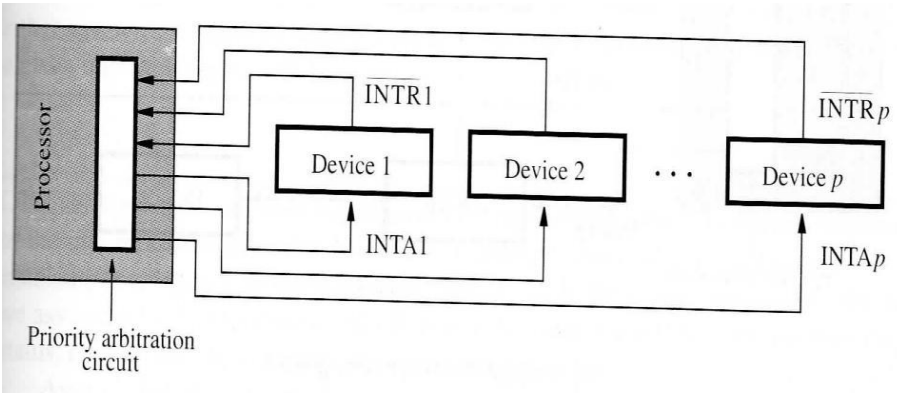


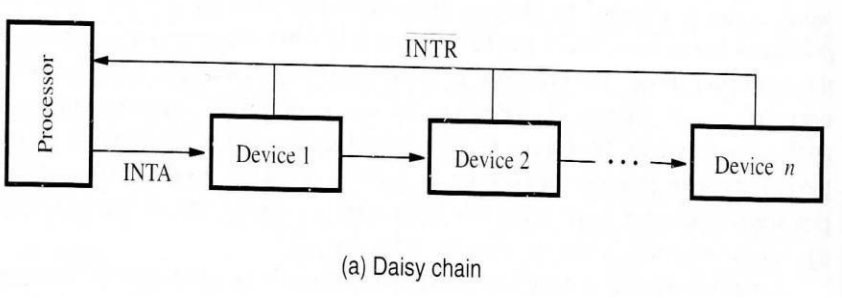
Fig: Handling Priority Interrupts

Daisy chain:

Consider the problem of simultaneous request from two or more devices. The processor has to decide which request to be serviced first.

Priority is determined by the order in which devices are polled. In daisy chain scheme, interrupt request line INTR is common to all devices.

The interrupt acknowledgement INTA, propagates serially through the devices.



(a) Daisy chain

- When several devices raise an interrupt request, the processor responds by setting INTA line to 1. The signal is received by device 1.

- Device 1 passes the signal on to device 2 only if it does not require any service. If device 1 has a pending request for interrupt, it block S the INTA signal and proceeds to put is device identifying code on the data lines.

- Therefore, in daisy chain arrangement, the device that is closest to the processor has the highest priority.

Priority groups:

- In this scheme, devices are organized in groups, and each group is connected at a different priority level within a group, the devices are connected in a daisy chain.

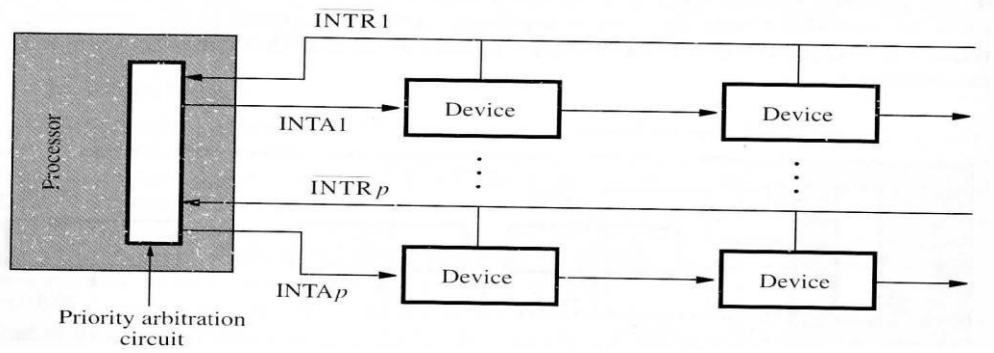


Fig. Arrangement of priority groups

Controlling device request:

I/O device generate an interrupt request when it is ready for data transfer, while handing interrupt requests it is requested to ensure that interrupts request are generated only by those I/O devices that are being used by a given program.

There are two mechanism used to control device request.

- One is at the device end
- And the other at processor end

- At the device end, an interrupt enable bit in the control and status register. The programmer is allowed to set or reset this interrupt-enable bit.

- It the processor end, either an interrupt-enable in the program status register (PS) or a priority structure determines whether the given request will be accepted.

Exceptions:

An interrupt is an event that causes the execution of one program to be suspended and begins execution of another program. Exception is referred to any event that causes an interruption. I/O interrupt is an example of exception.

- **Faults**
  - Faults are exceptions that are detected and serviced before the execution of the faulting instruction.
  - Example: Page fault in virtual memory system.
- **Traps**
  - Traps are exceptions that are reported immediately after the execution of the instructions which causes the problem.
  - Example: op-code field of instruction may not correspond to any level instruction.
- **Aborts:**
  - These are exceptions which do not permit the precise location of the instructions causing the exception to be determined. They report severe error-hardware error.

3. Explain the use of interrupts in operating system

Use Of Interrupts In Operating System:

The operating system is responsible for coordinating all activities within a computer. It makes extensive use of interrupts to perform input/output operations and communicate with and control the execution o user programs.

The interrupt mechanism enables the operating system to assign priorities, switch from one user program to another, implement security and protection features and coordinate input/output activities.

Multitasking is a mode of operation in which a processor executes several user programs at the same time. Each program runs for a short period called a time slice, then another program runs for its time slice and so on.

Operating system routines:

a) OS Initialization, services and scheduler

OSINIT	set interrupt vectors.
	Time-Slice-Clock-SHEDULER
	Software interrupt-OSSERVICES

Keyboard interrupt-IO Data.

OSSERVICES — Examine stack to determine requested operation call appropriate routine.

SCHEDULER — Save program state.
Select a runnable process.
Restore saved context to of new process.
Push new values for PS and PC on stack.
Return from interrupt.

b) Input/output routines

IOINIT — Set process status to Blocked.
Initialize memory Buffer address pointer and counter.
Call device driven to initialize device an enable interrupts in the device interface.

IODATA — Poll devices to determine source of interrupts call appropriate driver.
If END = 1, then set process status to Runnable Return from interrupt.

c) Keyboard Driver.

KBDINIT — Enable interrupt.
Return from subroutine.

KBDDATA — Check device status.
If ready, then transfer character.
If character = CR, then{ set END = 1;
Disable interrupts}
Else set END=0
Return from subroutine.

At the time the operating system is started, an initialization routine, called OSINIT is executed. OSINIT loads the starting address of a routine called SCHEDULER in the interrupt vector corresponding to the timer interrupt. Hence at the end of each time slice, the timer interrupt causes this routine to be executed.
The current state of execution of program is called a process.
A process can be in one of the three states.
Running -the program is currently being executed.
Runable – The program is ready for execution but is waiting to be selected by the scheduler.
Blocked – the program is not to be ready to resume execution for some reason. It may be waiting for completion of an input/output operation.

- Assume program A is in the Running state during a given time slice. At the end of the time slice, the times interrupt the execution of this program and start the execution of SCHEDULER. The information saved, which is called the program state, include register content, the program, counter, and the processor status word.

- Then, SCHEDULER selects for execution some other program B, which was suspended earlier and is in the Runnable state.

- Suppose that program a need to read an input line from the keyboard. It request input/output service from the operating system. It uses the stack or the processor register to pass information to the OS

describing the required operation, the input/output device and the address of a buffer in the program data read where the line should be placed. Then it executes a software interrupt instruction.

- The interrupt vector for the instruction points to the OSSERVICES routine. This routine examines the information on the stack and initiates the requested operation by calling IOINIT, which is responsible for starting input/output operations.

- The IOINIT routine seats the process associated with program A into the Blocked state. This routine carries the initiating address pointers and byte counter, and then calls KBDINIT, which performs any initialization operations needed by the device.

- KBDINIT enables interrupts in the interface circuit by setting the appropriate bit in its control register and then it returns to IONINIT, which returns to OSSERVICES. The keyboard is now ready to participate in a data transfer operations. It will generate an interrupt request whenever a key is pressed. This interrupt points to an OS routine called IODATA.

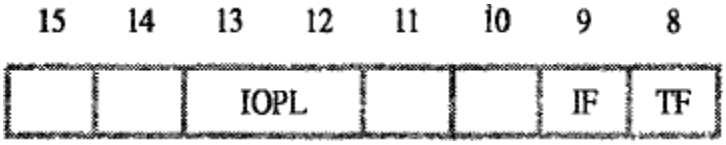
IODATA begins by polling these devices to determine the one requesting service. Then, it calls the device driver KBDDATA, which will transfer one character of data. If the character is a carriage Return, it will set END=1, to inform IODATA that the requested input/output operation has been completed.

4. Explain in detail about the Pentium interrupt structure.

The IA-32 architecture of which are examples of Pentium processors, uses two interrupt request lines, a non-maskable interrupt (NMI) and a maskable interrupt, also called user interrupt request, INTR.
Interrupt requests on NMI are always accepted by the processor.
Requests on INTR are accepted only if they have a higher privilege level than the program currently executing.

INTR requests can also be enabled or disabled by setting an interrupt-enable bit in the processor status register.
In addition to external interrupts, there are many events that arise during program execution that can cause an exception. These include invalid opcodes, division errors, overflow etc. The occurrence of any of these events causes the processor to branch to an interrupt-service routine. Each interrupt is assigned a vector number. In case of INTR, vector number is sent by the I/O device over the bus when the interrupt request is acknowledged. For all other exceptions, the vector number is pressigned. Based on the vector number, the processor determines the starting address of the interrupt-service routine from a table called the Interrupt Descriptor Table.

A companion chip to the Pentium processor is called the Advanced Programmable Interrupt Controller (APIC). Various devices are connected to the processor through this chip. The interrupt controller implements a priority structure among different devices and sends an appropriate vector number to the processor for each device.



The processor status register, called EFLAGS in Intel is shown above. It shows 8 to 15 bits of this register, which contain the Interrupt Enable Flag (IF), the Trap Flag (TF) and the I/O Privilege Level (IOPL). When IF=1, INTR interrupts are accepted. The Trap flag enables trace interrupts after every instruction.

When an interrupt request is received or when an exception occurs, the processor takes the following actions:

1. It pushes the processor status register, the current segment register (CS), and the instruction pointer (EIP) onto the processor stack pointed to by the processor stack pointer, ESP.
2. In the case of an exception resulting from an abnormal execution condition, it pushes a code on the stack describing the cause of the exception.
3. It clears the corresponding interrupt-enable flag, if appropriate, so that further interrupts from the same source are disabled
4. It fetches the starting address of the interrupt request, by transferring input or output data, the interrupt-service routine returns to the interrupted program using a return from interrupt instruction, IRET. This instruction pops EIP,CS and the processor status register from the stack into the corresponding registers, thus restoring the processor state.

Main program

```
        MOV    EOL,0
        MOV    BL,4
        OR     CONTROL,BL        Set KEN to enable keyboard interrupts.
        STI                      Set interrupt flag in processor register.
        .
        .
        .
```

Interrupt-service routine

```
READ    PUSH   EAX               Save register EAX on stack.
        PUSH   EBX               Save register EBX on stack.
        MOV    EAX,PNTR          Load address pointer.
        MOV    BL,DATAIN         Get input character.
        MOV    [EAX],BL          Store character.
        INC    DWORD PTR [EAX]   Increment PNTR.
        CMP    BL,0DH            Check if character is CR.
        JNE    RTRN
        MOV    BL,4
        XOR    CONTROL,BL        Clear bit KEN.
        MOV    EOL,1             Set EOL flag.
RTRN    POP    EBX               Restore register EBX.
        POP    EAX               Restore register EAX.
        IRET
```

Fig. An interrupt servicing routine to read one line from a keyboard using interrupts on IA-32 processors

5. What is DMA? Explain in detail. (Nov 12)

DIRECT MEMORY ACCESS

A special control unit may be provided to allow the transfer of large block of data at high speed directly between the external device and main memory, without continuous intervention by the processor. This approach is called DMA.

DMA transfers are performed by a control circuit called the DMA Controller.

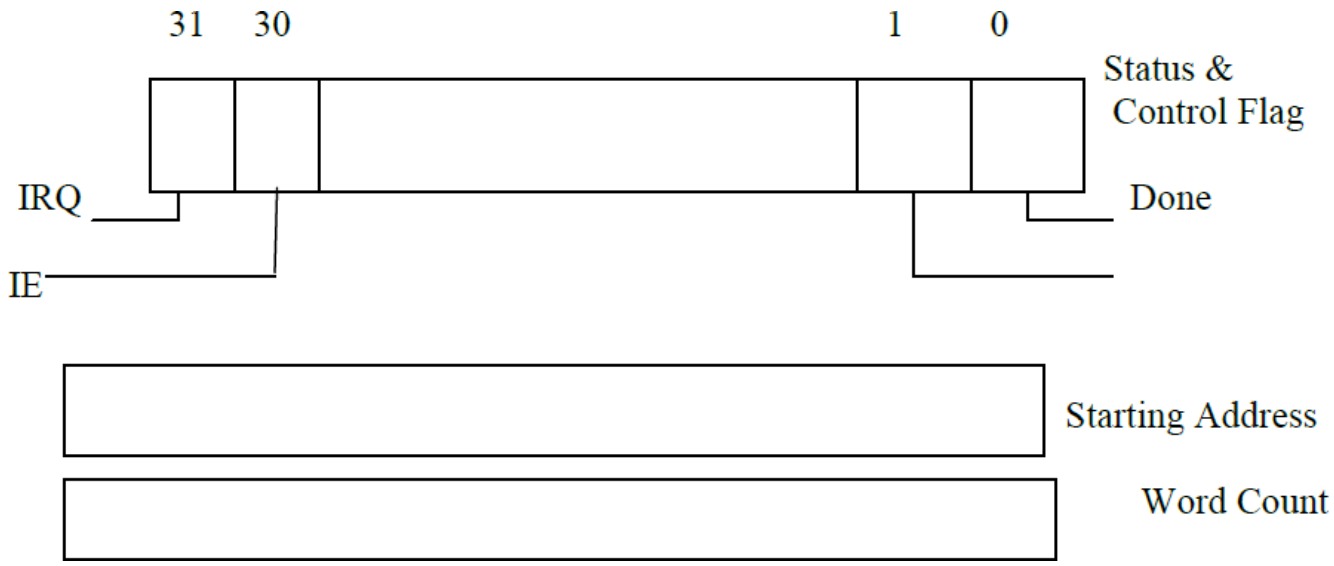
To initiate the transfer of a block of words, the processor sends,

- Starting address
- Number of words in the block
- Direction of transfer.

- When a block of data is transferred , the DMA controller increment the memory address for successive words and keep track of number of words and it also informs the processor by raising an interrupt signal.
- While DMA control is taking place, the program requested the transfer cannot continue and the processor can be used to execute another program.

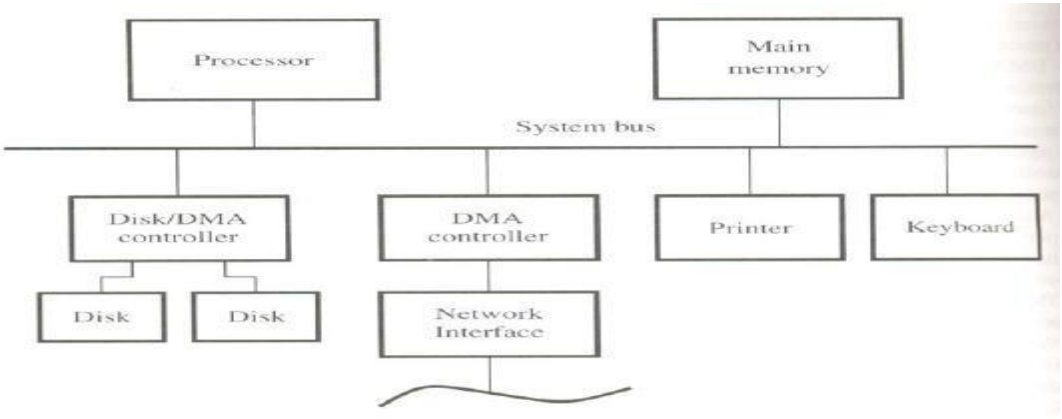
- After DMA transfer is completed, the processor returns to the program that requested the transfer.

Fig: Registers in a DMA Interface



R/W - determines the direction of transfer.
When
R/W =1, DMA controller read data from memory to I/O device.
R/W =0, DMA controller perform write operation.
Done Flag=1, the controller has completed transferring a block of data and is ready to receive another command.
IE=1, it causes the controller to raise an interrupt (interrupt Enabled) after it has completed transferring the block of data.
IRQ=1, it indicates that the controller has requested an interrupt.

Fig: Use of DMA controllers in a computer system



- A DMA controller connects a high speed network to the computer bus . The disk controller two disks, also has DMA capability and it provides two DMA channels.

- To start a DMA transfer of a block of data from main memory to one of the disks, the program write s the address and the word count inf. into the registers of the corresponding channel of the disk controller.
- When DMA transfer is completed, it will be recorded in status and control registers of the DMA channel (ie) Done bit=IRQ=IE=1.

Cycle Stealing:

- Requests by DMA devices for using the bus are having higher priority than processor requests.
- Top priority is given to high speed peripherals such as ,
- Disk
- High speed Network Interface and Graphics display device.
- Since the processor originates most memory access cycles, the DMA controller can be said to steal the memory cycles from the processor. This interviewing technique is called Cycle stealing.

Burst Mode:

The DMA controller may be given exclusive access to the main memory to transfer a block of data without interruption. This is known as Burst/Block Mode

Bus Master:

The device that is allowed to initiate data transfers on the bus at any given time is called the bus master.

Bus Arbitration:

It is the process by which the next device to become the bus master is selected and the bus mastership is transferred to it.

Types:

There are 2 approaches to bus arbitration. They are,

- Centralized arbitration (A single bus arbiter performs arbitration)
- Distributed arbitration (all devices participate in the selection of next bus master).

Centralized Arbitration:

- Here the processor is the bus master and it may grants bus mastership to one of its DMA controller.
- A DMA controller indicates that it needs to become the bus master by activating the Bus Request line (BR) which is an open drain line.
- The signal on BR is the logical OR of the bus request from all devices connected to it.
- When BR is activated the processor activates the Bus Grant Signal (BGI) and indicated the DMA controller that they may use the bus when it becomes free.
- This signal is connected to all devices using a daisy chain arrangement.
- If DMA requests the bus, it blocks the propagation of Grant Signal to other devices and it indicates to all devices that it is using the bus by activating open collector line, Bus Busy (BBSY).

Fig: A simple arrangement for bus arbitration using a daisy chain

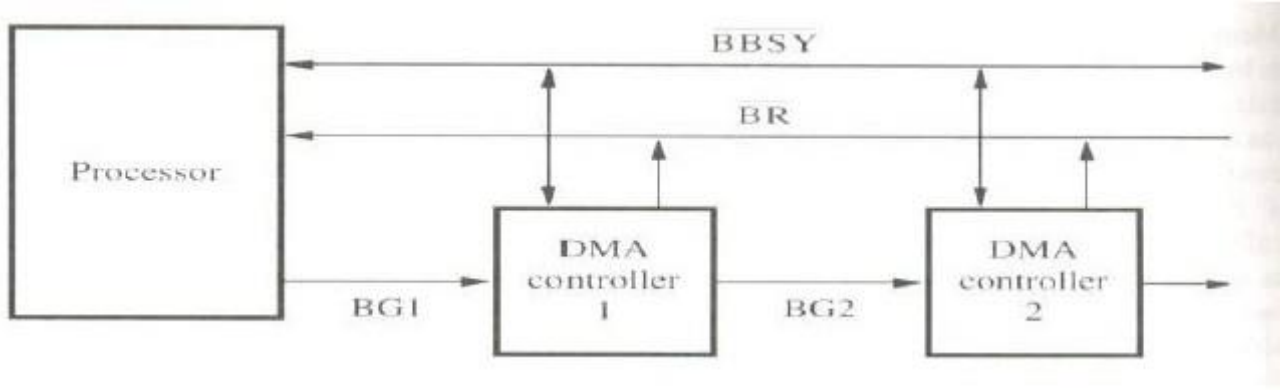
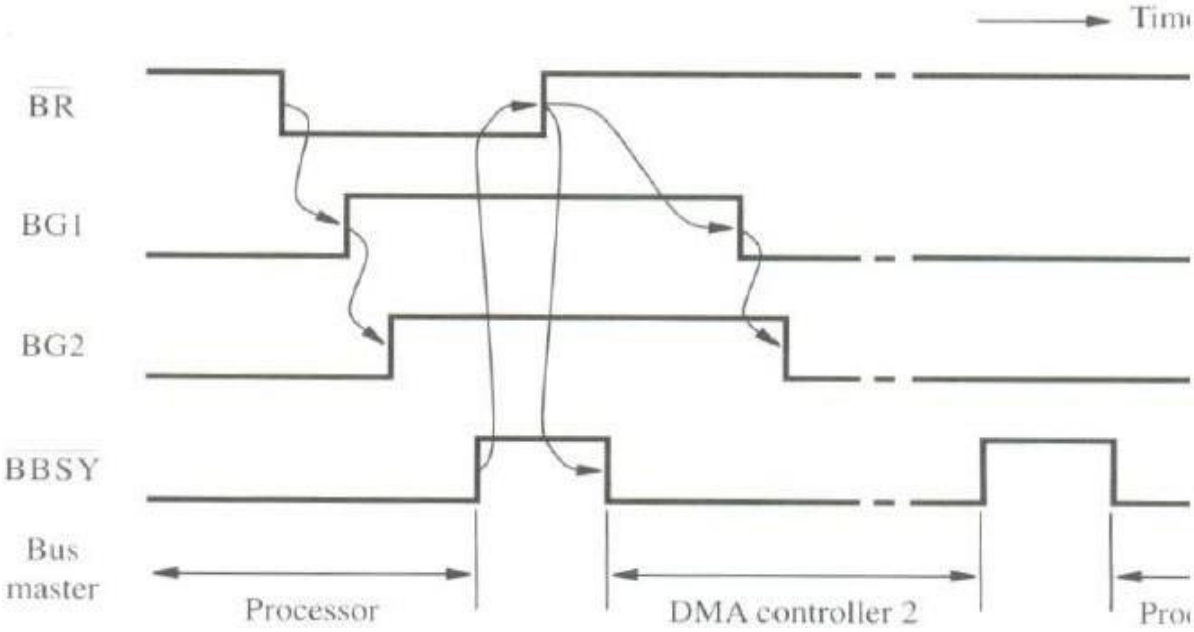


Fig: Sequence of signals during transfer of bus mastership for the devices



The timing diagram shows the sequence of events for the devices connected to the processor is shown. DMA controller 2 requests and acquires bus mastership and later releases the bus.

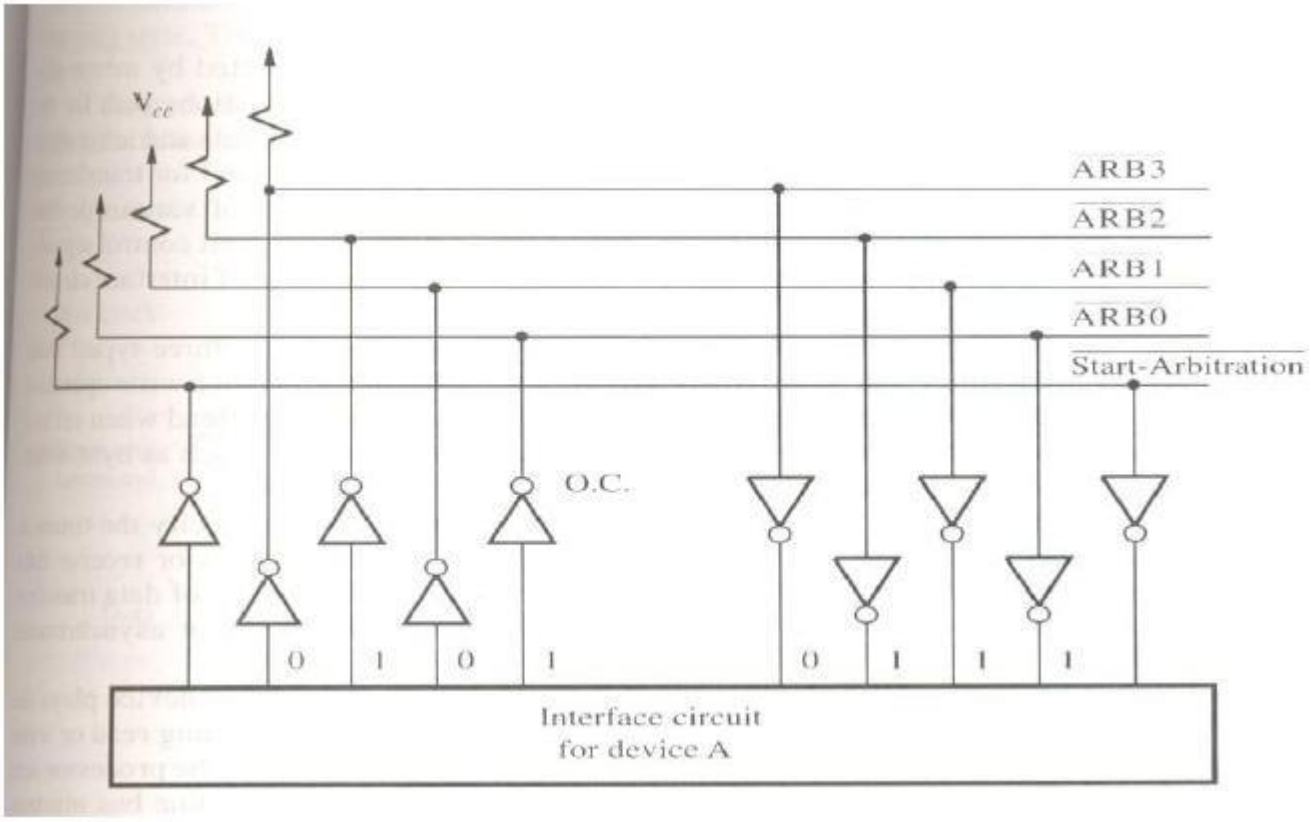
During its tenture as bus master, it may perform one or more data transfer.

After it releases the bus, the processor resources bus mastership

Distributed Arbitration:

It means that all devices waiting to use the bus have equal responsibility in carrying out the arbitration process.

Fig: A distributed arbitration scheme



- Each device on the bus is assigned a 4 bit id.
- When one or more devices request the bus, they assert the Start-Arbitration signal & place their 4 bit ID number on four open collector lines, ARB0 to ARB3.
- A winner is selected as a result of the interaction among the signals transmitted over these lines.
- The net outcome is that the code on the four lines represents the request that has the highest ID number.
- The drivers are of open collector type. Hence, if the i/p to one driver is equal to 1, the i/p to another driver connected to the same bus line is equal to "0"(ie. bus is in low-voltage state).

Eg:

Assume two devices A & B have their ID 5 (0101), 6(0110) and their code is 0111.

Each devices compares the pattern on the arbitration line to its own ID starting from MSB.

If it detects a difference at any bit position, it disables the drivers at that bit position. It does this by placing "0" at the i/p of these drivers.

In our eg. "A" detects a difference in line ARB1, hence it disables the drivers on lines ARB1 & ARB0.

This causes the pattern on the arbitration line to change to 0110 which means that "B" has won the contention.

6. Explain about Buses in detail.

Buses

A bus protocol is the set of rules that govern the behavior of various devices connected to the bus ie, when to place information in the bus, assert control signals etc.

The bus lines used for transferring data is grouped into 3 types. They are,

- Address line

- Data line
- Control line.

Control signals - Specifies that whether read / write operation has to be performed. It also carries timing information / (i.e) they specify the time at which the processor & I/O devices place the data on the bus & receive the data from the bus. During data transfer operation, one device plays the role of a "Master".

Master - device initiates the data transfer by issuing read / write command on the bus. Hence it is also called as Initiator. The device addressed by the master is called as Slave / Target.

Types of Buses:

There are 2 types of buses. They are,

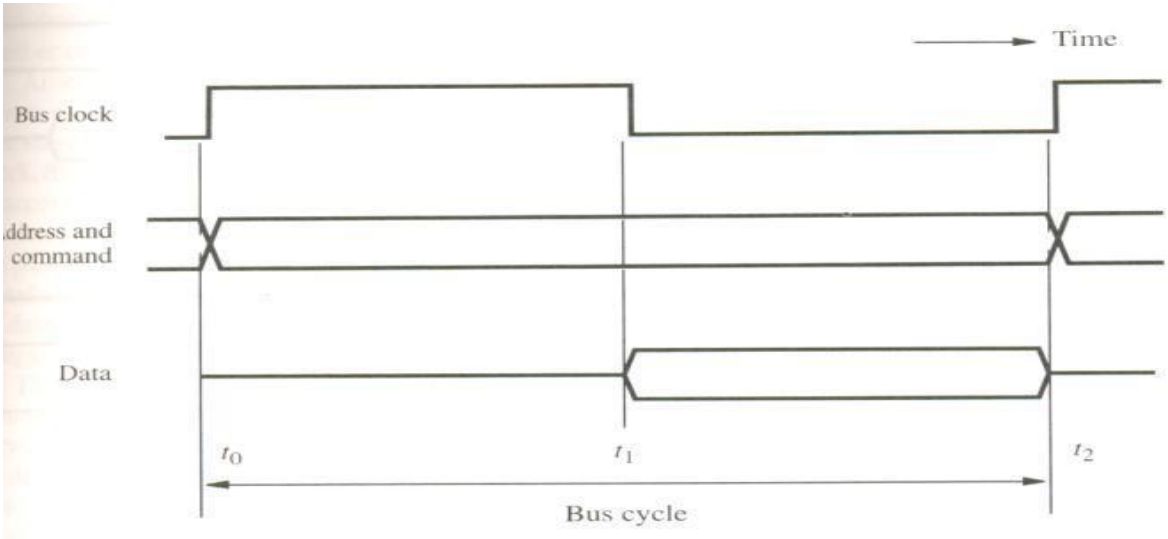
- Synchronous Bus
- Asynchronous Bus

Synchronous Bus:-

In synchronous bus, all devices derive timing information from a common clock line. Equally spaced pulses on this line define equal time. During a bus cycle, one data transfer on take place. The crossing points indicate the tone at which the patterns change.

A signal line in an indeterminate / high impedance state is represented by an intermediate half way between the low to high signal levels.

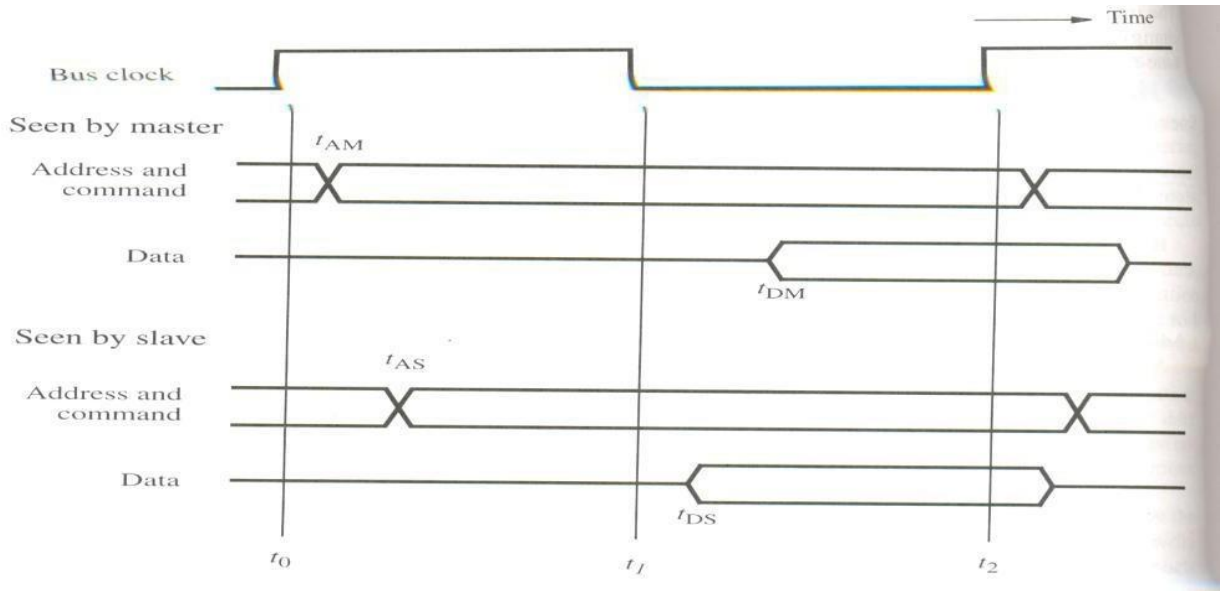
Fig: Timing of an input transfer of a synchronous bus.



- At time to, the master places the device address on the address lines & sends an appropriate command on the control lines.
- In this case, the command will indicate an input operation & specify the length of the operand to be read.
- The clock pulse width t1 – t0 must be longer than the maximum delay between devices connected to the bus.

- The clock pulse width should be long to allow the devices to decode the address & control signals so that the addressed device can respond at time t1.
- The slaves take no action or place any data on the bus before t1.

Fig: A detailed timing diagram for the input transfer



- The picture shows two views of the signal except the clock.
- One view shows the signal seen by the master & the other is seen by the salve.
- The master sends the address & command signals on the rising edge at the beginning of clock period (t0). These signals do not actually appear on the bus until tam.
- Sometimes later, at tAS the signals reach the slave.
- The slave decodes the address & at t1, it sends the requested data.
- At t2, the master loads the data into its i/p buffer.
- Hence the period t2, tDM is the setup time for the masters i/p buffer.
- The data must be continued to be valid after t2, for a period equal to the hold time of that buffers.

Demerits:

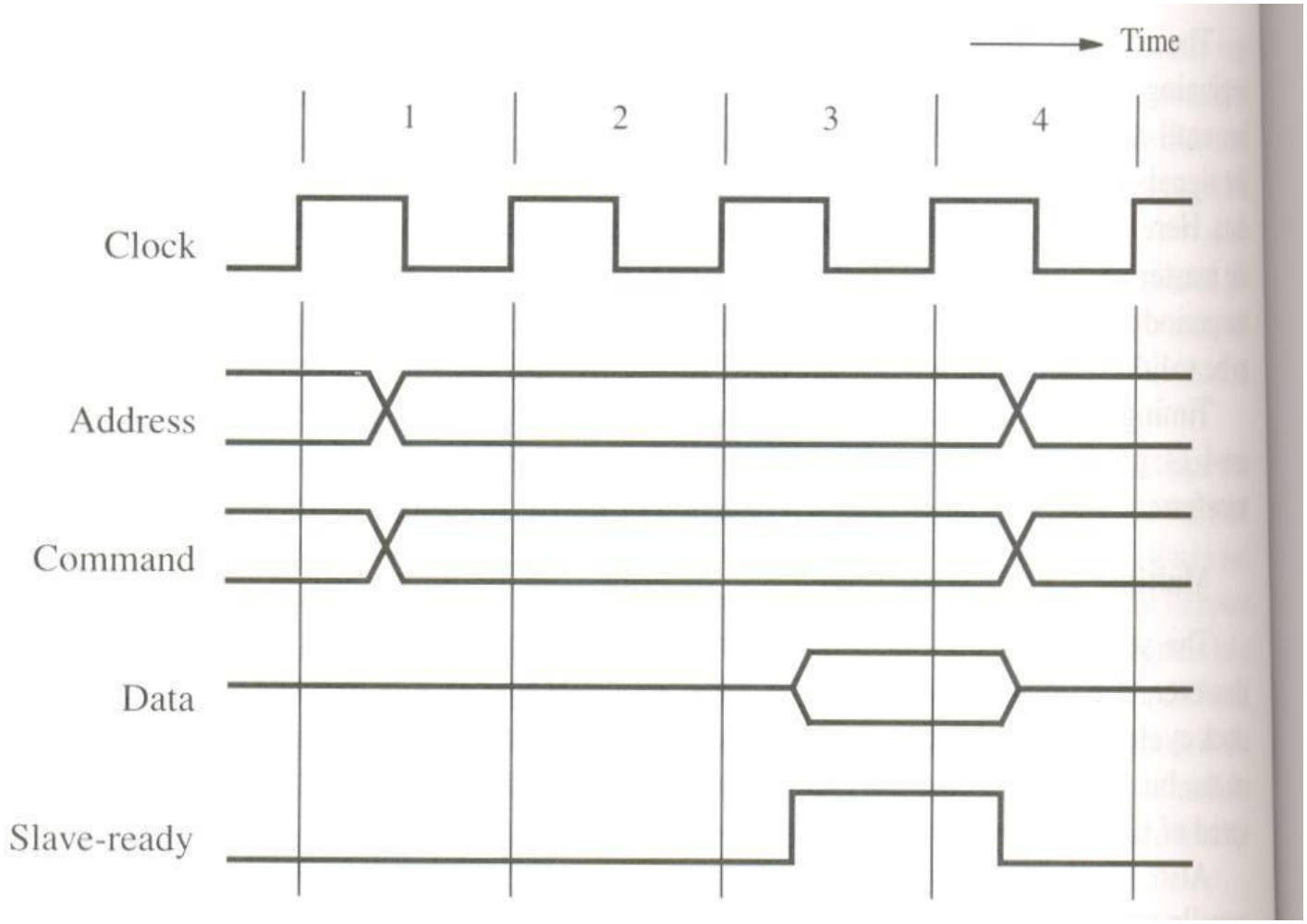
- The device does not respond.
- The error will not be detected.

Multiple Cycle Transfer:-

- During, clock cycle1, the master sends address & cmd infn/. On the bus requesting a read operation.
- The slave receives this information & decodes it.
- At the active edge of the clock (ie) the beginning of clock cycel2, it makes accession to respond immediately.
- The data become ready & are placed in the bus at clock cycle3.
- At the same times, the slave asserts a control signal called slave-ready.

- The master which has been waiting for this signal, strobes, the data to its i/p buffer at the end of clock cycle3.
- The bus transfer operation is now complete & the master sends a new address to start a new transfer in clock cycle4.
- The slave-ready signal is an acknowledgement form the slave to the master confirming that valid data has been sent.

Fig:An input transfer using multiple clock cycles



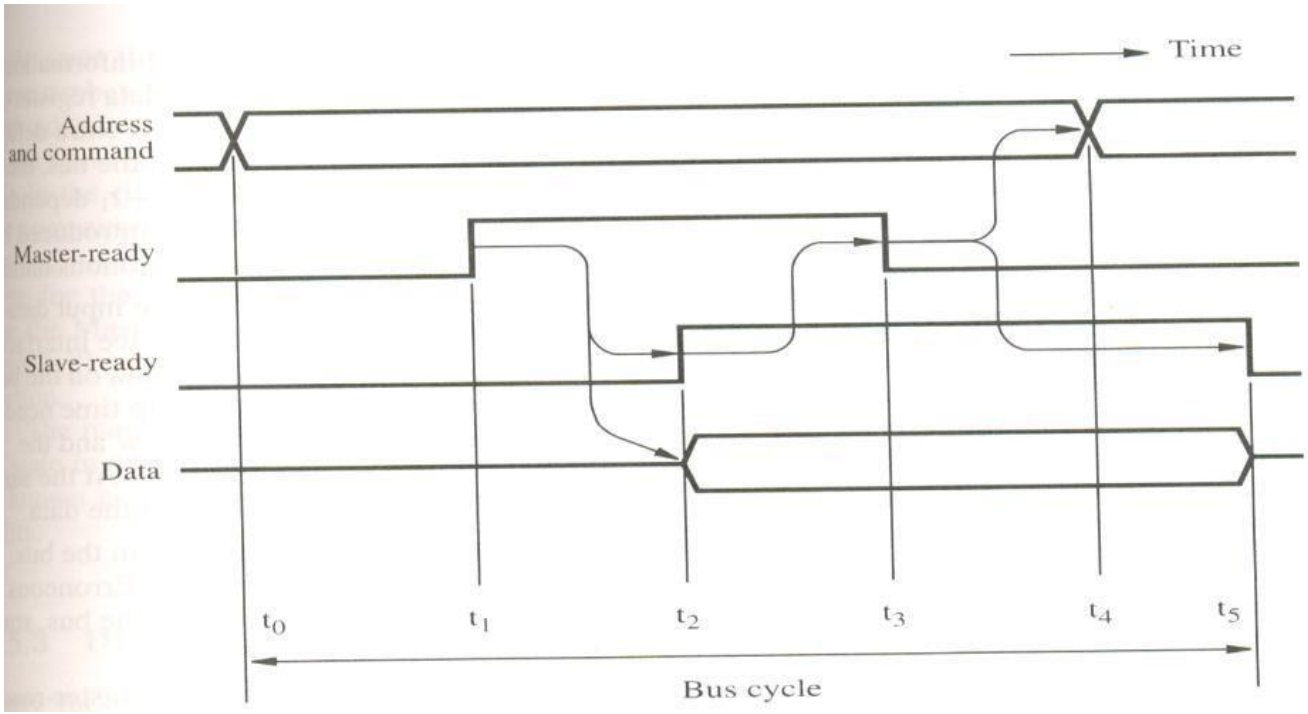
Asynchronous Bus:-

An alternate scheme for controlling data transfer on. The bus is based on the use of handshake between Master & the Slave. The common clock is replaced by two timing control lines.

They are

□ Master–ready
□ Slave ready.

Fig:Handshake control of data transfer during an input operation



The handshake protocol proceed as follows :

- At **t0** □ The master places the address and command information on the bus and all devices on the bus begin to decode the information
- At **t1** □ The master sets the Master ready line to 1 to inform the I/O devices that the address and command information is ready.
- The delay t1 – t0 is intended to allow for any skew that may occurs on the bus.
- The skew occurs when two signals simultaneously transmitted from one source arrive at the destination at different time.
- Thus to guarantee that the Master ready signal does not arrive at any device a head of the address and command information the delay t1 – t0 should be larger than the maximum possible bus skew.

- At **t2** □ The selected slave having decoded the address and command information performs the required i/p operation by placing the data from its data register on the data lines. At the same time, it sets the "slave – Ready" signal to 1.
- At **t3** □ The slave ready signal arrives at the master indicating that the i/p data are available on the bus.
- At **t4** □ The master removes the address and command information on the bus. The delay between t3 and t4 is again intended to allow for bus skew. Errorneous addressing may take place if the address, as seen by some device on the bus, starts to change while the master – ready signal is still equal to 1.
- At **t5** □ When the device interface receives the 1 to 0 tranitions of the Master – ready signal. It removes the data and the slave – ready signal from the bus. This completes the i/p transfer.

- In this diagram, the master place the output data on the data lines and at the same time it transmits the address and command information.
- The selected slave strobes the data to its o/p buffer when it receives the Master-ready signal and it indicates this by setting the slave – ready signal to 1.
- At time t0 to t1 and from t3 to t4, the Master compensates for bus.

- A change of state is one signal is followed by a change is the other signal. Hence this scheme is called as **Full Handshake**.
- It provides the higher degree of flexibility and reliability.

7. What are interface circuits? Explain

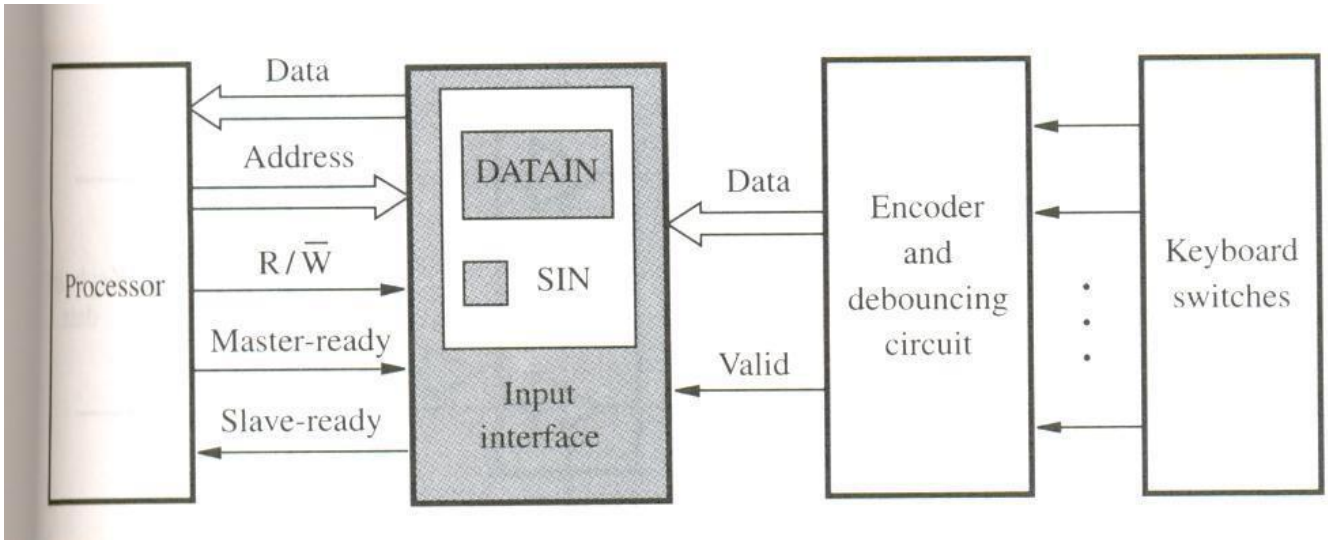
An I/O interface consists of the circuitry required to connect an I/O device to a computer bus. On one side of the interface are the bus signals for address, data and control. On the other side is a data path with its associated controls to transfer data between the interface and the I/O device. This side is called a port. This may be either a serial port or parallel port.

An I/O interface does the following

1. Provides a storage buffer for at least one word of data
2. Contains status flags that can be accessed by the processor to determine whether the buffer is full or empty.
3. Contains address-decoding circuitry to determine when it is being addressed by the processor.
4. Generates the appropriate timing signals required by the bus control scheme.
5. Performs any format conversion that may be necessary to transfer data between the bus and the I/O device, such as parallel -serial conversion in case of serial port.

Parallel port

A parallel port transfers data in the form of a number of bits, typically 8 or 16, simultaneously to or from the device. The connection between the device and the computer uses a multiple-pin connector and a cable with as many wires, typically arranged in a flat configuration. This arrangement is suitable for devices that are physically close to the computer.

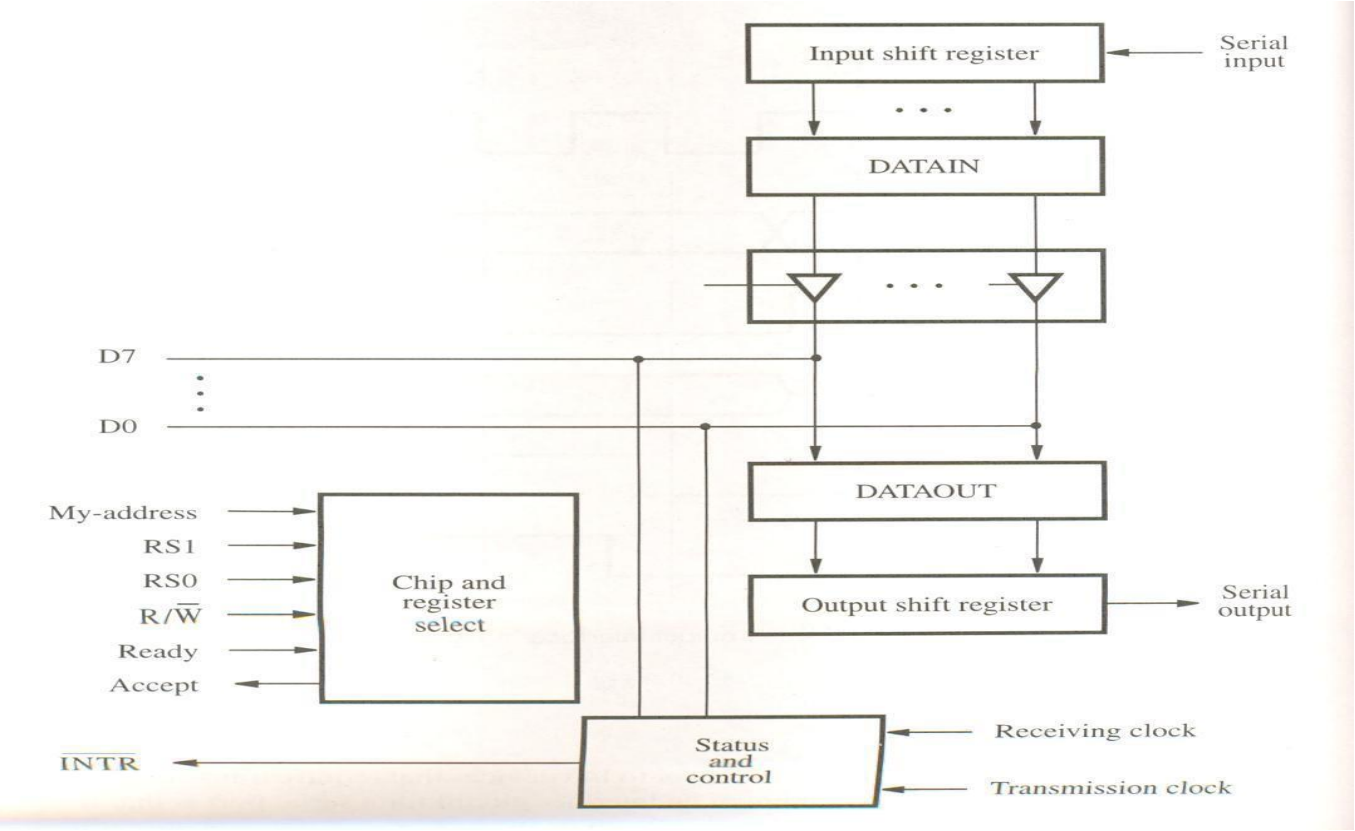


- The output of the encoder consists of the bits that represent the encoded character and one signal called valid,which indicates the key is pressed.
- The information is sent to the interface circuits,which contains a data register,DATAIN and a status flag SIN.

- When a key is pressed, the Valid signal changes from 0 to1,causing the ASCII code to be loaded into DATAIN and SIN set to 1.
- The status flag SIN set to 0 when the processor reads the contents of the DATAIN register.
- The interface circuit is connected to the asynchronous bus on which transfers are controlled using the Handshake signals Master ready and Slave-ready.

Serial port

A serial port transmits and receives data one bit at a time. The serial format is much more convenient and cost-effective where longer cables are needed. A serial port is used to connect the processor to I/O devices that require transmission of data one bit at a time. The key feature of an interface circuit for a serial port is that it is capable of communicating in a bit-serial fashion on the device side and in a parallel fashion on the bus side.



8. Explain in detail about PCI bus.

PCI: (Peripheral Component Inter Connect)

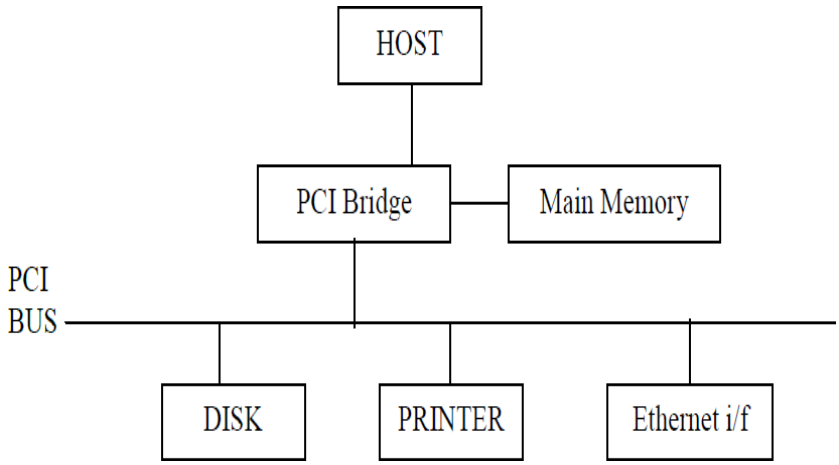
- PCI is developed as a low cost bus that is truly processor independent.
- It supports high speed disk, graphics and video devices.
- PCI has plug and play capability for connecting I/O devices.
- To connect new devices, the user simply connects the device interface board to the bus.

Data Transfer:

- The data are transferred between cache and main memory is the bursts of several words and they are stored in successive memory locations.

- When the processor specifies an address and request a „read" operation from memory, the memory responds by sending a sequence of data words starting at that address.
- During write operation, the processor sends the address followed by sequence of data words to be written in successive memory locations.
- PCI supports read and write operation.
- A read / write operation involving a single word is treated as a burst of length one.
- PCI has three address spaces. They are
    - Memory address space
    - I/O address space
    - Configuration address space
- I/O address space → It is intended for use with processor
- Configuration space → It is intended to give PCI, its plug and play capability.
- PCI Bridge provides a separate physical connection to main memory.
- The master maintains the address information on the bus until data transfer is completed.
- At any time, only one device acts as bus master.
- A master is called an initiator in PCI which is either processor or DMA.
- The addressed device that responds to read and write commands is called a target.
- A complete transfer operation on the bus, involving an address and bust of data is called a transaction.

Fig: Use of a PCI bus in a Computer system



Data Transfer Signals on PCI Bus:

Name Function

CLK - 33 MHZ / 66 MHZ clock

FRAME # - Sent by the indicator to indicate the duration of transaction

AD - 32 address / data line

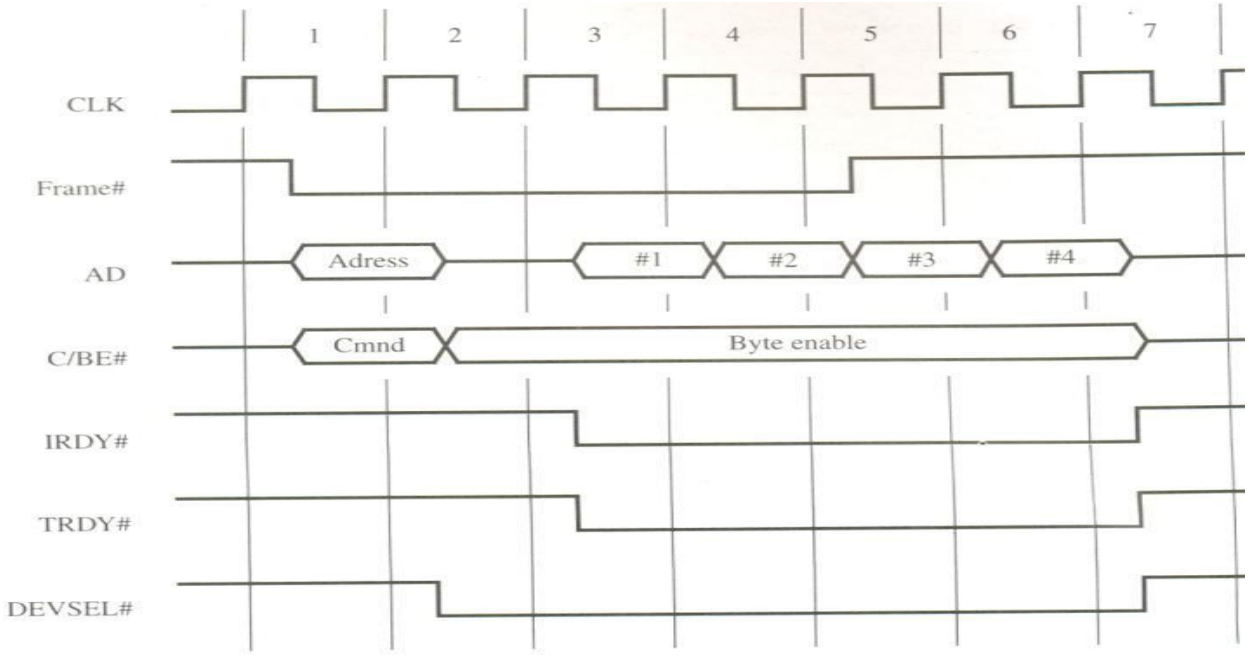
C/BE # - 4 command / byte Enable Lines

IRDY, TRDYA - Initiator Ready, Target Ready Signals

DEVSEL # - A response from the device indicating that it has recognized its address and is ready for data transfer transaction.

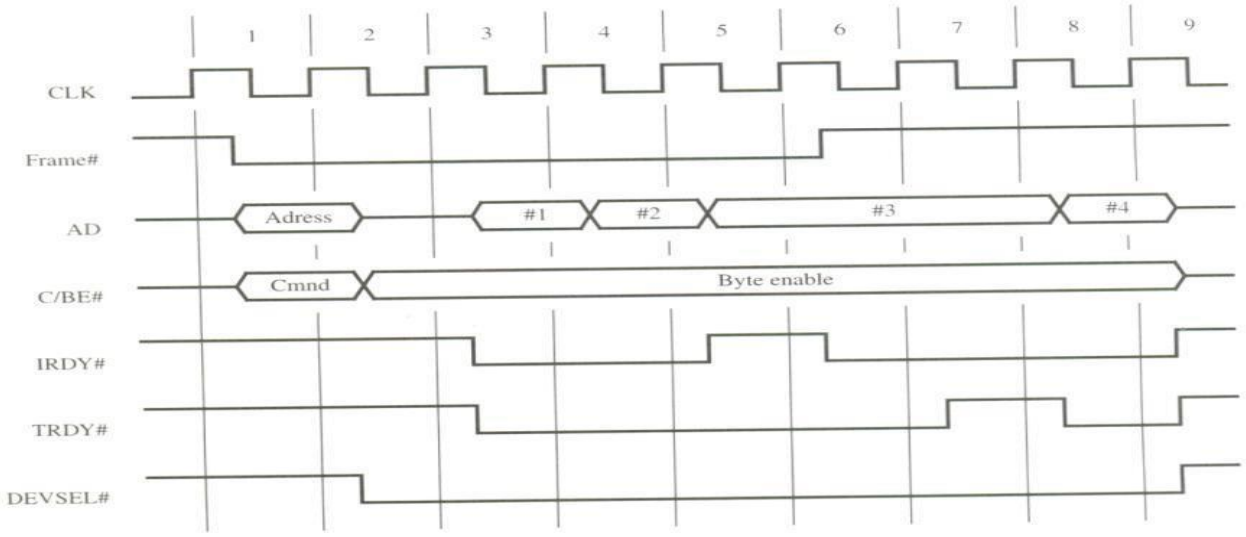
IDSEL # - Initialization Device Select Individual word transfers are called phases.

Fig: Read operation an PCI Bus



- In Clock cycle1, the processor asserts FRAME # to indicate the beginning of a transaction; it sends the address on AD lines and command on C/BE # Lines.
- Clock cycle2 is used to turn the AD Bus lines around ; the processor ; The processor removes the address and disconnects its drives from AD lines.
- The selected target enable its drivers on AD lines and fetches the requested data to be placed on the bus.
- It asserts DEVSEL # and maintains it in asserted state until the end of the transaction.
- C/BE # is used to send a bus command in clock cycle and it is used for different purpose during the rest of the transaction.
- During clock cycle 3, the initiator asserts IRDY #, to indicate that it is ready to receive data.
- If the target has data ready to send then it asserts TRDY #. In our eg, the target sends 3 more words of data in clock cycle 4 to 6.
- The indicator uses FRAME # to indicate the duration of the burst, since it read 4 words, the initiator negates FRAME # during clock cycle 5.
- After sending the 4th word, the target disconnects its drivers and negates DEVSEL # during clock cycle 7.

Fig: A read operation showing the role of IRDY# / TRY#



- It indicates the pause in the middle of the transaction.
- The first and words are transferred and the target sends the 3rd word in cycle 5.
- But the indicator is not able to receive it. Hence it negates IRDY#.
- In response the target maintains 3rd data on AD line until IRDY is asserted again.
- In cycle 6, the indicator asserts IRDY. But the target is not ready to transfer the fourth word immediately; hence it negates TRDY in cycle 7. Hence it sends the 4th word and asserts TRDY# at cycle 8.

Device Configuration:

- The PCI has a configuration ROM memory that stores information about that device.
- The configuration ROM" s of all devices are accessible in the configuration address space.
- The initialization s/w read these ROM" s whenever the S/M is powered up or reset
- In each case, it determines whether the device is a printer, keyboard, Ethernet interface or disk controller.
- Devices are assigned address during initialization process and each device has an w/p signal called IDSEL # (Initialization device select) which has 21 address lines (AD) (AD to AD31).
- During configuration operation, the address is applied to AD i/p of the device and the corresponding AD line is set to and all other lines are set to 0.

AD11 - AD31 **Upper address line**

A00 - A10 **Lower address line** → Specify the type of the operation and to access the content of device configuration ROM.

The configuration software scans all 21 locations. PCI bus has interrupt request lines. Each device may requests an address in the I/O space or memory space

Electrical Characteristics:

- The connectors can be plugged only in compatible motherboards PCI bus can operate with either 5 – 33V power supply.
- The motherboard can operate with signaling system.

9. Explain in detail about SCSI bus.

SCSI Bus:- (Small Computer System Interface)

SCSI refers to the standard bus which is defined by ANSI (American National Standard Institute).

SCSI bus the several options. It may be,

Narrow bus - It has 8 data lines & transfers 1 byte at a time.
Wide bus - It has 16 data lines & transfer 2 byte at a time.
Single-Ended Transmission - Each signal uses separate wire.
HVD (High Voltage Differential) - It was 5v (TTL cells)
LVD (Low Voltage Differential) - It uses 3.3v

Because of these various options, SCSI connector may have 50, 68 or 80 pins. The data transfer rate ranges from 5MB/s to 160MB/s 320Mb/s, 640MB/s.

The transfer rate depends on,

- Length of the cable
- Number of devices connected.
- To achieve high transfer rat, the bus length should be 1.6m for SE signaling and 12m for LVD signaling.
- The SCSI bus us connected to the processor bus through the SCSI controller.
- The data are stored on a disk in blocks called sectors.
- Each sector contains several hundreds of bytes. These data will not be stored in contiguous memory location.
- SCSI protocol is designed to retrieve the data in the first sector or any other selected sectors.
- Using SCSI protocol, the burst of data are transferred at high speed.
- The controller connected to SCSI bus is of 2 types. They are,
    - Initiator
    - Target

Initiator:

It has the ability to select a particular target & to send commands specifying the operation to be performed.

They are the controllers on the processor side.

Target:

The disk controller operates as a target.

It carries out the commands it receive from the initiator. The initiator establishes a logical connection with the intended target.

Steps:

- Consider the disk read operation, it has the following sequence of events.
- The SCSI controller acting as initiator, contends process, it selects the target controller & hands over control of the bus to it.
- The target starts an output operation, in response to this the initiator sends a command specifying the required read operation.

- The target that it needs to perform a disk seek operation, sends a message to the initiator indicating that it will temporarily suspends the connection between them.
- Then it releases the bus.
- The target controller sends a command to disk drive to move the read head to the first sector involved in the requested read in a data buffer. When it is ready to begin transferring data to initiator, the target requests control of the bus. After it wins arbitration, it reselects the initiator controller, thus restoring the suspended connection.
- The target transfers the controls of the data buffer to the initiator & then suspends the connection again. Data are transferred either 8 (or) 16 bits in parallel depending on the width of the bus.
- The target controller sends a command to the disk drive to perform another seek operation. Then it transfers the contents of second disk sector to the initiator. At the end of this transfer, the logical connection b/w the two controller is terminated.
- As the initiator controller receives the data, if stores them into main memory using DMA approach.
- The SCSI controller sends an interrupt to the processor to inform it that the requested operation has been completed.

Bus Signals:-

- The bus has no address lines.
- Instead, it has data lines to identify the bus controllers involved in the selection / reselection / arbitration process.
- For narrow bus, there are 8 possible controllers numbered from 0 to 7.
- For a wide bus, there are 16 controllers.
- Once a connection is established b/w two controllers, these is no further need for addressing & the datalines are used to carry the data.

SCSI bus signals:

Category	Name	Function
Data	- DB (0) to DB (7) - DB(P)	Datalines Parity bit for data bus.
Phases	- BSY - SEL	Busy Selection
Information type	- C/D - MSG	Control / Data Message
Handshake	- REQ - ACK	Request Acknowledge
Direction of transfer	I/O	Input / Output
Other	- ATN - RST	Attention Reset.

- All signal names are proceeded by minus sign.
- This indicates that the signals are active or that the dataline is equal to 1, when they are in the low voltage state.

Phases in SCSI Bus:-

The phases in SCSI bus operation are,

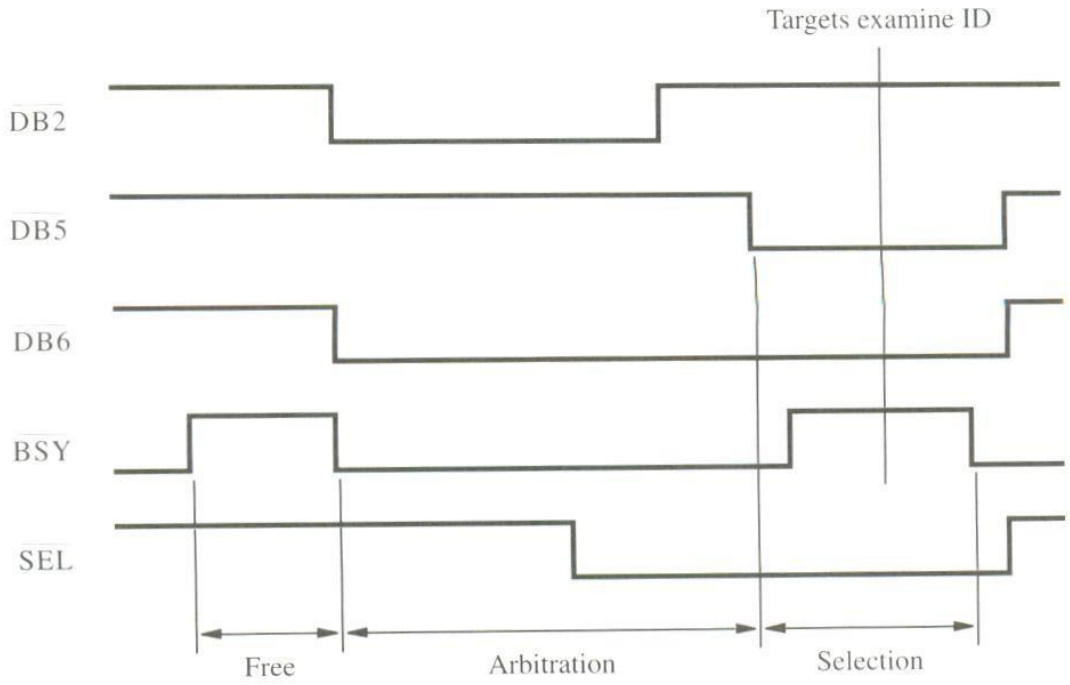
□ Arbitration
□ Selection
□ Information transfer
□ Reselection

Arbitration:-

- When the –BSY signal is in inactive state, the bus will he free & any controller can request the use of the bus.
- Since each controller may generate requests at the same time, SCSI uses distributed arbitration scheme.
- Each controller on the bus is assigned a fixed priority with controller 7 having the highest priority.
- When –BSY becomes active, all controllers that are requesting the bus examines the data lines & determine whether the highest priority device is requesting the bus at the same time.
- The controller using the highest numbered line realizes that it has won the arbitration process.
- At that time, all other controllers disconnect from the bus & wait for –BSY to become inactive again.

Fig:Arbitration and selection on the SCSI bus.Device 6 wins arbitration and select device 2

Selection:



Selection:

Here Device wons arbitration and it asserts –BSY and –DB6 signals.

The Select Target Controller responds by asserting –BSY.

This informs that the connection that it requested is established.

Reselection:

The connection between the two controllers has been reestablished,with the target in control the bus as required for data transfer to proceed.

10. What is USB? Explain in detail.

USB – Universal Serial Bus

USB supports 3 speed of operation. They are,

- Low speed (1.5Mb/s)
- Full speed (12mb/s)
- High speed (480mb/s)

The USB has been designed to meet the key objectives. They are,

□ It provide a simple, low cost & easy to use interconnection s/m that overcomes the difficulties due to the limited number of I/O ports available on a computer.

□ It accommodate a wide range of data transfer characteristics for I/O devices including telephone & Internet connections.

□ Enhance user convenience through**'Plug & Play'** mode of operation.

Port Limitation:-

- Normally the system has a few limited ports.
- To add new ports, the user must open the computer box to gain access to the internal expansion bus & install a new interface card.
- The user may also need to know to configure the device & the s/w.

Merits of USB:-

USB helps to add many devices to a computer system at any time without opening the computer box.

Device Characteristics:-

- The kinds of devices that may be connected to a cptr cover a wide range of functionality.
- The speed, volume & timing constrains associated with data transfer to & from devices varies significantly.

Eg:1 Keyboard □ Since the event of pressing a key is not synchronized to any other event in a computer system, the data generated by keyboard are called asynchronous.

The data generated from keyboard depends upon the speed of the human operator which is about 100bytes/sec.

Eg:2 Microphone attached in a cptr s/m internally / externally

The sound picked up by the microphone produces an analog electric signal, which must be converted into digital form before it can be handled by the cptr.

This is accomplished by sampling the analog signal periodically.

The sampling process yields a continuous stream of digitized samples that arrive at regular intervals, synchronized with the sampling clock. Such a stream is called isochronous (ie) successive events are separated by equal period of time.

If the sampling rate in „S" samples/sec then the maximum frequency captured by sampling process is s/2.

A standard rate for digital sound is 44.1 KHz.

Requirements for sampled Voice:-

It is important to maintain precise time (delay) in the sampling & replay process.

A high degree of jitter (Variability in sampling time) is unacceptable.

Eg-3:Data transfer for Image & Video:-

The transfer of images & video require higher bandwidth.

The bandwidth is the total data transfer capacity of a communication channel.

To maintain high picture quality, The image should be represented by about 160kb, & it is transmitted 30 times per second for a total bandwidth if 44MB/s.

Plug & Play:-

The main objective of USB is that it provides a plug & play capability.

The plug & play feature enhances the connection of new device at any time, while the system is operation.

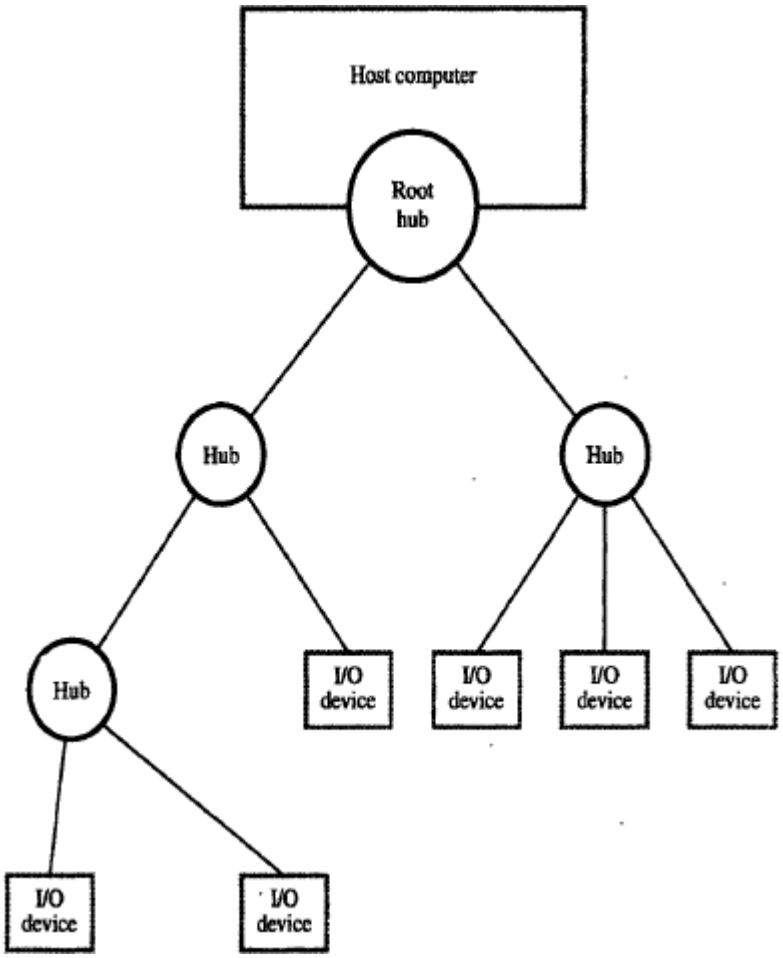
The system should,

- □ Detect the existence of the new device automatically.
- □ Identify the appropriate device driver s/w.
- □ Establish the appropriate addresses.
- □ Establish the logical connection for communication.

USB Architecture:-

- USB has a serial bus format which satisfies the low-cost & flexibility requirements.
- Clock & data information are encoded together & transmitted as a single signal.
- There are no limitations on clock frequency or distance arising form data skew, & hence it is possible to provide a high data transfer bandwidth by using a high clock frequency.
- To accommodate a large no/. of devices that can be added / removed at any time, the USB has the tree structure.

Fig:USB Tree Structure



- Each node of the tree has a device called hub, which acts as an intermediate control point b/w host & I/O devices.
- At the root of the tree, the „root hub" connects the entire tree to the host computer.
- The leaves of the tree are the I/O devices being served (eg. Keyboard, speaker, digital TV), which are called functions in USB terminology.
- The tree structure enables many devices to be connected while using only simple point-point serial links.
- Each hub has a port where devices maybe connected, including other hubs.
- In normal operation, a hub copies a message that it receives from its upstream connection to all its downstream ports. A message sent by host computer is broadcast to all I/O devices, but only the addressed device will respond to that message.

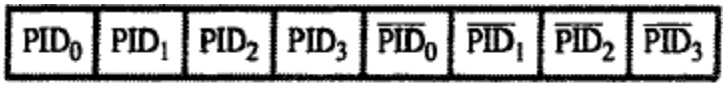
Addressing

- A device usually has several addressable locations to enable the software to send and receive control and status information and to transfer data.
- When a USB is connected to a host computer, its root hub is attached to the processor bus, where it appears as a single device.
- The host software communicates with individual devices attached to the USB by sending packets of information, which the root hub forwards to the appropriate device in the USB tree.
- Each device on the USB, whether it is a hub or an I/O device, is assigned a 7-bit address. This address is local to the USB tree and is not related in any way to the addresses used on the processor bus.

- A hub may have any number of devices and addresses are assigned arbitrarily.
- When a device is first connected to a hub or when it is powered on, it has the address 0.
- Locations in the device to or from which data transfer can take place, such as status, control and data registers are called endpoints. They are identified by a 4-bit number.

USB protocols

- The information transferred on the USB can be divided into two broad categories: control and data.
- Control packets perform tasks such as addressing a device to initiate data transfer, acknowledging that data have been received correctly, or indicating an error.
- Data packets carry information that is delivered to a device.
- A packet consists of one or more fields containing different kinds of information. The first field of any packet is called the packet identifier, PID, which identifies the type of that packet.
- There are 4 bits of information, but they are transmitted twice. The first time they are sent with their true values, and the second time with each bit complemented.

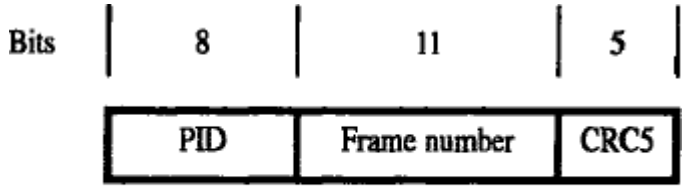


(a) Packet identifier field

Isochronous Traffic on USB

One of the key objectives of the USB is to support the transfer of isochronous data, such as sampled voice, in a simple way. Devices that generate or receive isochronous data require a time reference to control the sampling process. To provide this reference, transmission over the USB is divided into frames of equal length. A frame is 1 ms long for low- and full-speed data. The root hub generates a Start Of Frame control packet (SOF) precisely once every 1 ms to mark the beginning of a new frame.

The arrival of an SOF packet at any device constitutes a regular clock signal that the device can use for its own purposes. The main requirement for isochronous traffic is consistent timing.



(a) SOF Packet

Isochronous data are allowed only on full-speed and high-sped links. For high-speed links, SOF packet is repeated eight times at equal intervals within the 1 ms frame to create eight microframes of 125 µs each.

Electrical Characteristics

The cables used for USB connections consist of four wires.

Two used to carry power, +5v and Ground.