

Chapter 7: Input/Output

- ❖ **Introduction**
- ❖ **I/O interface**
 - ❖ **Accessing I/O Devices**
 - **Memory-mapped I/O**
 - **Isolated I/O**
- ❖ **Input/output mechanism**
 - **Programmed I/O**
 - **Interrupts**
 - **Direct Memory Access**

1. Input/output (I/O)

- The devices that Computers Interact with Outside World
- Otherwise, how will we ever tell a computer what to do or exploit the results of its work?
- Computers without I/O are not useful

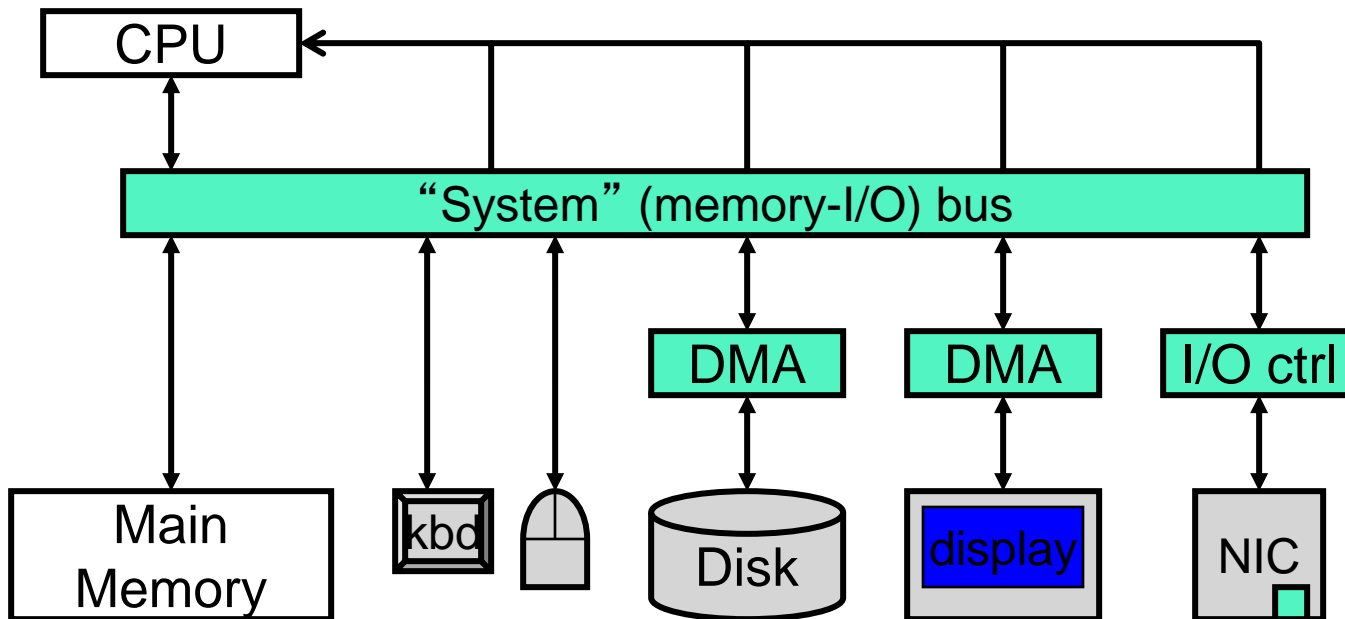
1.Input /output

- A part from the processor and a set of memory modules, the third key element of a computer system is a set of input-output subsystem
- provides an efficient mode of communication between the central system and the outside environment.

1. I/O System

A computer system

- CPU, including cache(s)
- Memory (DRAM)
- **I/O peripherals**: disks, input devices, displays, network cards, ...
With built-in or separate I/O (or DMA) controllers. All connected by a **system bus**



1. I/O: Control + Data Transfer

I/O devices have two ports

Control: commands and status reports

How we tell I/O what to do

How I/O tells us about itself

Control is the tricky part (especially status reports)

Data

Labor-intensive part

“Interesting” I/O devices do data transfers (to/from memory)

Display: video memory → monitor

Disk: memory ↔ disk

Network interface: memory ↔ network

1. Operating System (OS) Plays a Big Role

I/O interface is typically under OS control

- User applications access I/O devices indirectly

Why?

- Device **drivers** are “programs” that OS uses to manage devices

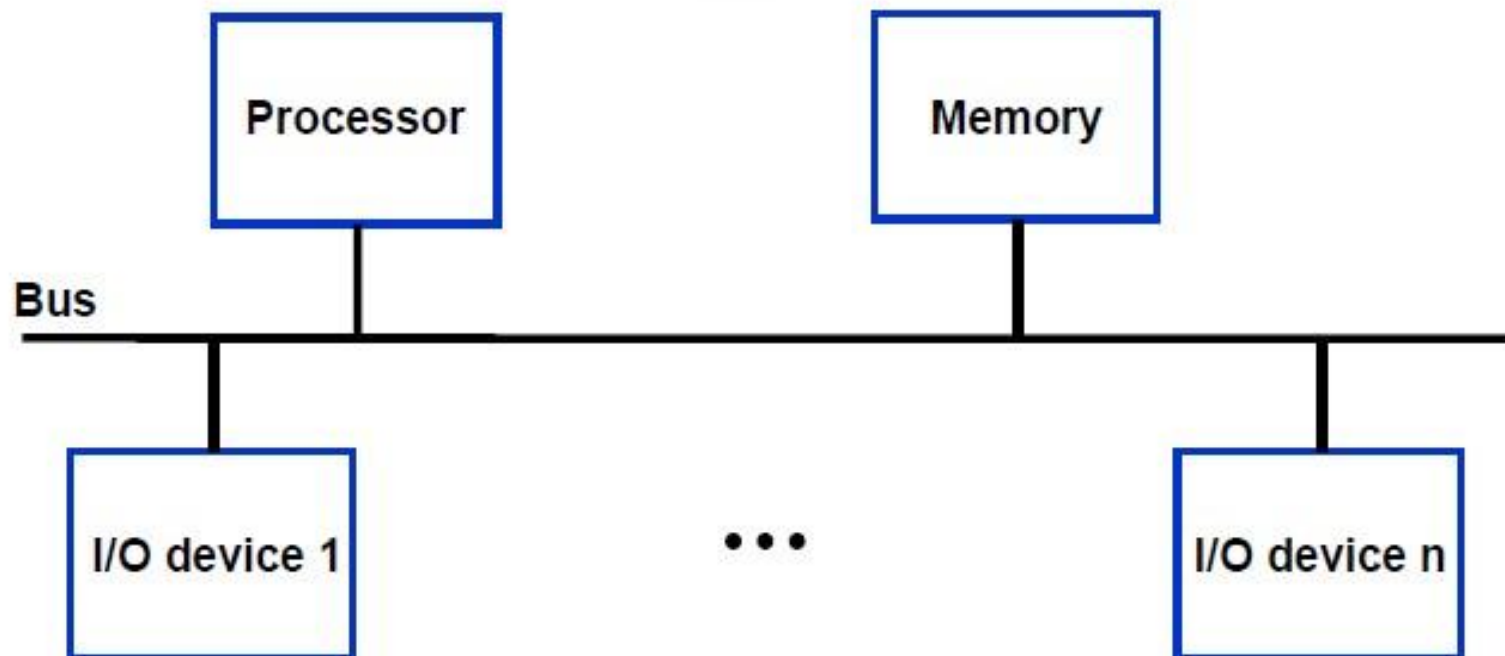
1. External Devices classification

- Human readable (human interface)
 - Monitor, printer, keyboard, mouse
- Machine readable
 - Disk, tape, sensors
- Communication
 - Modem
 - Network Interface Card (NIC)

Accessing I/O Devices

➤ Single-bus structure

- ◆ The bus enables all the devices connected to it to exchange information
- ◆ Typically, the bus consists of three sets of lines used to carry address, data, and control signals
- ◆ Each I/O device is assigned a unique set of addresses



Accessing I/O Devices

- Bus consists of 3 set of lines : ***Address, Data, Control***
- Processor places a particular address (unique for an I/O Dev.) **on address lines**
- Device which recognizes this address responds to the commands issued on the **Control lines**
- Processor requests for either **Read / Write**
- The data will be placed **on Data lines**

Input/Output Problems

- Wide variety of peripherals
 - Delivering different amounts of data
 - At different speeds
 - In different formats
- Some slower than CPU and RAM
- Need I/O modules

Device	Behavior	Partner	Data rate (Mbit/sec)
Keyboard	input	human	0.0001
Mouse	input	human	0.0038
Voice input	input	human	0.2640
Sound input	input	machine	3.0000
Scanner	input	human	3.2000
Voice output	output	human	0.2640
Sound output	output	human	8.0000
Laser printer	output	human	3.2000
Graphics display	output	human	800.0000–8000.0000
Modem	input or output	machine	0.0160–0.0640

input/output interface(i/o modules)

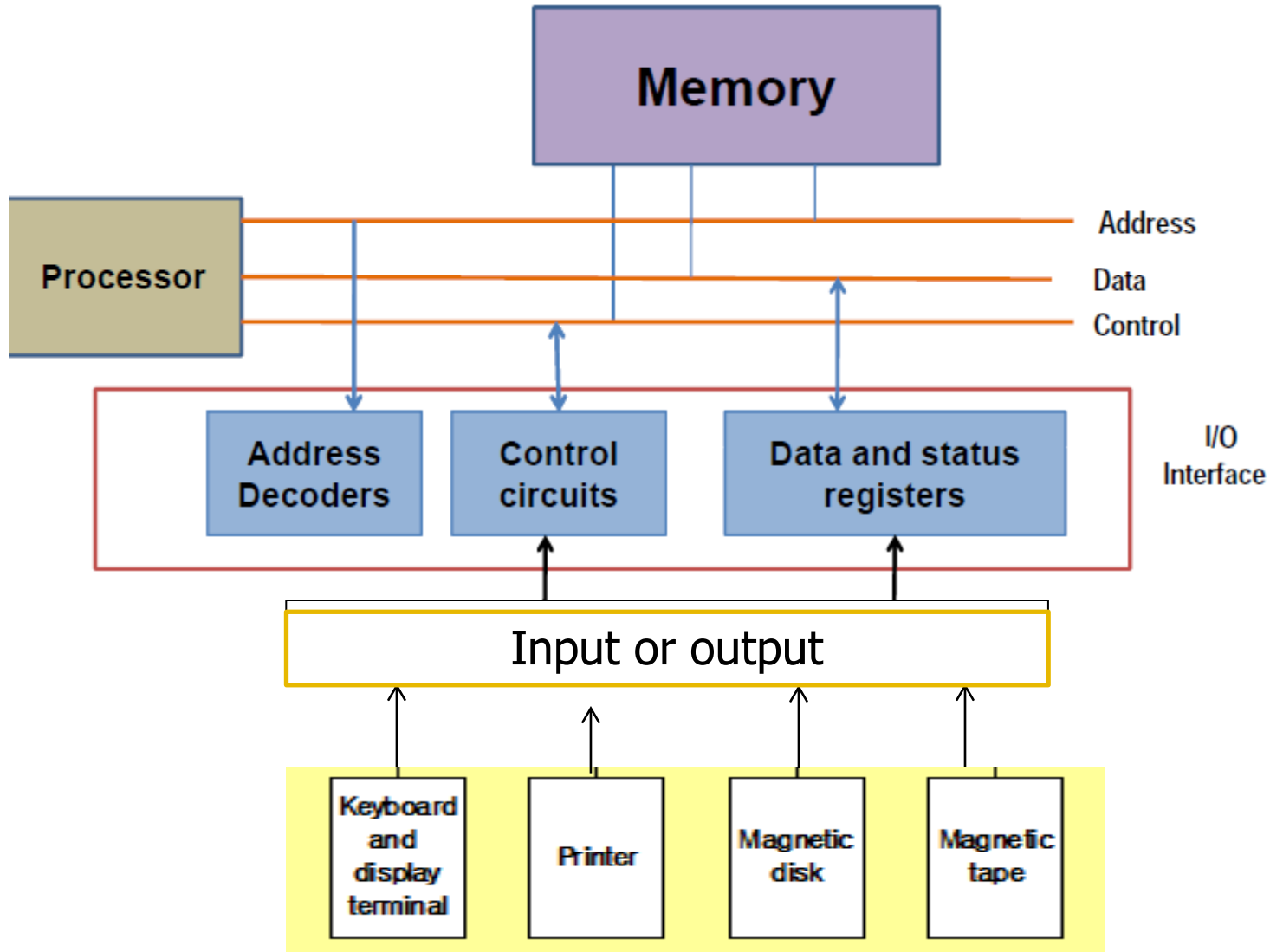
❖ Interface

- Special hardware components between the CPU and peripherals
- Supervise and Synchronize all input and output transfers

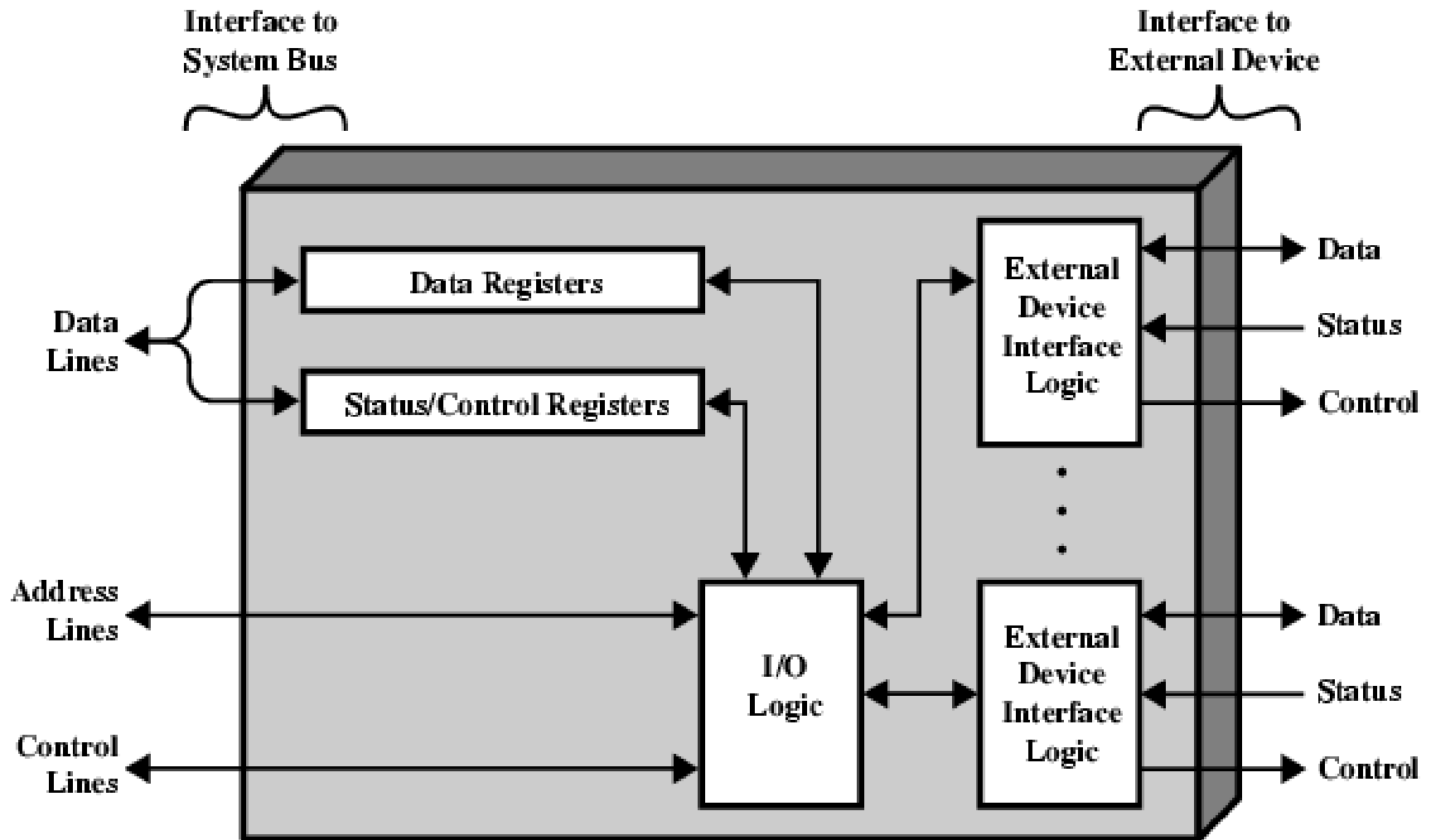
❖ I/O module has two major functions

- Interface to CPU and Memory
- Interface to one or more peripherals

input/output interface



I/O Interface(Circuit)



I/O Interface(Circuit)

- **Address Decoder**
- **Control Circuits**
- **Data registers**
- **Status registers**

I/O Interface functions:

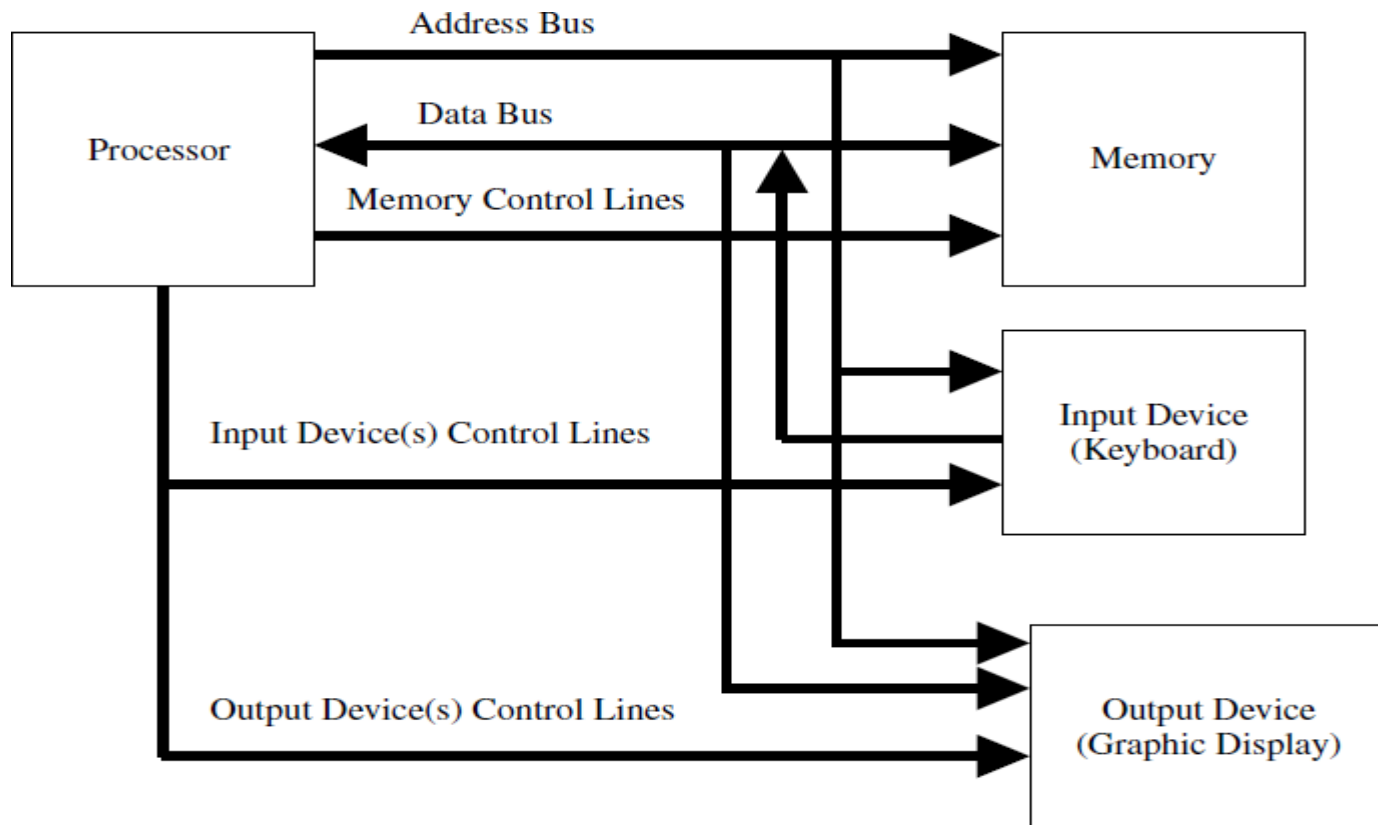
- **Control and timing** :I/O module includes control and timing to coordinate the flow of traffic between internal resources and external devices
- **Communication:**
 - **processor**
 - Command decoding
 - Data
 - Status reporting
 - Address recognition
 - **Device**
- **Data Buffering:**
- **Error Detection:**

2. I/O Access Steps

- CPU checks (interrogates) I/O module device status
- I/O module returns status
- If ready, CPU requests data transfer by sending a command to the I/O module
- I/O module gets a unit of data (byte, word, etc.) from device
- I/O module transfers data to CPU
- Variations of these steps for output.

Accessing I/O Devices

- The CPU Use one common bus for both memory and I/O but have separate control lines for each
- Each I/O device is assigned a unique address.



Two ways of addressing to I/O devices

❖ **Isolated I/O** : i/o device are assign address space separate memory address space.

❖ **MEMORY Mapped I/O** : used same memory space for both MM and i/o device.

Isolated I/O

- I/O devices are assigned particular addresses, isolated from the address space assigned to the memory.
- There is the need to have special input and output instructions in the processor instruction set.

example: input 2000.

- Isolated I/O arrangement is mostly adopted by Intel.

Memory mapped I/O

- Devices and memory share the memory address space.
- I/O looks just like memory read/write operation
 - Ex: Read #6
- No special commands for I/O
- Result in reduction in the available memory address space.
 - example: MM of 1024 address , 0-516 used by MM, while 517-1023 used by i/o devices.
- The memory-mapped I/O has been mostly adopted by Motorola.

3.Three techniques for I/O operations:

- Programmed
- Interrupt driven
- Direct Memory Access (DMA)

1. Programmed I/O

- CPU has direct control over I/O
 - Sensing status
 - Read/write commands
 - Transferring data
- CPU waits for I/O module to complete operation
- Wastes CPU time.

Programmed I/O Basic Operation

- CPU requests I/O operation(ready or write)
- I/O module performs operation
- I/O module sets status bits
- CPU checks status bits periodically
- I/O module does not inform CPU directly
- I/O module does not interrupt CPU
- CPU may wait or come back later

2. Interrupt Driven I/O

- Overcomes CPU waiting
- No repeated CPU checking of device
- I/O module interrupts when ready.

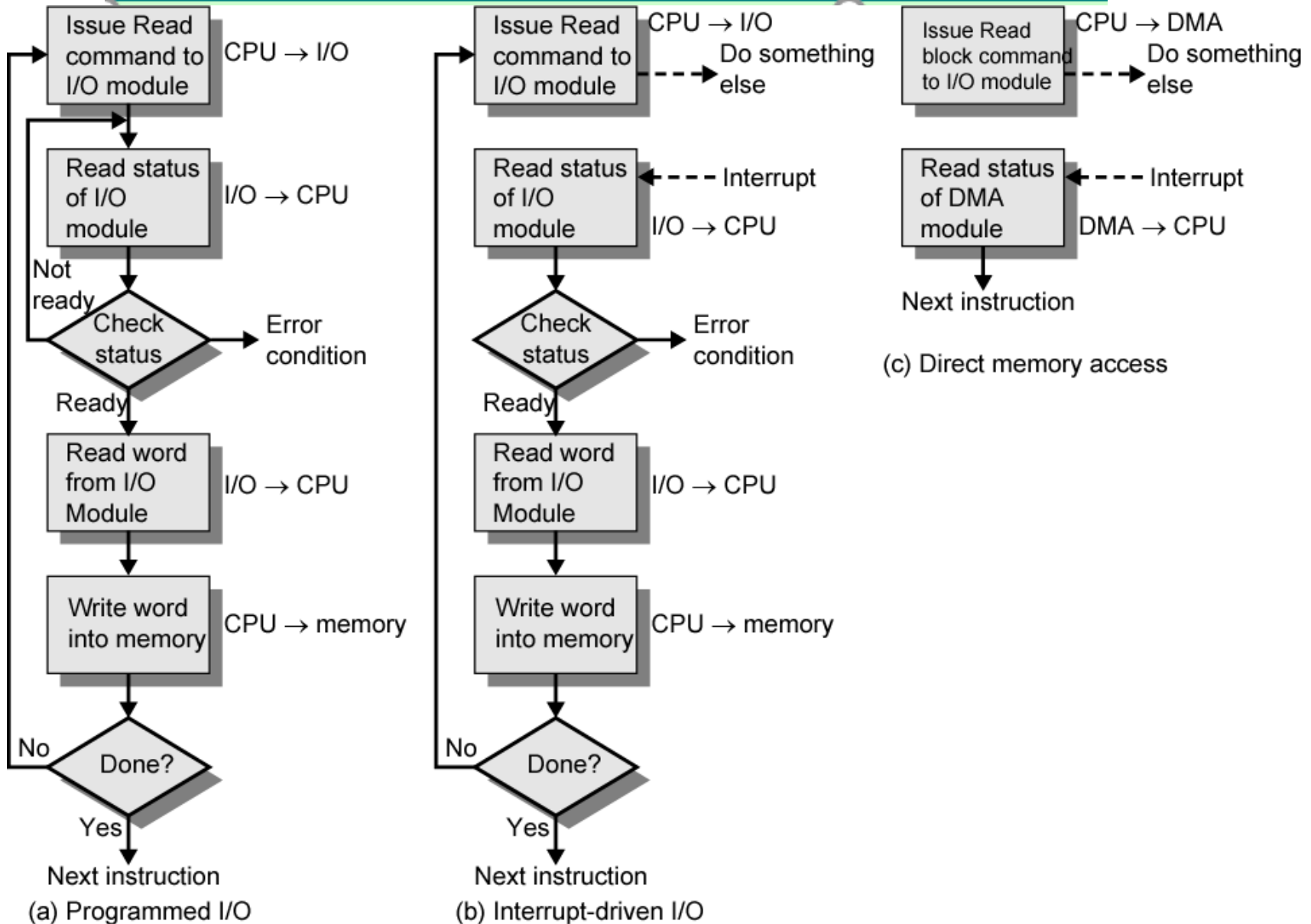
Interrupt Driven I/O Basic Operation

- CPU issues read command
- I/O module gets data from peripheral whilst CPU does other work
- I/O module interrupts CPU
- CPU requests data
- I/O module transfers data

CPU Viewpoint

- Issue read command
- Do other work
- If interrupted:-
 - Save context (stack registers)
 - Process interrupt
 - Fetch data & store
 - Resume the stopped work

Three techniques for I/O operations



Handling Multiple Interrupt

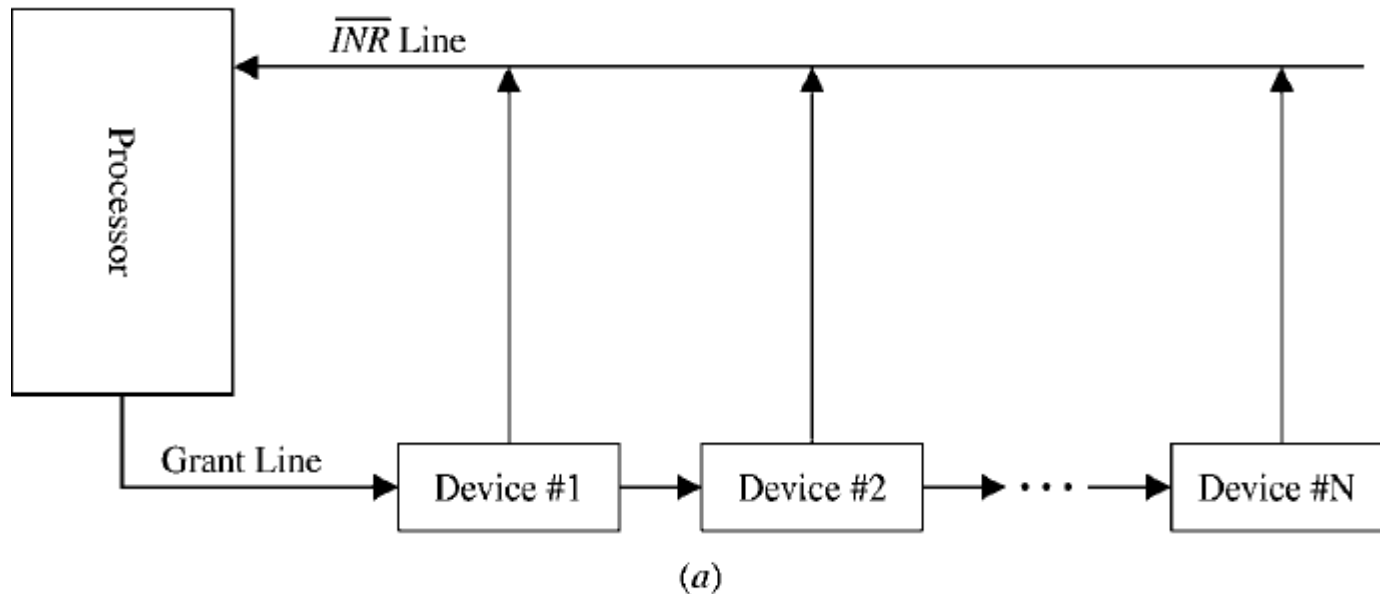
- Computers are provided with interrupt hardware CALLED interrupt lines to the processor.
- CPU have the ability to handle simultaneous interrupt requests and to recognize the interrupting device.
- Three basic schemes are used:
 - ❖ Polling Scheme
 - ❖ daisy chain bus arbitration(DCBA)
 - ❖ independent source bus arbitration (ISBA).

1. Software Polling Scheme

- The IRQ (interrupt request) bit in the status register is set when a device is requesting an interrupt.
- **The Interrupt service routine** polls the I/O devices connected to the bus.
- The first device encountered with the IRQ bit set is serviced.
- Easy to implement, but too much time spent on
- checking the IRQ bits of all devices.

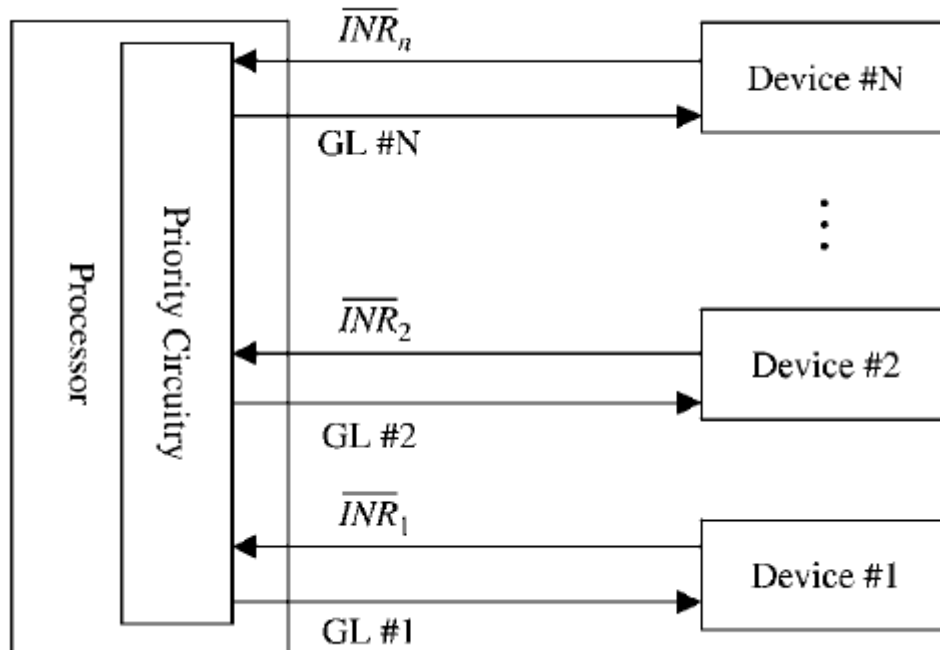
2. Daisy chain bus arbitration(DCBA)

- the DCBA arrangement gives highest priority to the device physically nearer to the processor.



3. Independent source bus arbitration (ISBA)

- each I/O device has its own interrupt request line
- each I/O device has its own grant line
- I/O device priority in the ISBA does not depend on the device location.



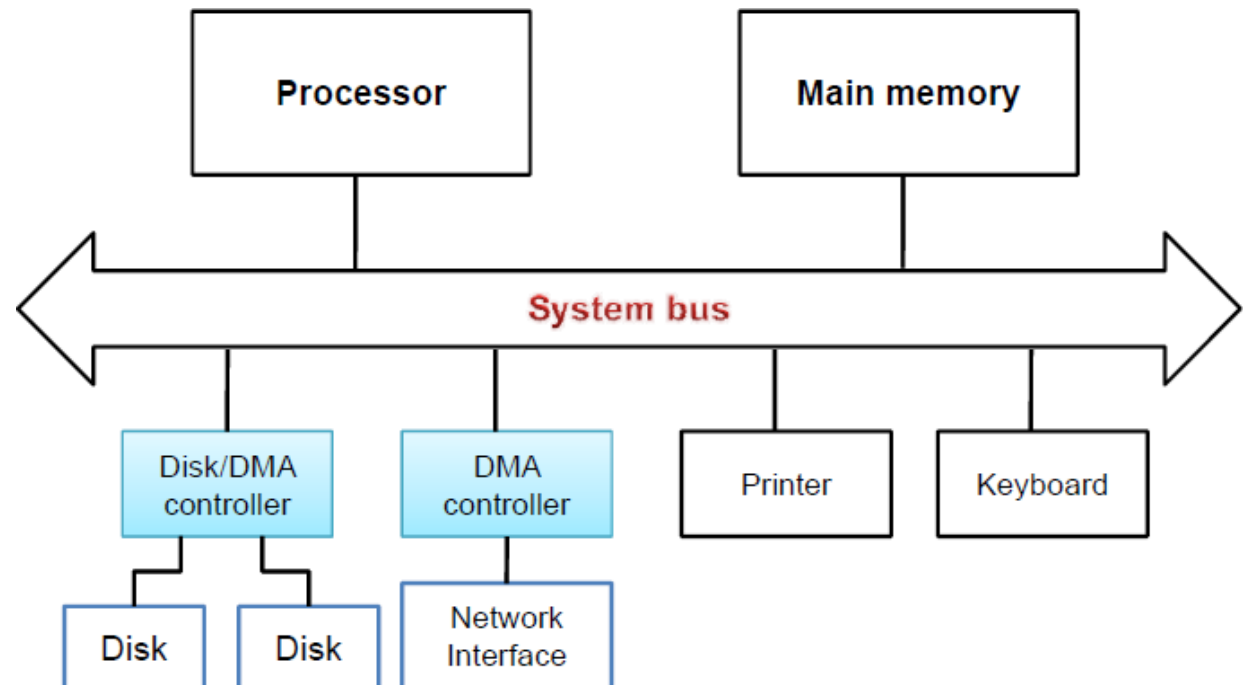
Independent source bus arbitration (ISBA)

Priority arbitration circuit Perform :

- Organizing I/O devices in a prioritized structure.
- Each of the interrupt-request lines is assigned a different priority level.
- The processor is interrupted only by a high priority device.

3. Direct Memory Access(DMA)

- To transfer large blocks of data at high Speed, between EXTERNAL devices & Main Memory, DMA approach is often used.
- DMA controller allows data transfer directly between I/O device and Memory, with minimal intervention of processor.



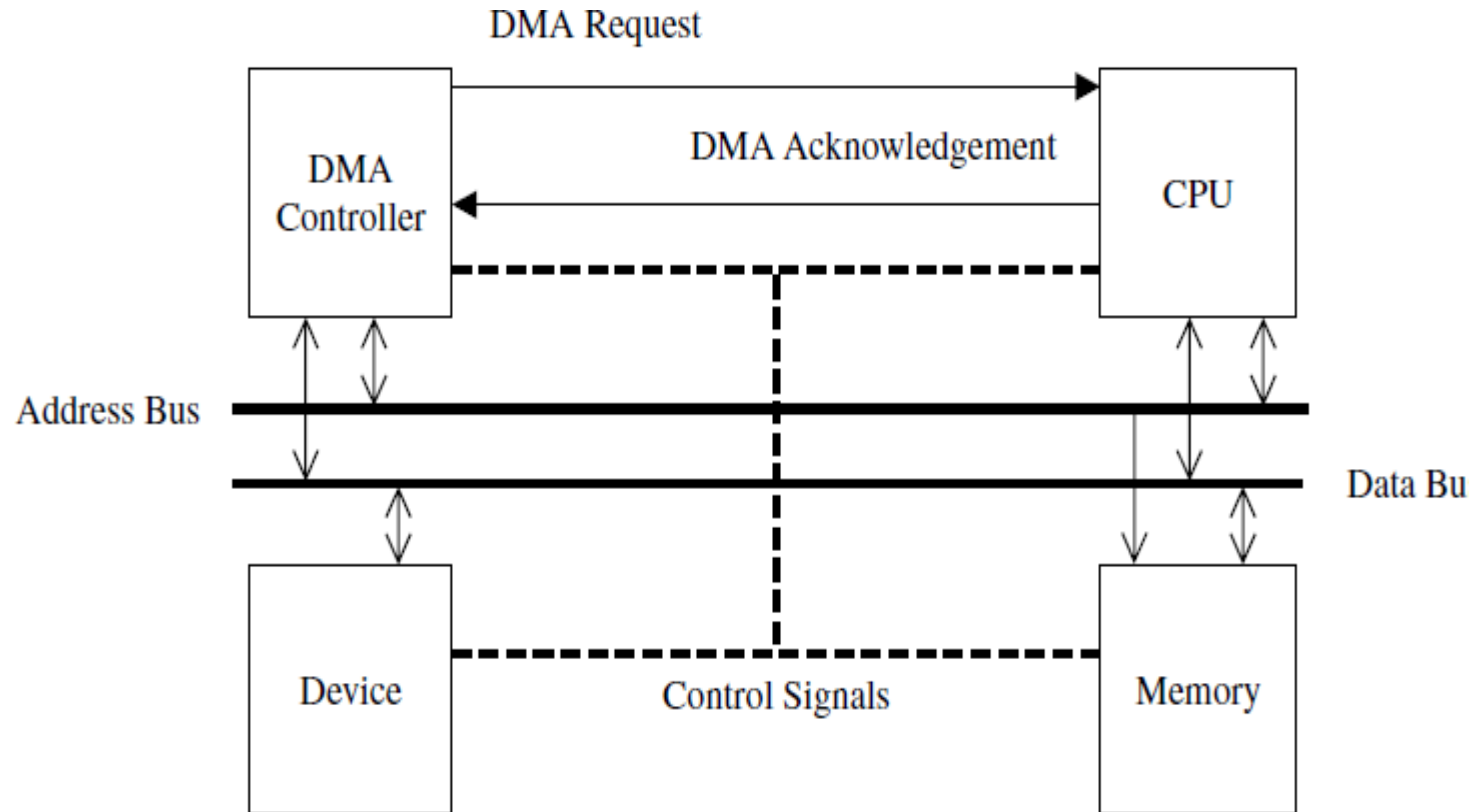
DMA

- DMA controller is control circuit that is part of the I/O device interface.
- DMA controller takes over from CPU for I/O

CPU and DMA Operation

- CPU tells DMA controller:
 - Read/Write
 - Device address
 - Starting address of memory block for data
 - Amount of data to be transferred
- CPU carries on with other work
- DMA controller deals with transfer
- DMA controller sends interrupt when finished

DMA Memory Access



DMA controller shares the CPU's memory bus

DMA operations summary

1. DMA controller initiates data transfer.
2. Data is moved (increasing the address in memory, and reducing the count of words to be moved).
3. When word count reaches zero, the DMA informs the CPU of the termination by means of an interrupt.
4. The CPU regains access to the memory bus

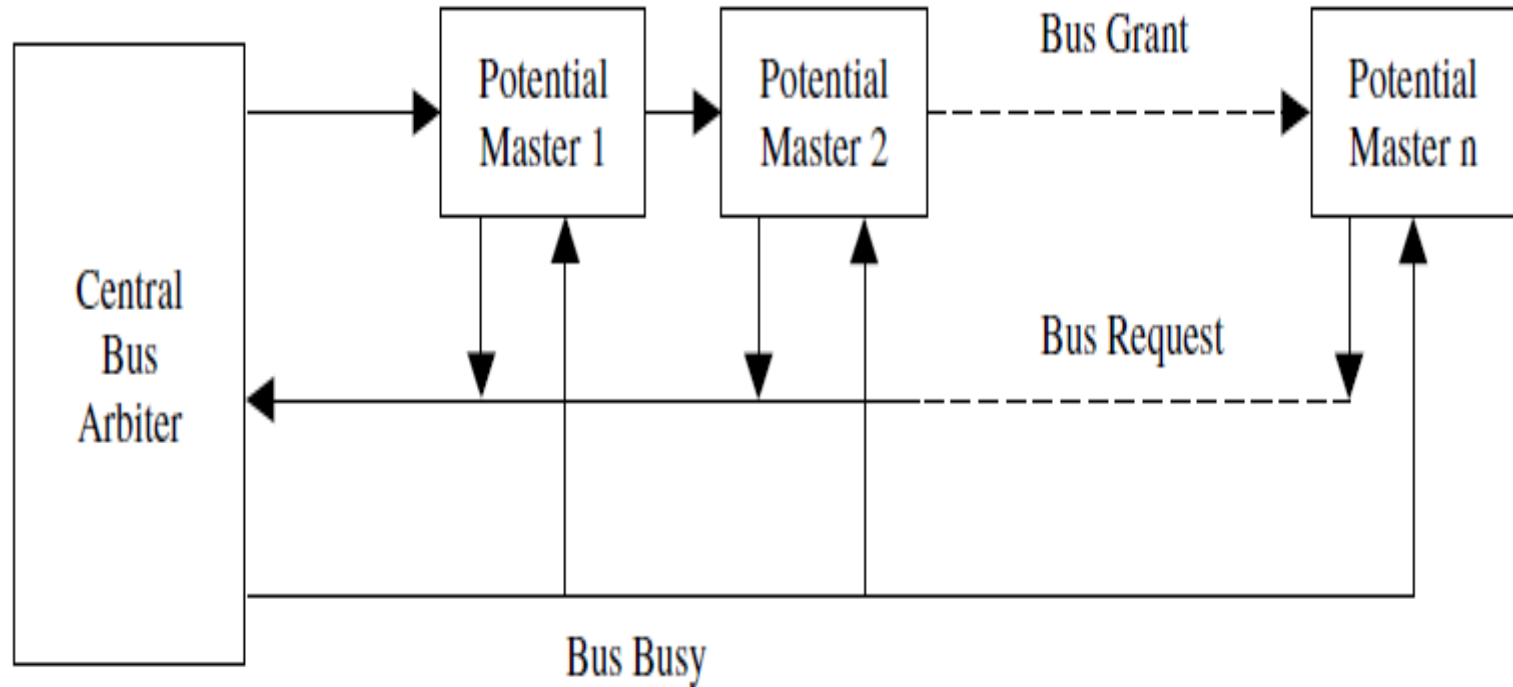
DMA Transfer mode:

- **In burst mode:** DMA controller keeps control of the bus until all the data has been transferred.
- **In single-cycle mode (cycle stealing):**
 - ❖ DMA controller relinquishes the bus after each transfer of one word.
 - ❖ Not an interrupt
 - ❖ it requires the bus again: request/acknowledge sequence be performed for every single transfer.

Bus Arbitration

- A conflict may arise if both DMA controllers try to use the bus at the same time to access the main memory.
- The device that is allowed to initiate data transfer on the bus at any given time is called the bus master.
- To resolve this problem, an arbitration procedure on bus is needed:
 - ❖ Centralized
 - ❖ distributed

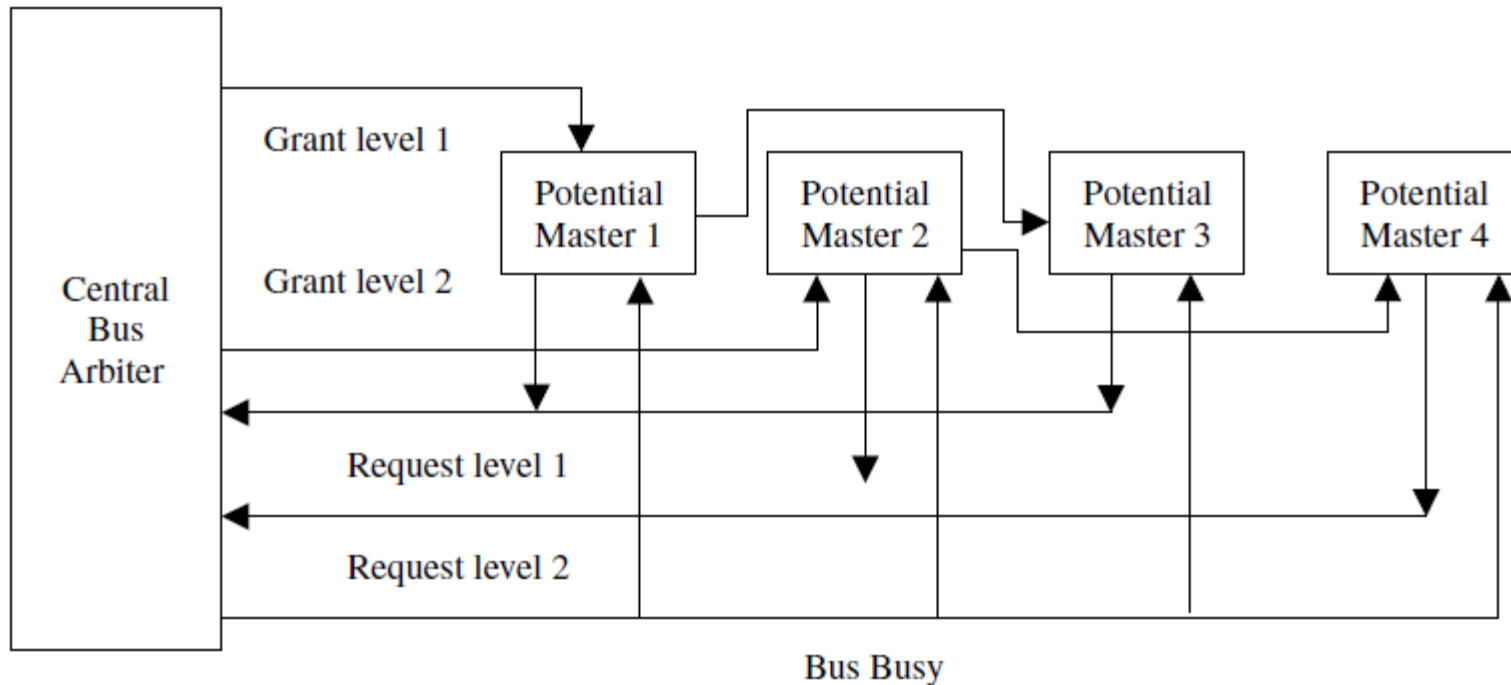
Centralized bus Arbitration



Centralized arbiter in a daisy-chain scheme

Distributed bus Arbitration

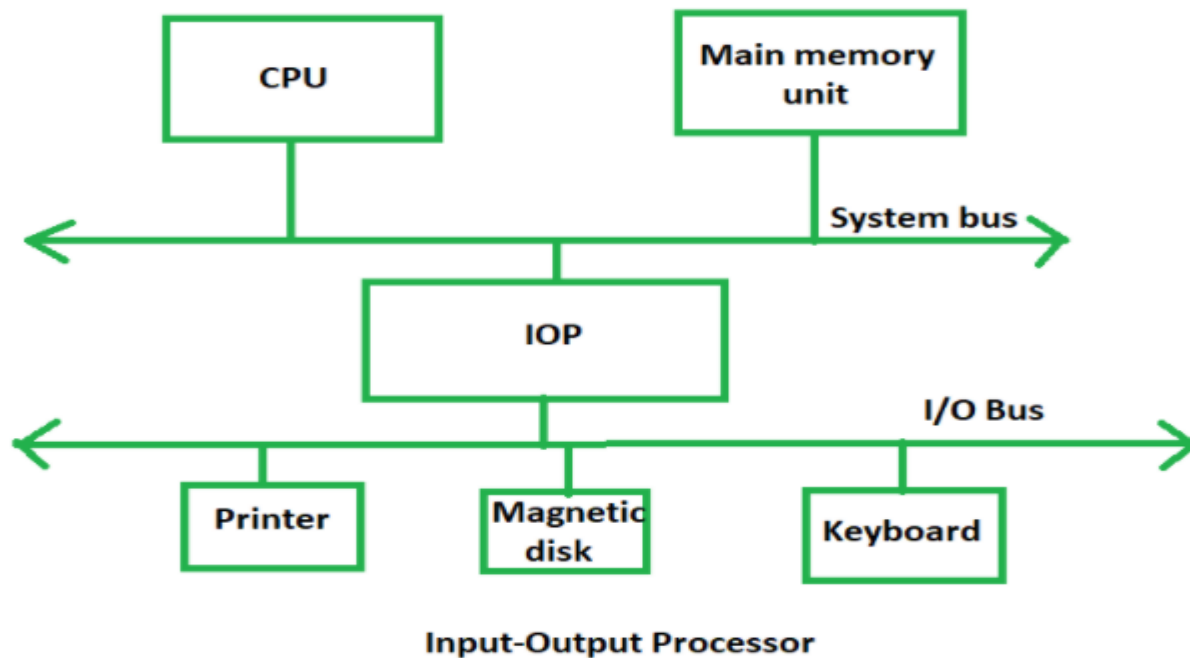
- Each potential master has a unique arbitration number.



Centralized arbiter with two priority levels (four devices)

4. Input/Output Processor or IO channel.

- An input-output processor (IOP) is a processor with direct memory access capability.
- Each IOP controls and manage the input-output tasks



Input/Output Processor

- The I/O channel represents an extension of the DMA concept.
 - An I/O channel has the ability to execute I/O instructions, which gives it complete control over I/O operations
 - 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

- 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