Input and Output Organization: Data Transfer Techniques

- The I/O subsystem of a computer provides an efficient mode of communication between the central system and the outside environment.
- It handles all the input-output operations of the computer system.

Input and Output Organization: Data Transfer Techniques

- **Peripheral Devices:**
  - Input or output devices that are connected to computer are called **peripheral devices.**
  - These devices are designed to read information into or out of the memory unit upon command from the CPU and are considered to be the part of computer system. These devices are also called **peripherals.**

  For example: Keyboards, display units and printers are common peripheral devices.

- There are three types of peripherals:
  - **Input peripherals :** Allows user input, from the outside world to the computer. Example: Keyboard, Mouse etc.
  - **Output peripherals:** Allows information output, from the computer to the outside world. Example: Printer, Monitor etc.
  - **Input-Output peripherals:** Allows both input (from outside world to computer) as well as, output(from computer to the outside world). Example: Touch screen etc.

Input and Output Organization: Data Transfer Techniques

- **Interfaces:**

  Interface is a shared boundary between two separate components of the computer system which can be used to attach two or more components to the system for communication purposes.

- There are two types of interface:
  - **CPU Inteface**
  - **I/O Interface**

- **Input-Output Interface:**
  - Peripherals connected to a computer need special communication links for interfacing with CPU.
  - It is a special hardware components between the CPU and peripherals to control or manage the input-output transfers. These components are called input-output interface units because they provide communication links between processor bus and peripherals.
  - They provide a method for transferring information between internal system and input-output devices.

Modes of I/O Data Transfer

- Data transfer between the central unit and I/O devices can be handled in generally three types of modes which are given below:
  - **Programmed I/O**
  - **Interrupt Initiated I/O**
  - **Direct Memory Access**

Programmed I/O

- Programmed I/O instructions are the result of I/O instructions written in computer program. Each data item transfer is initiated by the instruction in the program.
- Usually the program controls data transfer to and from CPU and peripheral. Transferring data under programmed I/O requires constant monitoring of the peripherals by the CPU.

Interrupt Initiated I/O

- In the programmed I/O method the CPU stays in the program loop until the I/O unit indicates that it is ready for data transfer. This is time consuming process because it keeps the processor busy needlessly.
- This problem can be overcome by using interrupt initiated I/O. In this when the interface determines that the peripheral is ready for data transfer, it generates an interrupt. After receiving the interrupt signal, the CPU stops the task which it is processing and service the I/O transfer and then returns back to its previous processing task.

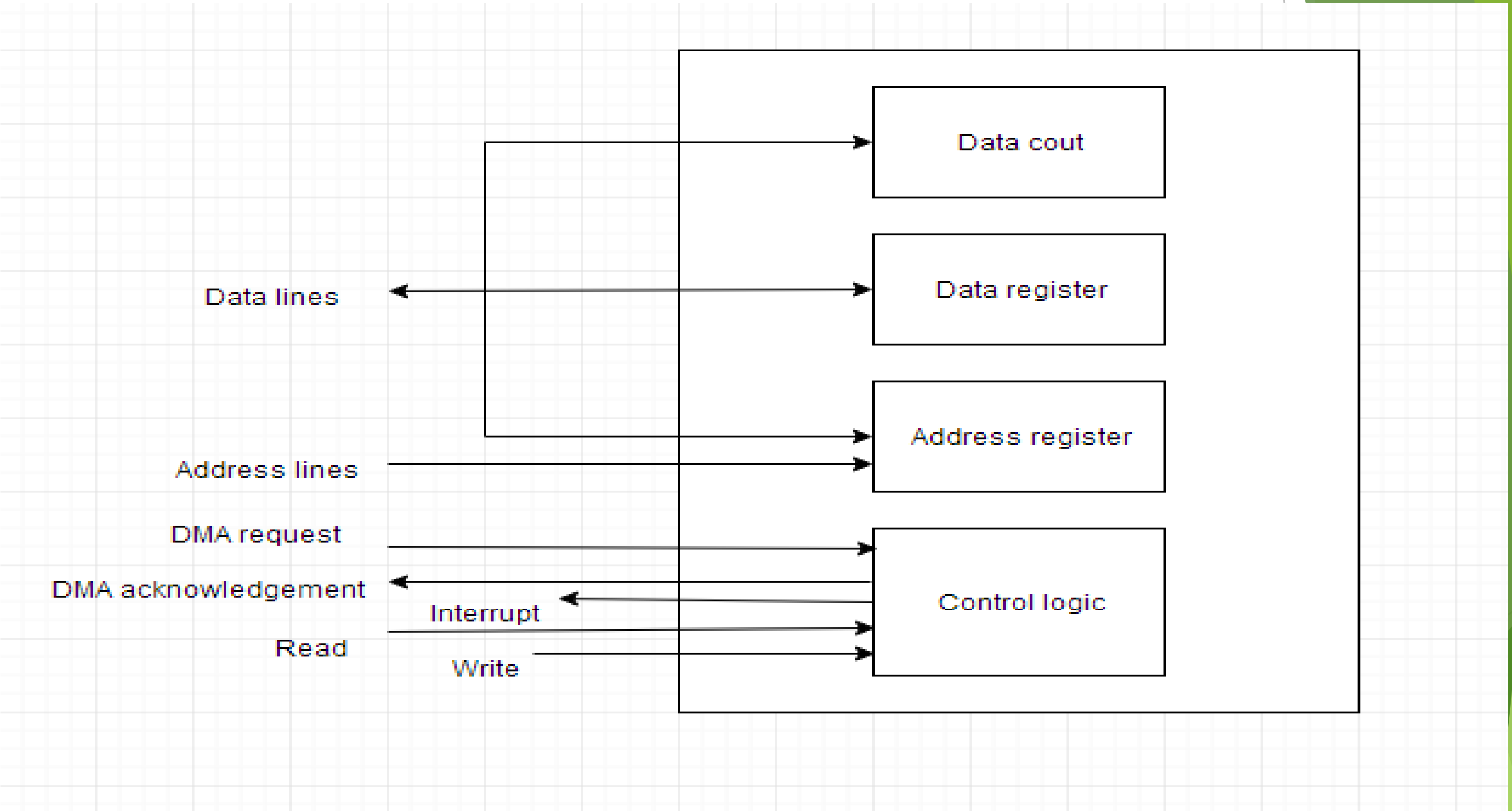
Direct Memory Access (DMA)

- Removing the CPU from the path and letting the peripheral device manage the memory buses directly would improve the speed of transfer. This technique is known as **DMA.**

- In this, the interface transfer data to and from the memory through memory bus. A DMA controller manages to transfer data between peripherals and memory unit.

- Many hardware systems use DMA such as disk drive controllers, graphic cards, network cards and sound cards etc. It is also used for intra chip data transfer in multicore processors. In DMA, CPU would initiate the transfer, do other operations while the transfer is in progress and receive an interrupt from the DMA controller when the transfer has been completed.

Direct Memory Access



Input/Output Processor

- An **Input-Output Processor (IOP)** is a processor with direct memory access capability. In this, the computer system is divided into a memory unit and number of processors.

- Each IOP controls and manage the input-output tasks. The IOP is similar to CPU except that it handles only the details of I/O processing. The IOP can fetch and execute its own instructions. These IOP instructions are designed to manage I/O transfers only.

Input/Output Processor

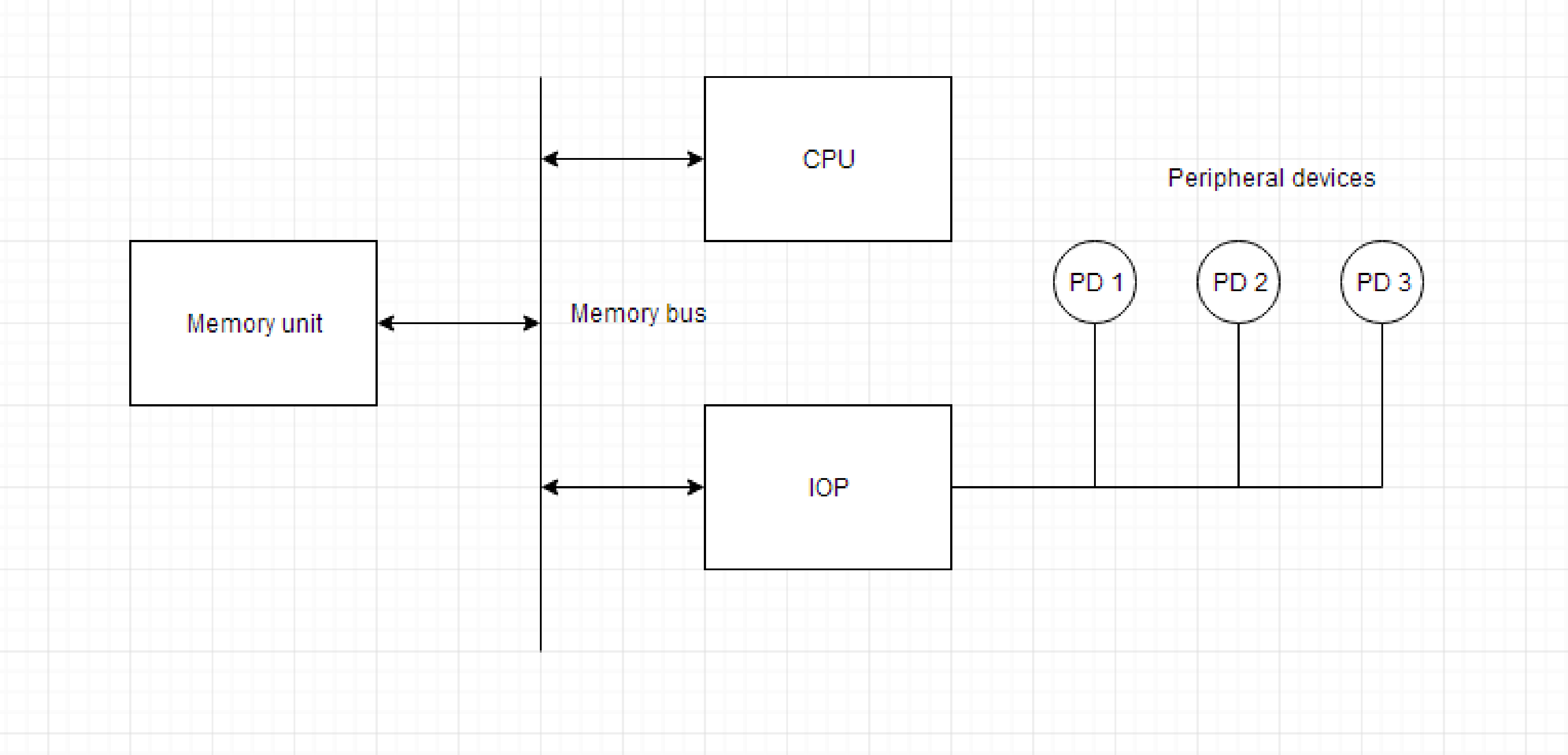
- Below is a block diagram of a computer along with various I/O Processors. The memory unit occupies the central position and can communicate with each processor.

- The CPU processes the data required for solving the computational tasks. The IOP provides a path for transfer of data between peripherals and memory. The CPU assigns the task of initiating the I/O program.

- The IOP operates independent from CPU and transfer data between peripherals and memory.

Input/Output Processor

- The communication between the IOP and the devices is similar to the program control method of transfer. And the communication with the memory is similar to the direct memory access method.
- In large scale computers, each processor is independent of other processors and any processor can initiate the operation.
- The CPU can act as master and the IOP act as slave processor. The CPU assigns the task of initiating operations but it is the IOP, who executes the instructions, and not the CPU. CPU instructions provide operations to start an I/O transfer. The IOP asks for CPU through interrupt.
- Instructions that are read from memory by an IOP are also called commands to distinguish them from instructions that are read by CPU. Commands are prepared by programmers and are stored in memory. Command words make the program for IOP. CPU informs the IOP where to find the commands in memory.



Input/Output Processor

Need for Input/Output Processor

- 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 communication link is to resolve the differences that exist between the central computer and each peripheral.

Need for Input/Output Processor

- The Major Differences are:-
  - Peripherals are electromechanically and electromagnetic devices and CPU and memory are electronic devices. Therefore, a conversion of signal values may be needed.
  - The data transfer rate of peripherals is usually slower than the transfer rate of CPU and consequently, a synchronization mechanism may be needed.
  - Data codes and formats in the peripherals differ from the word format in the CPU and memory.
  - The operating modes of peripherals are different from each other and must be
  - controlled so as not to disturb the operation of other peripherals connected to the CPU.

The Memory System

Memory Management

Memory management

- Operating system is concerned with transferring programs and data between secondary storage and main memory.
- Operating system needs memory routines in addition to the other routines.
- Operating system routines are assembled into a virtual address space called system space.
- System space is separate from the space in which user application programs reside.
  - This is user space.
- Virtual address space is divided into one

  system space + several user spaces.

Memory management (contd..)

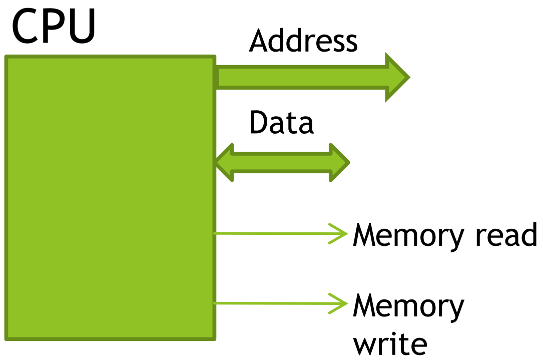
- Recall that the Memory Management Unit (MMU) translates logical or virtual addresses into physical addresses.
- MMU uses the contents of the page table base register to determine the address of the page table to be used in the translation.
  - Changing the contents of the page table base register can enable us to use a different page table, and switch from one space to another.
- At any given time, the page table base register can point to one page table.
  - Thus, only one page table can be used in the translation process at a given time.
  - Pages belonging to only one space are accessible at any given time.

Memory management (contd..)

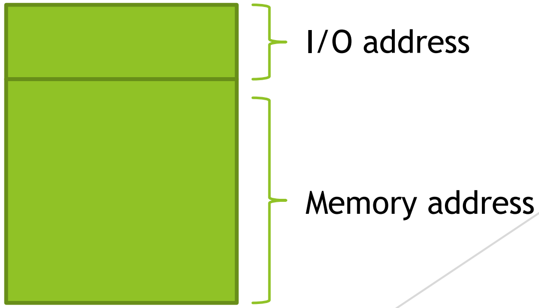
- When multiple, independent user programs coexist in the main memory, how to ensure that one program does not modify/destroy the contents of the other?
- Processor usually has two states of operation:
  - Supervisor state.
  - User state.
- Supervisor state:
  - Operating system routines are executed.
- User state:
  - User programs are executed.
  - Certain privileged instructions cannot be executed in user state.
  - These privileged instructions include the ones which change page table base register.
  - Prevents one user from accessing the space of other users.

Memory Mapped I/O

- There is no specific input 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.
- Each interface is organized as set of registers(read & write in normal address space).
- Memory mapped I/O can use memory type instructions to access I/O data.
- It allows the computer to use the same instructions for either i/o transfer or for memory transfers.
- The advantage is that the load and store instructions used for reading and writing from memory can be used to input and output data from I/O registers.



a) CPU Signals



b) Address space division

Difference between Memory mapped I/O and I/O mapped I/O

	Memory Mapped Input/Output	Input/Output Mapped Input/Output
1.	Each port is treated as a memory location.	Each port is treated as an independent unit.
2.	CPU’s memory address space is divided between memory and input/output ports.	Separate address spaces for memory and input/output ports.
3.	Single instruction can transfer data between memory and port.	Two instruction are necessary to transfer data between memory and port.
4.	Data transfer is by means of instruction like MOVE.	Each port can be accessed by means of IN or OUT instructions.

Program Controlled I/O

- Program controlled I/O is one in which the processor repeatedly checks a status flag to achieve the required synchronization between processor & I/O device.
- The processor polls the device.
- It is useful in small low speed systems where hardware cost must be minimized.
- It requires that all input/output operators be executed under the direct control of the CPU.
- The transfer is between CPU registers(accumulator) and a buffer register connected to the input/output device.
- The i/o device does not have direct access to main memory.
- A data transfer from an input/output device to main memory requires the execution of several instructions by the CPU, including an input instruction to transfer a word from the input/output device to the CPU and a store instruction to transfer a word from CPU to main memory.
- One or more additional instructions may be needed for address communication and data word counting.

Typical Program Controlled Instructions

Name	Mnemonic
Branch	BR
Jump	JMP
Skip	SKP
Call	CALL
Return	RET
Compare	CMP
Test(by ADDing)	TST

Interrupts

- A Suspension of a process such as the execution of a computer program, caused by an event external to that process, and performed in such a way that the process can be resumed. A way to improve processor utilization.

Need For Interrupts?

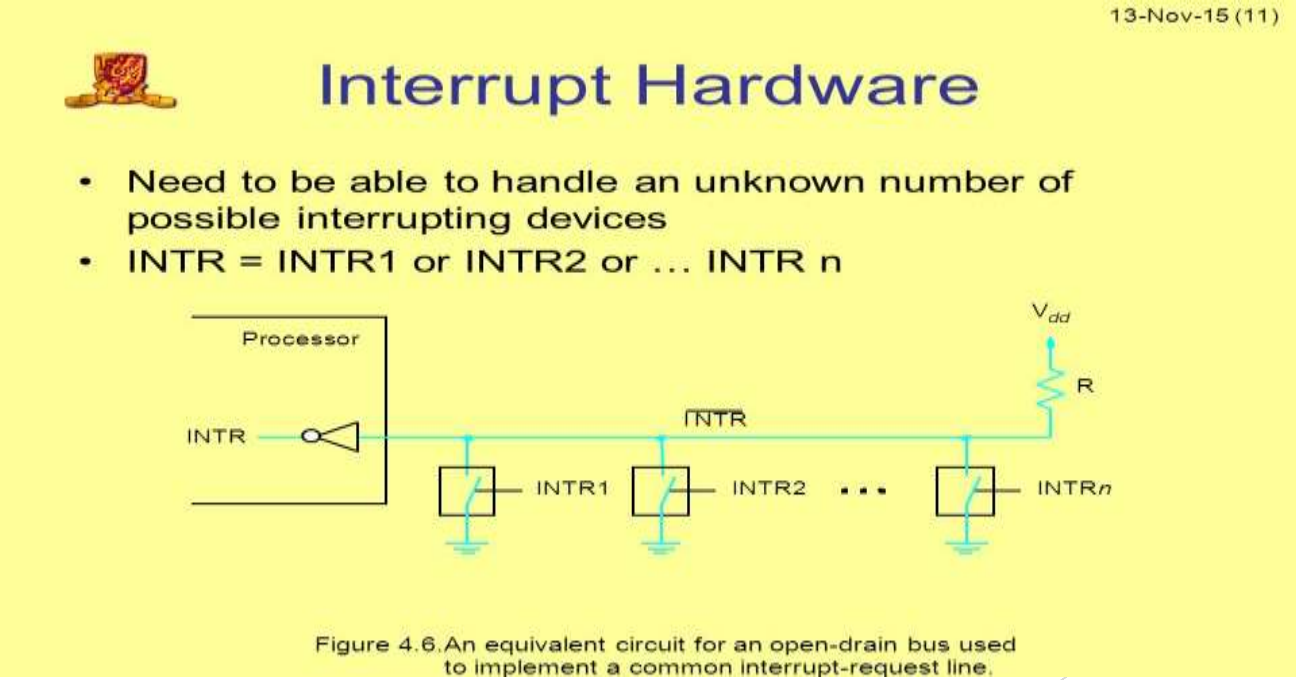
- The OS is a reactive program

1. When you give some input
2. It will perform computations
3. Produces output BUT
4. Meanwhile you can interact with the system by interrupting the running process or
5. You can stop and start another process.

- This reactive ness is due to interrupts
- Modern Operating Systems Are Interrupt driven

Interrupt Hardware

- I/O device request an interrupt by activating a bus line called interrupt-request.
- A single interrupt request line may be used to serve n devices.
- All devices are connected to interrupt request line via



13-Nov-15 (11)

Interrupt Hardware

- Need to be able to handle an unknown number of possible interrupting devices
- INTR = INTR1 or INTR2 or ... INTR n

Figure 4.6.An equivalent circuit for an open-drain bus used to implement a common interrupt-request line.

Interrupt Hardware Cont….

- To request an interrupt, a device closes its associated switch.
- If all interrupt-request signals INTR1 to INTRn are inactive, that is, if all switches are open, the voltage on the interrupt request line will equal to Vdd.
- This is an inactivate state of the line.
- When a device requests an interrupt by closing its switch, the voltage on the line drops to 0, causing the interrupt- request signal INTR received by the processor to go to 1.
- If closing of one (or) more switches that cause the line value to drop to 0, the value of logical OR of the request from individual devices, that is

  INTR=INTR1+INTR2+INTR3..............

- Use the complement form of INTR to name of the interrupt signal on the common line because this signal is active in the low voltage state

Enabling and Disabling Interrupts

- A processor has the facility to enable and disable interrupts as desired.
- When a device request the interrupt during the processor service for another interrupt, the result cause the processor enter into the infinite loop.
- This can be handled by the following 2 ways:
  - The processor ignore the interrupt request line(INTR) until the Interrupt Service Routine(ISR) is completed.
  - This can be done by using interrupt-Disable as first instruction and interrupt-Enable as the last instruction.

Enabling and Disabling Interrupts Cont…

- The second option is processor automatically disable interrupts before starting the execution of the ISR.
- The status register PS stored in the stack with PC value.
- The processor set this register bit 1 when the interrupt accept and when a return instruction is executed, the contents of the PS are cleared (0)and stored in the stack again.

Handling Multiple Devices

- When the number of devices initiating interrupts.
- For example, device X may request an interrupt while an interrupt caused by device Y is being serviced.
- Hence all the device using the common interrupt line.
- Additional information require to identify the device that activated the request.
- When the two devices activated the line at the same time, we must break up the tie and chose one the device request among two. Some scheme should be used by the processor.

Handling Multiple Devices Contd…

1. Polling Scheme

   The device that raises the interrupt will set one of the bit (IRQ) in status register to the processor will poll the devices to find which raised an interrupt first.

   Disadvantage:

   Time spend in interrogating the IRQ bits of the devices that may not be requesting any service.

2. Vectored Interrupts

   To reduce the time involved in the polling scheme, a device requesting an interrupt may identify itself directly to the processor. A device can send a special code to the processor over the bus. The code is used to identify the device. If the interrupt produces a CALL to a predetermined memory location, which is the starting address of ISR, then that address is called vectored address and such interrupts are called vectored interrupts.

Handling Multiple Devices Contd…

3. Interrupt priority

When a interrupt arrives from one (or) more devices simultaneously, the processor has to decide which request should be serviced first. • The processor takes this decision with the help of interrupt priorities. • The processor accepts interrupt request having highest priority. • Each request assign a different priority level. • The request received from the interrupt request line are sent to a priority arbitration circuit in the processor. • The request is accepted only if it has a higher priority level than that currently assigned to the processor.

4. Controlling device request

The processor allow only the input / output devices requested(interrupt), that are being used by a given program. • Other devices should not be allowed to generate interrupt requests even though they are ready to transfer the data. • Hence, we need a mechanism in the interface circuits of individual devices to control whether the device is allowed to generate an interrupt request. •

Two mechanism for control request:

1. One is at the device end- interrupt enable bit in the control register(IRQ).

2. Processor end- enable bit in the program status register(PS) or priority structure determine whether a given interrupt request will be accepted.

Thank You