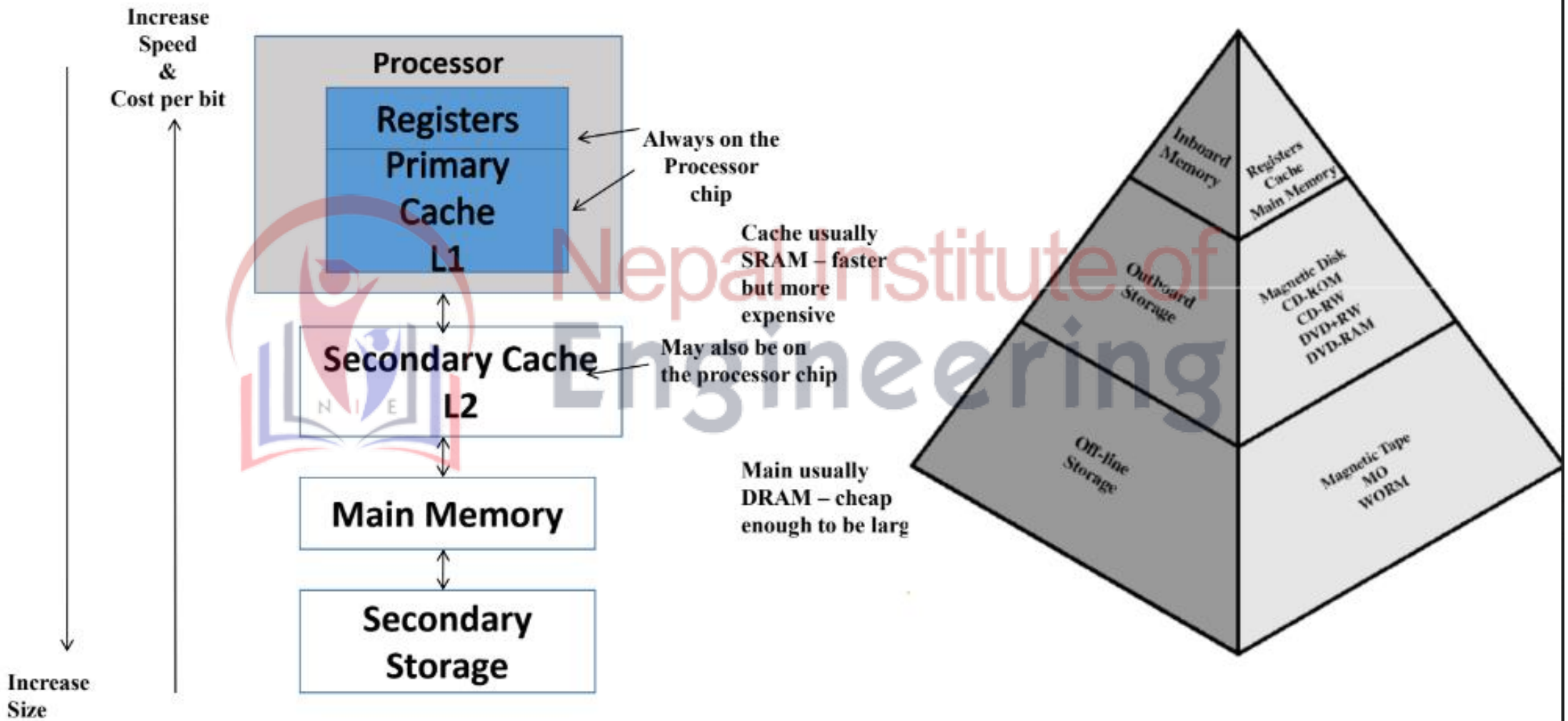
Nepal Institute of
Engineering

Memory Hierarchy

- In the design of computer system, a processor, as well as large amount of memory devices, has been used. The main problem is, these parts are expensive. So the memory organization of the system can be done by memory hierarchy.
- Memory hierarchy is an enhancement of to organize the memory such that it can minimize the access time. The memory hierarchy was developed based on a program behavior known as locality of reference.
- Memory hierarchy design is divided into two main types-
    1. External / secondary Memory- Comprising of magnetic disk, optical disk, magnetic tape i.e. peripheral storage devices which are accessible by the processor via I/O module.
    2. Internal or Primary memory- comprising of main memory & CPU registers. This is directly accessible via processor.
- Based on a speed and use memory in a computer can be divided into five hierarchies. The processor can move from one level to another level based on its requirement.
- The five hierarchies in the memory are registers, cache, main memory, magnetic discs, and magnetic tape. The first three hierarchies are volatile memories where are the last two hierarchies are non volatile.



Characteristics of Memory Hierarchy

- Performance :-
  - Previously designing of computer system was done without memory hierarchy, and the speed gap among the main memory as well as the cpu registers enhances because of the huge disparity in access time, which will cause the lower performance of the system.
  - The enhancement of this was designed in the memory hierarchy model due to system performance increase.
- Ability :-
  - The ability of memory hierarchy is the total amount of data the memory can store.
- Access Time :-
  - The access time in a memory hierarchy is the interval of time among the data availability as well as request to read or write. When we move from top to bottom inside the memory hierarchy , then access time will increase.
- Cost per bit :-
  - When we move from top to bottom inside the memory hierarchy, then cost per bit will decrease.

Interfacing

- Interface is the path for communication between two components. Interfacing is of two types:- i) memory interfacing and ii) I/O interfacing
- **Memory Interfacing**

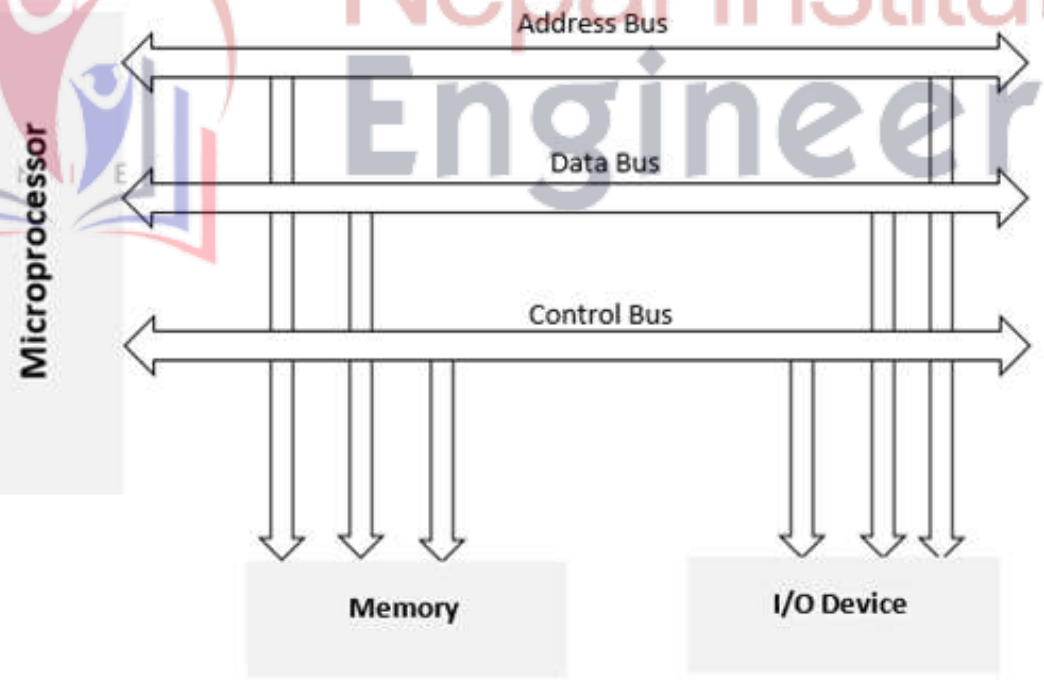
✓ When we are executing any instruction, we need the microprocessor to access the memory for reading instruction codes and the data stored in the memory.

✓Both the memory and the microprocessor requires some signals to read from and write to registers.

✓The interfacing process includes some key factors to match with the memory requirements and microprocessor signals. The interfacing circuit therefore should be designed in such a way that it matches the memory signal requirements with the signals of the microprocessor.

Interfacing

- **IO Interfacing**
- There are various communication devices like the keyboard, mouse, printer, etc. So, we need to interface the keyboard and other devices with the microprocessor by using latches and buffers. This type of interfacing is known as I/O interfacing.



8085 Interfacing Pins

- Following is the list of 8085 pins used for interfacing with other devices –
  - A15 - A8 (Higher Address Bus)
  - AD7 - AD0(Lower Address/Data Bus)
  - ALE
  - RD
  - WR
  - READY

Ways of communication-

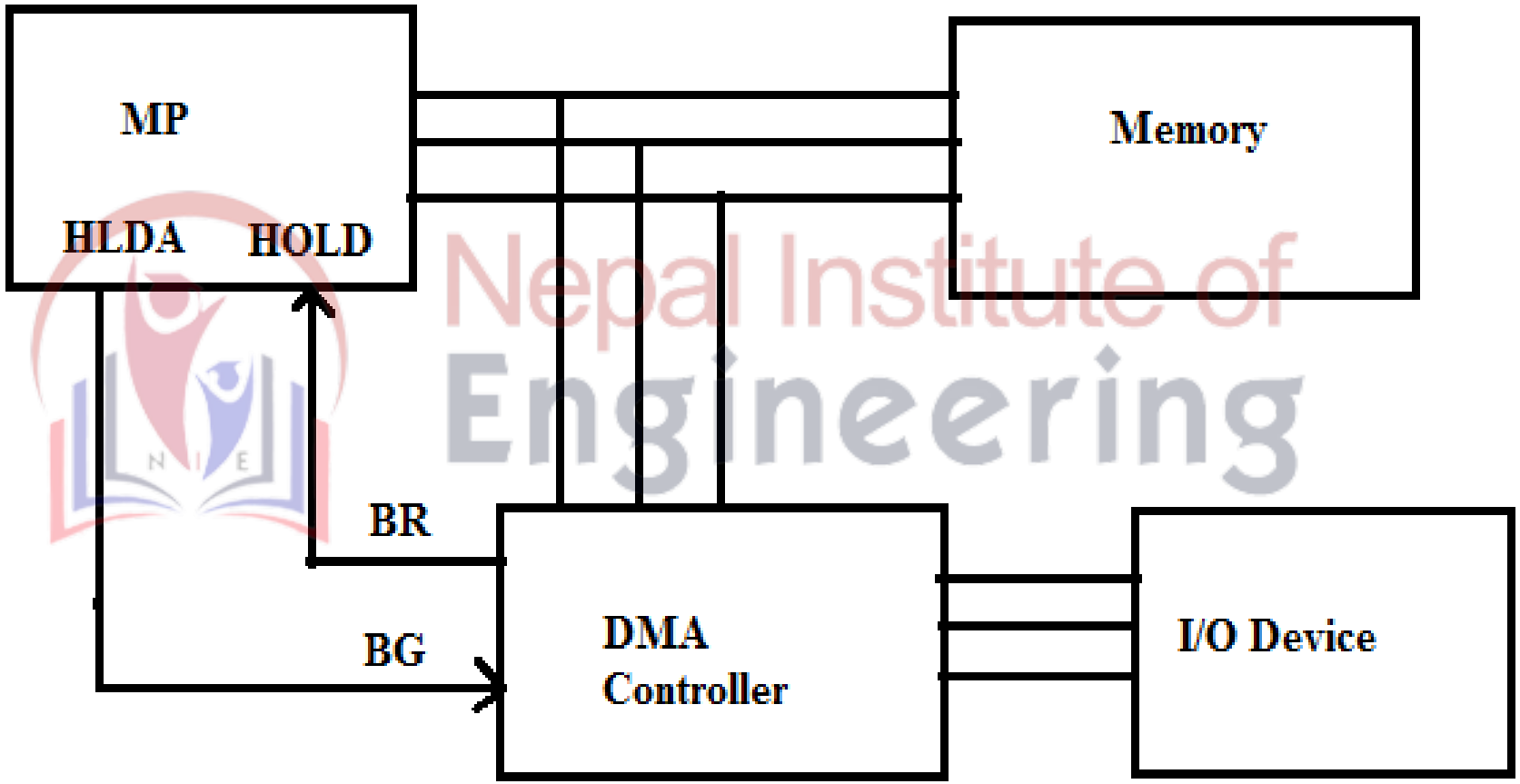
- **Microprocessor with the outside World?**

✓There are two ways of communication in which the microprocessor can connect with the outside world.

- **Serial Communication Interface –**
  - ✓ In this type of communication, the interface gets a single byte of data from the microprocessor and sends it bit by bit to the other system serially and vice-a-versa.
- **Parallel Communication Interface –**
  - ✓ In this type of communication, the interface gets a byte of data from the microprocessor and sends it bit by bit to the other systems in simultaneous (or) parallel fashion and vice-a-versa.

8257 DMA Controller

- DMA stands for Direct Memory Access.
- It is designed by Intel to transfer data at the fastest rate.
- It allows the device to transfer the data directly to/from memory without any interference of the CPU.
- Using a DMA controller, the device requests the CPU to hold its data, address and control bus, so the device is free to transfer data directly to/from the memory.
- The DMA data transfer is initiated only after receiving HLDA signal from the CPU.



Direct Memory Access

- The transfer of data between the peripheral and memory without the interaction of CPU and letting the peripheral device manage the memory bus directly is termed as Direct Memory Access (DMA).
- The two control signals Bus Request and Bus Grant are used to fascinate the DMA transfer.
- The bus request input is used by the DMA controller to request the CPU for the control of the buses.
- When BR signal is high, the CPU terminates the execution of the current instructions and then places the address, data, read and write lines to the high impedance state and sends the bus grant signal.
- The DMA controller now takes the control of the buses and transfers the data directly between memory and I/O without processor interaction.
- When the transfer is completed, the bus request signal is made low by DMA. In response to which CPU disables the bus grant and again CPU takes the control of address, data, read and write lines.
- The transfer of data between the memory and I/O of course facilitates in two ways which are DMA Burst and Cycle Stealing.

Direct Memory Access

- 1. DMA Burst
- The block of data consisting a number of memory words is transferred at a time.
- 2. Cycle Stealing
- DMA transfers one data word at a time after which it must return control of the buses to the CPU.
- CPU is usually much faster than I/O (DMA), thus CPU uses the most of the memory cycles
- DMA Controller steals the memory cycles from CPU
- For those stolen cycles, CPU remains idle
- For those slow CPU, DMA Controller may steal most of the memory cycles which may cause CPU remain idle long time

How DMA Operations are Performed?

- Following is the sequence of operations performed by a DMA –

✓Initially, when any device has to send data between the device and the memory, the device has to send DMA request (DRQ) to DMA controller.

✓The DMA controller sends Hold request (HRQ) to the CPU and waits for the CPU to assert the HLDA.

✓Then the microprocessor tri-states all the data bus, address bus, and control bus. The CPU leaves the control over bus and acknowledges the HOLD request through HLDA signal.

✓Now the CPU is in HOLD state and the DMA controller has to manage the operations over buses between the CPU, memory, and I/O devices.

Features of 8257

- **Here is a list of some of the prominent features of 8257** –
  - ✓It has four channels which can be used over four I/O devices.
  - ✓Each channel has 16-bit address and 14-bit counter.
  - ✓Each channel can transfer data up to 64kb.
  - ✓Each channel can be programmed independently.
  - ✓Each channel can perform read transfer, write transfer and verify transfer operations.
  - ✓It generates MARK signal to the peripheral device that 128 bytes have been transferred.
  - ✓It requires a single phase clock.
  - ✓Its frequency ranges from 250Hz to 3MHz.
  - ✓It operates in 2 modes, i.e., Master mode and Slave mode.

8255A -Programmable Peripheral Interface

- The 8255A is a general purpose programmable I/O device designed to transfer the data from I/O to interrupt I/O under certain conditions as required.
- It can be used with almost any microprocessor.
- It consists of three 8-bit bidirectional I/O ports (24I/O lines) which can be configured as per the requirement.

Ports of 8255A

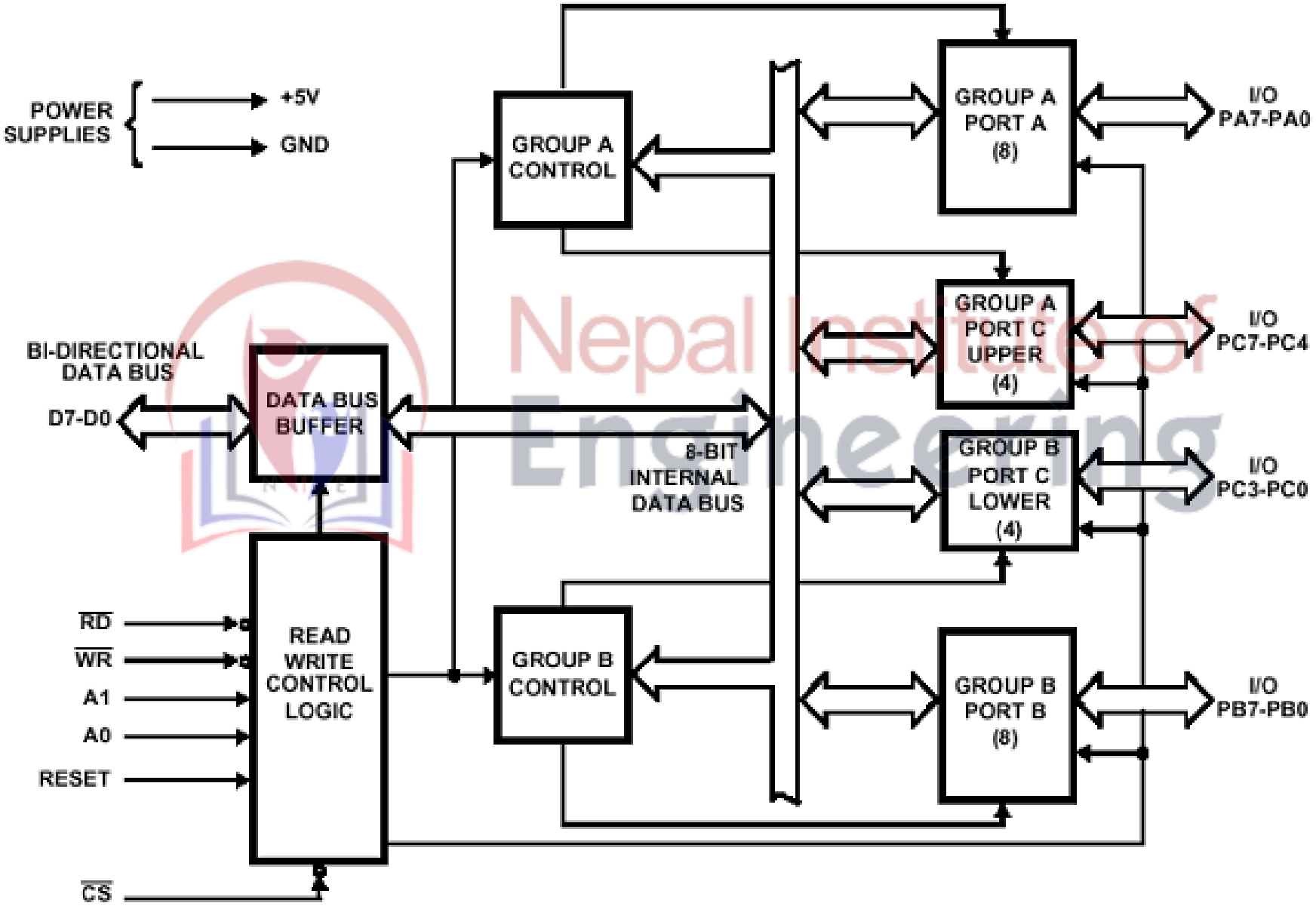
- 8255A has three ports, i.e., PORT A, PORT B, and PORT C.
- **Port A** contains one 8-bit output latch/buffer and one 8-bit input buffer.
- **Port B** is similar to PORT A.
- **Port C** can be split into two parts, i.e. PORT C lower (PC0-PC3) and PORT C upper (PC7-PC4) by the control word.
- These three ports are further divided into two groups, i.e. Group A includes PORT A and upper PORT C. Group B includes PORT B and lower PORT C.
- These two groups can be programmed in three different modes, i.e. the first mode is named as mode 0, the second mode is named as Mode 1 and the third mode is named as Mode 2.

Operating Modes

- 8255A has three different operating modes −
- **Mode 0 –**
  - In this mode, Port A and B is used as two 8-bit ports and Port C as two 4- bit ports. Each port can be programmed in either input mode or output mode where outputs are latched and inputs are not latched. Ports do not have interrupt capability.
- **Mode 1 –**
  - In this mode, Port A and B is used as 8-bit I/O ports. They can be configured as either input or output ports. Each port uses three lines from port C as handshake signals. Inputs and outputs are latched.
- **Mode 2 –**
  - In this mode, Port A can be configured as the bidirectional port and Port B either in Mode 0 or Mode 1. Port A uses five signals from Port C as handshake signals for data transfer. The remaining three signals from Port C can be used either as simple I/O or as handshake for port B.

The features of 8255A are as follows –

- ✓It consists of 3 8-bit IO ports i.e. PA, PB, and PC.
- ✓Address/data bus must be externally demux'd.
- ✓It is TTL compatible.
- ✓It has improved DC driving capability.


POWER SUPPLIES
+5V
GND
GROUP A CONTROL
GROUP A PORT A (8)
I/O PA7-PA0
GROUP A PORT C UPPER (4)
I/O PC7-PC4
BI-DIRECTIONAL DATA BUS
D7-D0
DATA BUS BUFFER
8-BIT INTERNAL DATA BUS
GROUP B PORT C LOWER (4)
I/O PC3-PC0
RD
WR
A1
A0
RESET
READ WRITE CONTROL LOGIC
GROUP B CONTROL
GROUP B PORT B (8)
I/O PB7-PB0
CS

8255A Architecture

- **Data Bus Buffer**
- It is a tri-state 8-bit buffer, which is used to interface the microprocessor to the system data bus. Data is transmitted or received by the buffer as per the instructions by the CPU. Control words and status information is also transferred using this bus.
- **Read/Write Control Logic**
- This block is responsible for controlling the internal/external transfer of data/control/status word. It accepts the input from the CPU address and control buses, and in turn issues command to both the control groups.

8255A Architecture

- CS
- It stands for Chip Select.
- A LOW on this input selects the chip and enables the communication between the 8255A and the CPU. It is connected to the decoded address, and A0 & A1 are connected to the microprocessor address lines. Their result depends on the following conditions −

CS	A_1	A0	Result
0	0	0	PORT A
0	0	1	PORT B
0	1	0	PORT C
0	1	1	Control Register
1	X	X	No Selection

8255A Architecture

- WR
  - ✓ It stands for write. This control signal enables the write operation. When this signal goes low, the microprocessor writes into a selected I/O port or control register.
- RESET
  - ✓ This is an active high signal. It clears the control register and sets all ports in the input mode.
- RD
  - ✓ RD It stands for Read. This control signal enables the Read operation. When the signal is low, the microprocessor reads the data from the selected I/O port of the 8255.
- A0 and A1
  - ✓ These input signals work with RD, WR, and one of the control signal. Following is the table showing their various signals with their result.

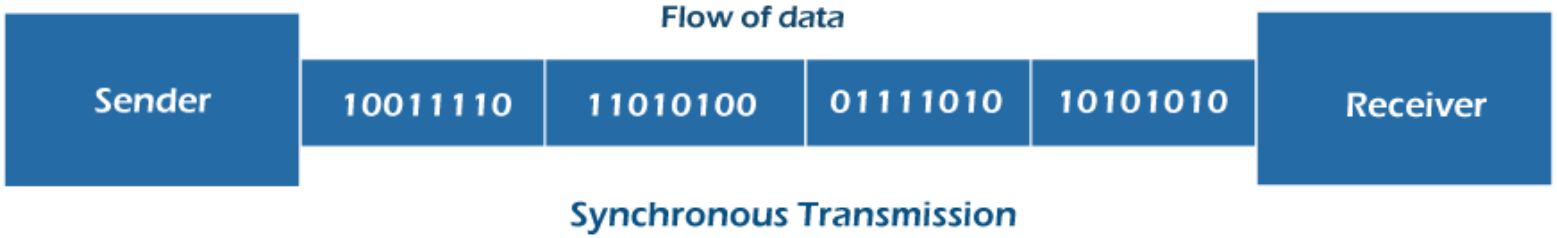
A_1	A_0	RD	WR	CS	Result
0	0	0	1	0	**Input Operation** PORT A → Data Bus
0	1	0	1	0	PORT B → Data Bus
1	0	0	1	0	PORT C → Data Bus
0	0	1	0	0	**Output Operation** Data Bus → PORT A
0	1	1	0	0	Data Bus → PORT A
1	0	1	0	0	Data Bus → PORT B

Data Transmission

- The action of transferring data or anything from one place to other is referred to as transmission.
- It is a method of sharing data between two devices linked by a network, also known as communication mode.
- Synchronous and asynchronous transmissions are the two main types of transmission used in computer networking.
- In both synchronous and asynchronous transmission, data is sent between the transmitter and the receiver based on a clock pulse utilized for synchronization.
- These serial data transmission techniques are both known as synchronous transmission.

Synchronous Transmission

- A data transfer method in which a continuous stream of data signals is accompanied by timing signals (generated by an electronic clock) to ensure that the transmitter and the receiver are in synchronized with one another.
- The data is sent in blocks spaced by fixed time intervals.
- Are used when large amounts of data must be transferred very quickly from one location to the other.
- It synchronizes transmission speeds at both the receiving and sending end of the transmission using clock signals built into each component. A continual stream of data is then sent between the two nodes.
- Most network protocols (such as Ethernet, SONET, Token Ring) use synchronous transmission.

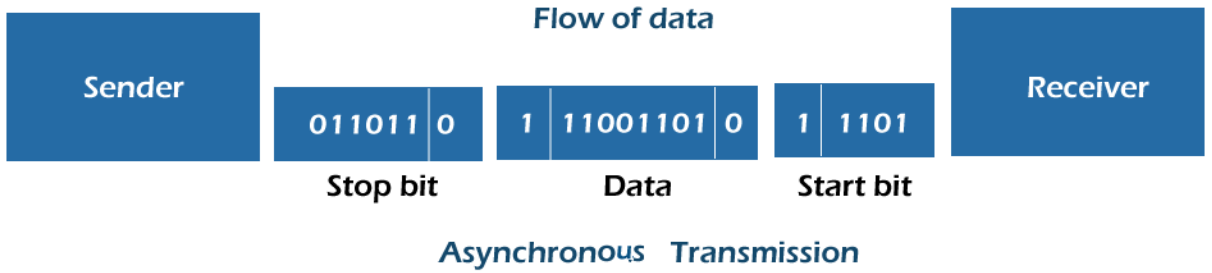


Characteristics of Synchronous Transmission

- The following is a list of characteristics specific to synchronous communication:
  - ✓There are no gaps between characters being transmitted.
  - ✓Timing is supplied by modems or other devices at each end of the connection.
  - ✓Special syn characters precede the data being transmitted.
  - ✓The syn characters are used between blocks of data for timing purposes.

Asynchronous Transmission

- asynchronous transmission works in spurts and must insert a start bit before each data character and a stop bit at its termination to inform the receiver where it begins and ends.
- The term *asynchronous* is used to describe the process where transmitted data is encoded with start and stop bits, specifying the beginning and end of each character.
- These additional bits provide the timing or synchronization for the connection by indicating when a complete character has been sent or received; thus, timing for each character begins with the start bit and ends with the stop bit.

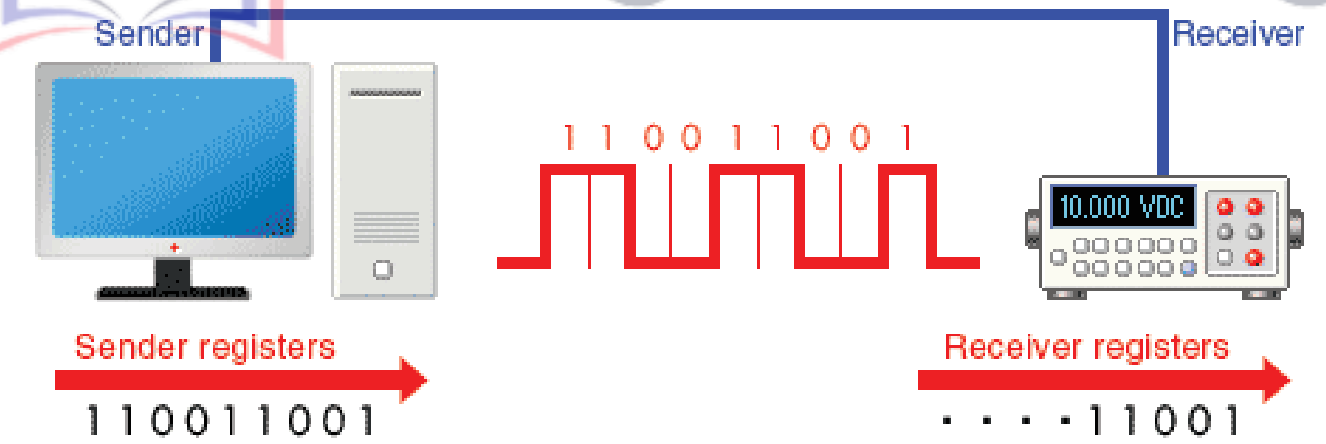


Characteristics of Asynchronous Transmission

- The following is a list of characteristics specific to Asynchronous communication:
  - Each character is preceded by a start bit and followed by one or more stop bits.
  - Gaps or spaces between characters may exist.

Serial Communication

- Serial communication is a communication method that uses one or two transmission lines to send and receive data, and that data is continuously sent and received one bit at a time.
- Since it allows for connections with few signal wires, one of its merits is its ability to hold down on wiring material and relaying equipment costs.t



Serial communication standards

- RS-232C/RS-422A/RS-485 are EIA (Electronic Industries Association) communication standards.
- Of these communication standards, RS-232C has been widely adopted in a variety of applications, and it is even standard equipment on computers and is often used to connect modems and mice.
- Sensors and actuators also contain these interfaces, many of which can be controlled via serial communication.
- RS-232C
  - ✓ This serial communication standard is widely used and is often equipped on computers as standard.
  - ✓ It is also called "EIA-232".The purpose and timing of the signal lines and the connectors have been defined (D-sub 25-pin or D-sub 9-pin).
  - ✓ Currently the standard has been revised with the addition of signal lines and is formally called "ANSI/EIA-232-E".However, even now it is generally referred to as "RS-232C".

Parallel communication

- Is a means of transmitting multiple binary digits (bits) simultaneously in data transmission. It differs from serial communication, which sends only one bit at a time; this distinction is one method to classify a communication channel.
- The data stream must be transmitted through "n" communication lines, which necessitates using many wires.
- This is an expensive mode of transportation; hence it is usually limited to shorter distances.

Parallel communication	Serial communication
More expensive	Less expensive
Needs “n” wires to transfer “n” bits of data	Needs a minimum of one wire to transfer all bits
Harder and more expensive to implement	Easier to implement
Mostly used for short-distance communication	Better for long-distance communication
Transfers more data at a time	Slower data transmission speed
Easier to analyze	Harder to analyze
Crosstalk more of an issue	Crosstalk significantly less of an issue
Clock skew can be an issue	No issues with clock skew
Example: Computer connected to a printer	Example: Computer connected to a smartphone

Interrupts

- The interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention.
- It alerts the processor to a high-priority process requiring interruption of the current working process. In I/O devices one of the bus control lines is dedicated for this purpose and is called the **Interrupt Service Routine (ISR***)*.
- While the processor is handling the interrupts, it must inform the device that its request has been recognized so that it stops sending the interrupt request signal.
- Also, saving the registers so that the interrupted process can be restored in the future, increases the delay between the time an interrupt is received and the start of the execution of the ISR. This is called **Interrupt Latency.**

Interrupts

- **Software Interrupts:**
  - A sort of interrupt called a software interrupt is one that is produced by software or a system as opposed to hardware.
  - Traps and exceptions are other names for software interruptions. They serve as a signal for the operating system or a system service to carry out a certain function or respond to an error condition.
  - A particular instruction known as a “interrupt instruction” is used to create software interrupts.
- **Hardware Interrupts:**
  - The interrupt signal generated from external devices and i/o devices are made interrupt to CPU when the instructions are ready. Hardware interrupts are classified into two types which are as follows −
- **Maskable Interrupt** − The hardware interrupts that can be delayed when a highest priority interrupt has occurred to the processor.
- **Non Maskable Interrupt** − The hardware that cannot be delayed and immediately be serviced by the processor.

Handling Multiple Devices:

- When more than one device raises an interrupt request signal, then additional information is needed to decide which device to be considered first. The following methods are used to decide which device to select:
- **Polling:**
  - ✓ In polling, the first device encountered with the IRQ bit set is the device that is to be serviced first. Appropriate ISR is called to service the same. It is easy to implement but a lot of time is wasted by interrogating the IRQ bit of all devices.
- **Vectored Interrupts:**
  - ✓ A device requesting an interrupt identifies itself directly by sending a special code to the processor over the bus. This enables the processor to identify the device that generated the interrupt. The special code can be the starting address of the ISR or where the ISR is located in memory and is called the interrupt vector.
- **Interrupt Nesting:**
  - ✓ The I/O device is organized in a priority structure. Therefore, an interrupt request from a higher priority device is recognized whereas a request from a lower priority device is not. The processor accepts interrupts only from devices/processes having priority.

Interrupt processing

- Step 1 – First device issues interrupt to CPU.
- Step 2 – Then, the CPU finishes execution of current instruction.
- Step 3 – CPU tests for pending interrupt request. If there is one, it sends an acknowledgment to the device which removes its interrupt signal.
- Step 4 – CPU saves program status word onto control stack.
- Step 5 – CPU loads the location of the interrupt handler into the PC register.
- Step 6 – Save the contents of all registers from the control stack into memory.
- Step 7 – Find out the cause of interrupt, or interrupt type or invokes appropriate routine.
- Step 8 – Restore saved registers from the stack.
- Step 9 – Restore PC to dispatch the original process.

ANY QUESTION ?