B.Sc (HONS.) IN ECE, PART-III, SIXTH SEMESTER EXAMINATION, 2019

COMPUTER PERIPHERALS AND INTERFACING

Subject Code: ECE-315

Time – 3 hours

Full Marks – 80

[N.B. The figures in the right margin indicate full marks. Answer any five questions.]

(a) Define the term peripheral. What are the roles of computer peripherals? (4 Marks)

Peripheral:

A peripheral is any hardware device that is external to the CPU and memory unit but is connected to the computer to perform input, output, storage, or communication functions.

Roles of peripherals:

1. **Input devices** – provide data to the computer (e.g., keyboard, mouse, scanner).
2. **Output devices** – display processed data (e.g., monitor, printer).
3. **Storage devices** – store large amounts of data (e.g., hard disk, SSD, optical disk).
4. **Communication devices** – enable data transfer between systems (e.g., modem, network card).

Q1(b) What does interfacing mean? Discuss the basic interfacing unit with proper diagram. (6 Marks)

Answer:

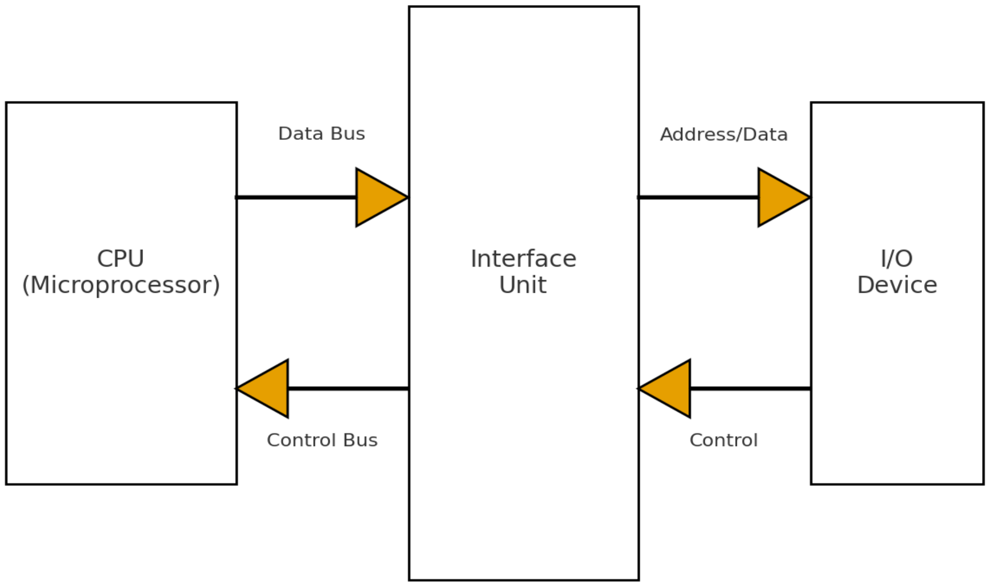
Interfacing:

Interfacing is the method of connecting the microprocessor with input/output (I/O) devices and memory so that they can exchange information correctly.

Because the CPU, memory, and peripherals often differ in speed, data format, and voltage levels, an **interface unit** is required to match them.

Functions of Interfacing Unit:

1. **Data transfer** between CPU and peripherals.
2. **Address decoding** to select the correct I/O device.
3. **Synchronization** between fast CPU and slow devices.
4. **Buffering** to protect CPU from device loading.
5. **Control signal generation** (e.g., Read/Write, Chip Select).



Q1(c) What are the basic requirements for proper interface between a microprocessor and an I/O device? Discuss with diagrams. (6 Marks)

Answer:

When a microprocessor communicates with an I/O device, certain requirements must be met so that data transfer is reliable and error-free.

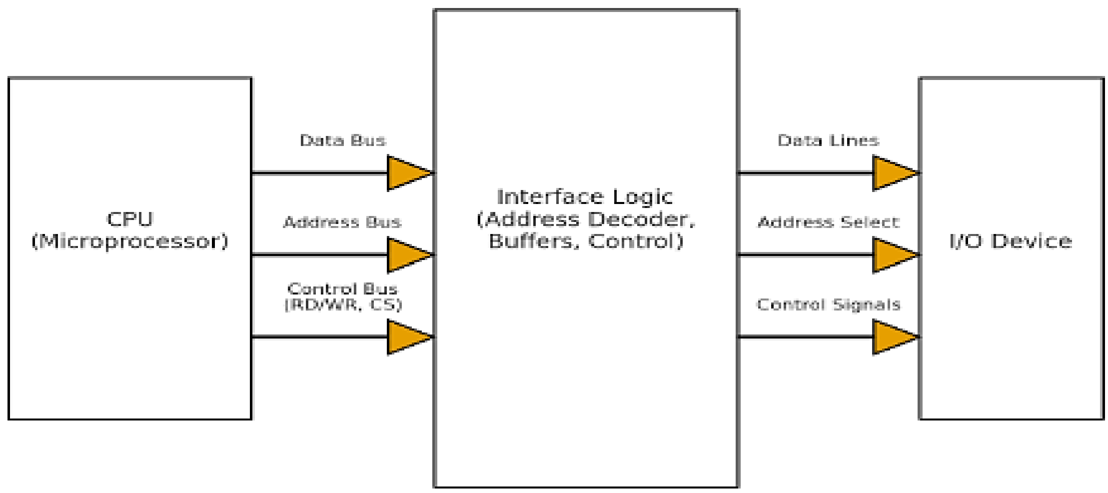
Basic Requirements:

1. **Data Compatibility:**
   - CPU may be 8-bit, 16-bit, or 32-bit.
   - The interface must ensure that the device matches the data size.
2. **Timing Compatibility:**
   - CPU works very fast, while peripherals are usually slow.
   - Interface must provide synchronization using control signals like **READY** or **WAIT**.
3. **Voltage/Logic Level Matching:**
   - Devices may use different voltage levels (e.g., TTL, CMOS).
   - Interface ensures proper logic level conversion.
4. **Address Decoding:**
   - Each device should have a unique I/O address.
   - Address decoder in the interface selects the correct device.
5. **Control Signals:**
   - Read (RD), Write (WR), and Chip Select (CS) signals are needed to activate the device.
6. **Buffering:**
   - Buffers protect the CPU from heavy current loads and isolate signals.

Diagram:

Now, let me draw the block diagram for **Microprocessor–I/O device interface.**

Matplotlib Chart



Here's the **diagram of microprocessor–I/O device interface** 👆

- CPU sends **address, data, and control signals.**
- The **interface logic** (decoder, buffers, control) ensures compatibility.
- Finally, the **I/O device** receives or sends data correctly.

Q2

(a) What is DMA? Why does the DMA generally have priority over the CPU in the usage of accessing memory? (6 Marks)

Direct Memory Access (DMA):

- DMA is a method where an external controller transfers data directly between I/O devices and memory, bypassing the CPU.
- It is used for **high-speed data transfer** (e.g., disk, graphics, audio).

Why DMA has priority over CPU:

1. Data transfer must be fast to avoid data loss (especially from high-speed devices like disk/communication lines).
2. If CPU always controlled transfer, it would waste cycles.
3. DMA controller temporarily takes control of the system bus (data, address, control) from CPU → known as **cycle stealing**.
4. CPU is slower compared to direct hardware transfers → so DMA is prioritized.

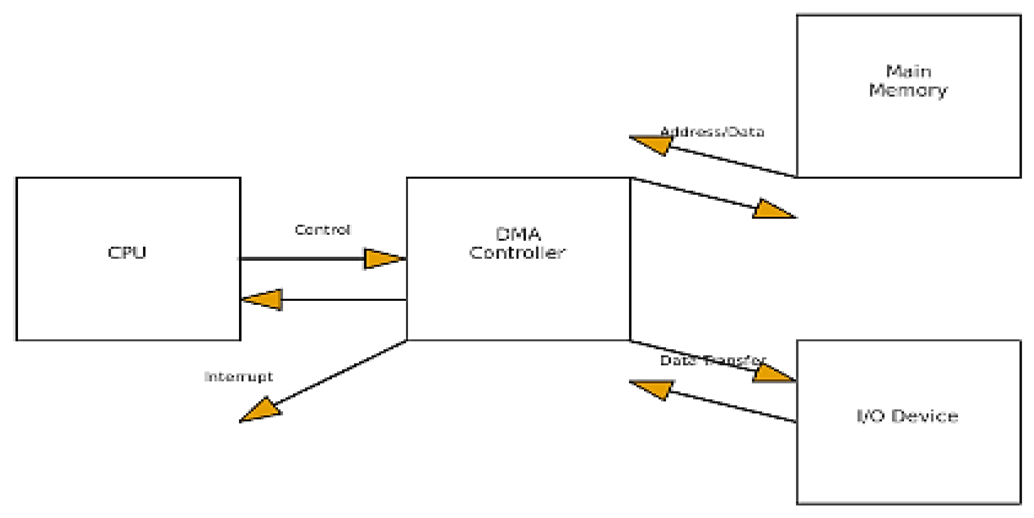
(b) Draw the block diagram showing how a DMA controller operates with memory support system. (6 Marks)

Working Principle:

- CPU gives command to DMA controller with source, destination, and count.
- DMA controller takes over the system bus.
- It transfers data directly between I/O device and memory.
- After completion, it sends an **interrupt** to CPU.

Block Diagram:

Matplotlib Chart



Here's the **DMA controller block diagram** 👆

- CPU gives commands to DMA.
- DMA directly transfers data between I/O and Memory.
- After completion, DMA interrupts CPU.

(c) Explain the input connection of a stepper motor. How is it interfaced to a microprocessor? (4 Marks)

Stepper Motor:

- A stepper motor moves in discrete angular steps (e.g., 1.8° per step).
- Controlled by energizing coils in a specific sequence.
- Input: digital pulses → each pulse = one step rotation.

Input Connection:

- Requires multiple windings (usually 4-phase or 2-phase).
- Inputs are driven by a **driver circuit (transistors/ULN2003)** because microprocessor signals are weak.

Interfacing with Microprocessor:

1. Microprocessor sends pulse sequence via output port (e.g., 8255 PPI).
2. Driver circuit amplifies current.
3. Stepper motor rotates in desired direction and speed (depends on frequency of pulses).

Diagram (simplified):

- Microprocessor → 8255 PPI → Driver Circuit → Stepper Motor windings.

Q4

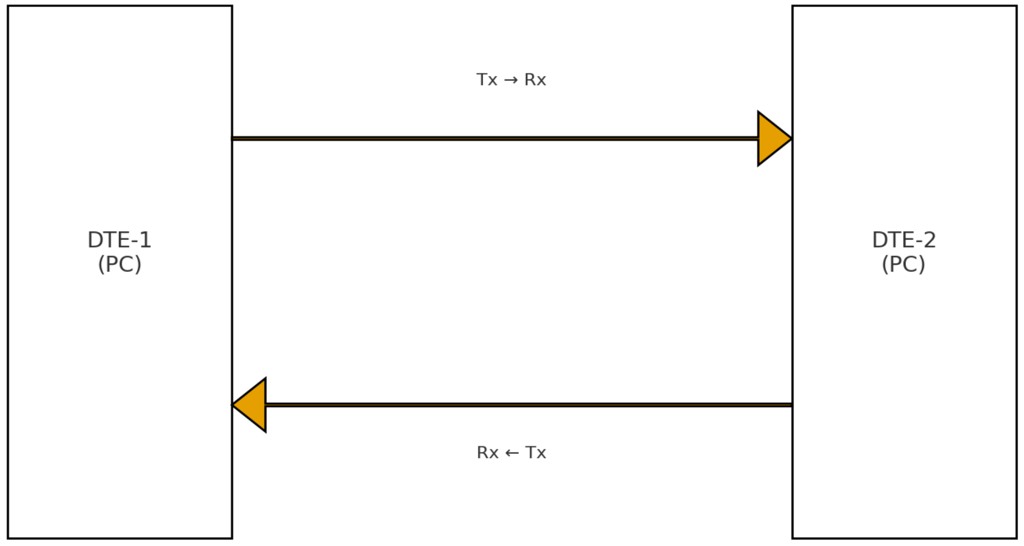
(a) What is a null modem? Why is null modem used in RS-232-C/V.24 serial data interface? (6 Marks)

Null Modem:

- A *null modem* is a communication method that connects two DTE (Data Terminal Equipment, e.g., two PCs) directly using an RS-232 serial cable **without a modem.**
- It works by **crossing the transmit (Tx) and receive (Rx) lines.**

Why Used in RS-232-C/V.24:

- In RS-232, normally DTE ↔ DCE (PC ↔ modem).
- If two DTEs must communicate directly, then Tx of one must go to Rx of the other → hence a null modem cable is used.
- Control signals (RTS/CTS, DTR/DSR) may also be cross-connected.



(b) Describe the IEEE 488 bus structure briefly. (6 Marks)

IEEE 488 (also called GPIB – General Purpose Interface Bus):

- A standard digital interface for connecting and controlling instruments (oscilloscopes, signal generators, etc.) with computers.
- Developed for laboratory instrumentation.

Structure:

- It is a **parallel bus** (8-bit data lines).
- Supports **up to 15 devices** (1 controller + multiple instruments).
- Maximum cable length: ~20 meters.

Bus Lines (24 lines total):

1. **8 Data lines (DIO1–DIO8)** – bidirectional data.
2. **8 Control lines** – ATN, DAV, NRFD, NDAC, IFC, SRQ, REN, EOI.
3. **8 Ground lines** – signal grounds.

Features:

- Handshaking for reliable data transfer.
- Multidevice communication (one device acts as *talker*, others as *listeners*).
- Standardized for scientific and industrial instruments.

(c) In interfacing of DCE, how will you use IC-8251 USART with a modem for transmitting data over a long-range? (4 Marks)

8251 USART (Universal Synchronous/Asynchronous Receiver/Transmitter):

- It is a programmable device that handles serial communication.
- Converts parallel data from CPU → serial form (Tx), and serial data → parallel (Rx).

Interfacing with Modem (DCE):

1. **Microprocessor** sends data to **8251 USART** in parallel form.
2. **8251 USART** converts it to serial data (using baud rate generator).
3. Serial data is sent to **modem (DCE)** via RS-232 standard.
4. Modem transmits over long-distance communication line.

Signals used:

- TxD, RxD for data.
- RTS/CTS and DTR/DSR for control.

Diagram (conceptual):

- Microprocessor → 8251 USART → RS-232 driver (MAX232) → Modem (DCE).

Q5

(a) Describe interfacing of 7-segment LED with 8255A. (6 marks)

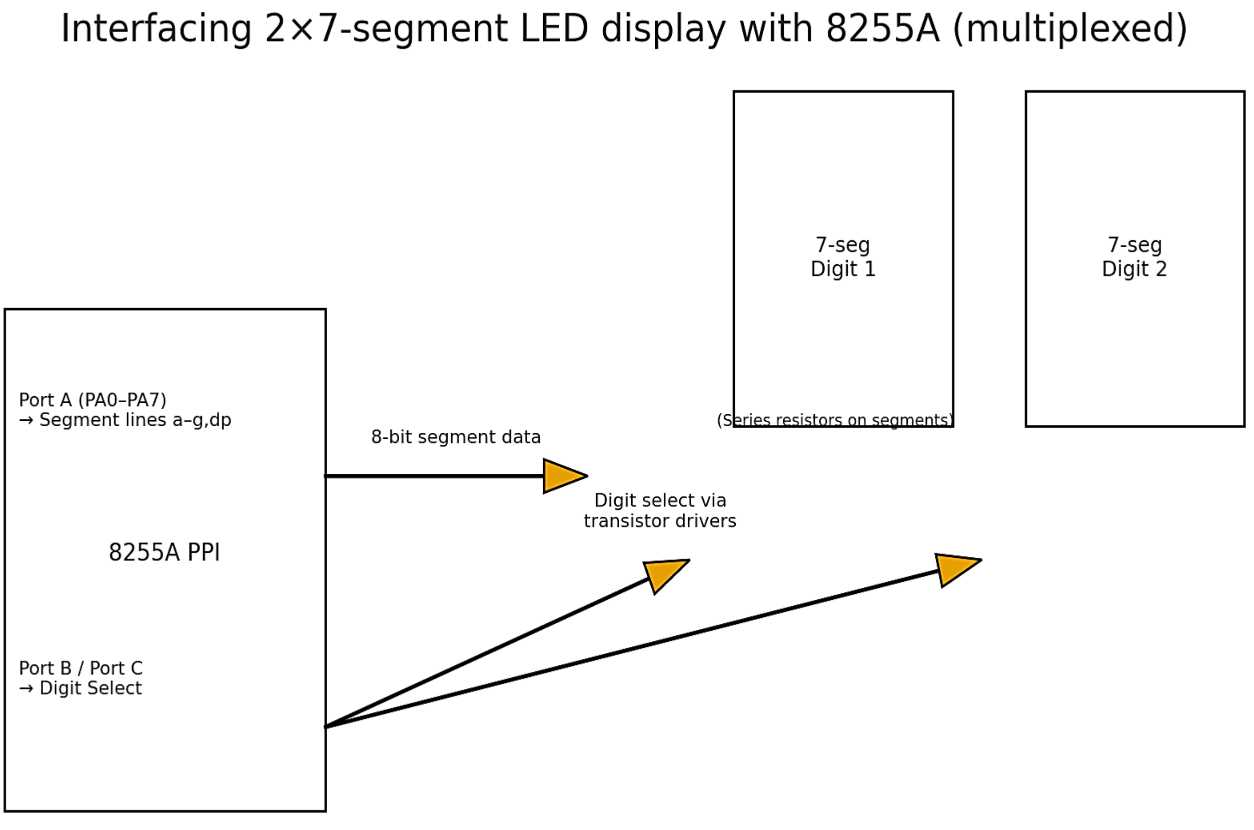
Idea: Use 8255A as the parallel port interface. Send the 7-segment code (a–g, dp) through one 8-bit port; use another port to select which digit turns ON (for multiplexing multiple digits). Drivers (transistor/ULN2003) limit current.

Steps (for 2 digits, multiplexed):

1. **Port A (PA0–PA7)** → connect to segments **a–g, dp** through series resistors.
2. **Port B or Port C bits** → drive **digit-enable** lines via NPN transistors (for common-anode use PNP/high-side; for common-cathode use NPN/low-side).
3. Program 8255 in **Mode-0 output** for Port-A and the digit-select bits.
4. **Refresh loop** in code:
   - Output 7-segment pattern of Digit-1 on PA → enable Digit-1 line → small delay.
   - Output pattern of Digit-2 → enable Digit-2 → delay.
   - Repeat fast (≈1–2 ms per digit) so both appear continuously ON.

Diagram (conceptual, multiplexed 2-digit):

(Rendered for you — use in your answer.)



(b) What do PPI and PIC stand for? Draw the internal architecture of 8255 PPI and discuss its ports. (6 marks)

Expansions:

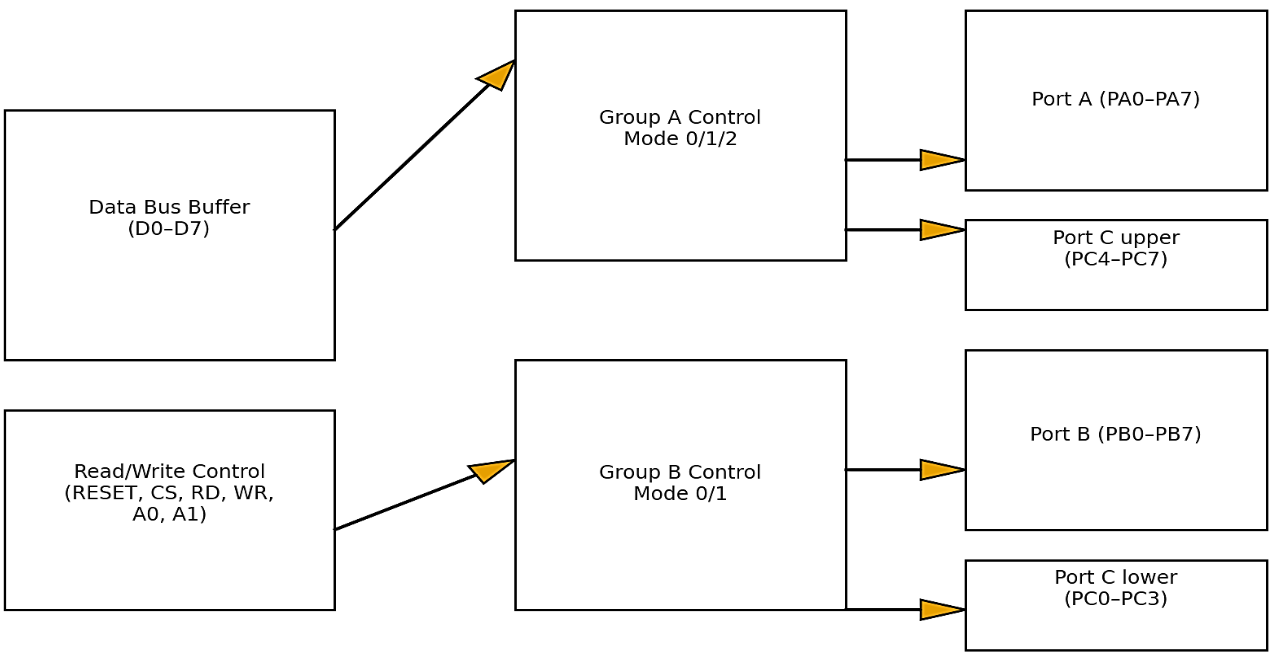
- PPI → *Programmable Peripheral Interface*
- PIC → *Programmable Interrupt Controller* (e.g., 8259)

8255 PPI – Internal architecture & ports:

- **Data Bus Buffer (D0–D7):** Tri-state buffer to the system data bus.
- **Read/Write Control block:** Handles `RESET, CS, RD, WR, A0, A1`; decodes addresses; latches **Control Word**.
- **Group A Control:** Manages **Port-A** and **Port-C upper** (PC4–PC7); supports **Mode 0, 1, 2.**
- **Group B Control:** Manages **Port-B** and **Port-C lower** (PC0–PC3); supports **Mode 0, 1.**
- **Ports:**
  - **Port-A (8-bit):** Data I/O; in Mode-2 can act as bidirectional bus with handshake.
  - **Port-B (8-bit):** Data I/O; simple (Mode-0) or handshake (Mode-1).
  - **Port-C (8-bit split):** Upper PC4–PC7 & Lower PC0–PC3; used either as general I/O or **handshake/control** lines for Modes 1/2; individual bit set/reset available.

Diagram:

8255 PPI – Internal Architecture



A0,A1 select ports/control; Control word programs modes; Port C bits used for handshaking.

(e) Plotters

- Output devices for producing high-precision drawings, graphs, and engineering diagrams.
- Unlike printers, plotters **draw with pens** on paper.
- Types:
  1. Drum plotter (paper moves on drum).
  2. Flatbed plotter (pen moves across stationary paper).
- Features: High accuracy, vector graphics, used in CAD/CAM.

7e

(c) Explain the operation of A/D converters. (4 marks)

Purpose: Convert an analog input V_{in} into an **N-bit digital code** representing one of 2^N quantization levels between 0 and V_{ref}.

Common steps (generic A/D):

1. **Sample & Hold:** capture the analog value at an instant.
2. **Quantization & Encoding:** choose the nearest discrete level; output the corresponding binary code.
3. **Conversion complete:** raise **EOC** (End-Of-Conversion) for processor read.

Popular ADC types (mention 2–3 with one-line working):

- **Successive Approximation (SAR):** Binary search using internal DAC + comparator; good speed, low power; most common in µC.
- **Flash (Parallel):** $2^N - 1$ comparators compare simultaneously; extremely fast; hardware-heavy.
- **Dual-Slope/Integrating:** Integrate V_{in} for fixed time, then de-integrate with V_{ref}; very accurate, excellent noise rejection; slower (used in DMMs).
- *(Also: Pipeline, Sigma-Delta for high-resolution audio, etc. — optional mention.)*

Key specifications to state: Resolution $= \frac{V_{ref}}{2^N}$, conversion time, accuracy, INL/DNL, S/H requirement, reference stability.

(a) What are the differences between active and passive transducers? (4 Marks)

Transducer: A device that converts one form of energy into another (typically physical quantity → electrical signal).

Differences:

Feature	Active Transducer	Passive Transducer
Power source	Does not require external power (self-generating).	Requires external excitation.
Output signal	Produces voltage/current directly.	Produces variation in resistance, inductance, or capacitance.
Examples	Thermocouple, Piezoelectric crystal, Photovoltaic cell.	LVDT, Strain gauge, Thermistor, Capacitive sensor.
Working principle	Converts non-electrical energy directly into electrical.	Changes electrical parameters which need external circuitry.

(d) R-2R Ladder type DAC

- A simple and popular **Digital-to-Analog Converter** using only two resistor values: R and 2R.
- Each bit of digital input controls a switch (connects to reference voltage or ground).
- The ladder network sums weighted currents, generating proportional analog voltage.
- Advantages: Easy to design, requires only two resistor values, good accuracy.
- Limitation: Speed limited by resistor-capacitor settling.

7 d

(b) Explain the construction and working principle of a Linear Voltage Differential Transformer (LVDT). (6 Marks)

Construction:

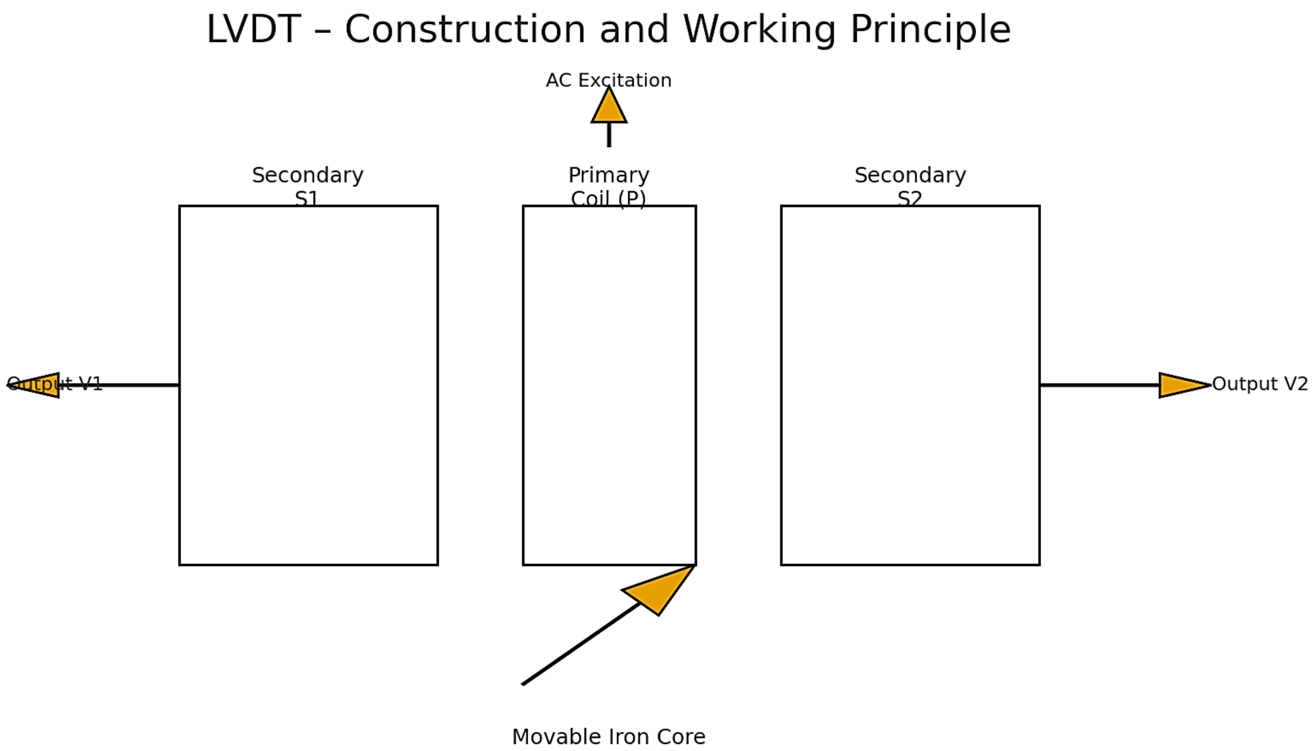
- A cylindrical transformer with a **primary coil** in the center and two **secondary coils** (S1 and S2) symmetrically placed on either side.
- A **soft iron movable core** slides inside the coil assembly.
- Secondary coils are connected in series opposition.

Working Principle:

1. AC excitation applied to primary coil.
2. Core position determines mutual inductance between primary and secondaries.
3. At **null position (core centered):** voltages induced in S1 and S2 are equal → differential output = 0.
4. If core moves towards S1 → output voltage increases in one phase.
5. If core moves towards S2 → output voltage increases in opposite phase.
6. Output magnitude ∝ displacement, output polarity ∝ direction.

Applications: Displacement measurement, position sensors in control systems.

Diagram:



(c) CRT Monitor

- **Cathode Ray Tube (CRT):** a display device that uses an electron beam to illuminate a phosphor-coated screen.
- **Working:**
  - Electron gun produces beam.
  - Deflection coils (electromagnetic) steer the beam horizontally and vertically.
  - Phosphor screen glows where beam strikes.
- **Modes:** Raster scan (used in TV/monitors), Vector scan (oscilloscope).
- Disadvantages: bulky, high power. Replaced by LCD/LED.

7c

(c) What is a sensor? Describe the different types of sensors. (6 Marks)

Sensor:
A sensor is a device that detects or measures a physical parameter (temperature, pressure, displacement, light, etc.) and converts it into an electrical signal suitable for processing.

Types of Sensors (by measurand):

1. **Temperature Sensors:**
   - Thermocouples, RTD, Thermistors, IC temperature sensors.
2. **Displacement/Position Sensors:**
   - Potentiometer, LVDT, Optical encoders.
3. **Pressure Sensors:**
   - Strain-gauge pressure transducer, Piezoelectric sensor.
4. **Proximity Sensors:**
   - Inductive, Capacitive, Ultrasonic, Infrared sensors.
5. **Light Sensors:**
   - Photodiode, LDR, Photovoltaic cell.
6. **Speed/Flow Sensors:**
   - Tachometer, Hall-effect sensor, Anemometer.

Classification (other view):

- **Active vs Passive** (self-generating vs needs excitation).
- **Analog vs Digital** (continuous vs discrete output).
- **Contact vs Non-contact** sensors.

(a) 8259 Interrupt Controller

- The **8259 Programmable Interrupt Controller (PIC)** manages hardware interrupts for a microprocessor.
- Can handle **8 interrupt inputs (IR0–IR7)**, expandable to 64 by cascading multiple 8259s.
- Provides **priority resolution**: either fixed or rotating priority.
- Sends an **interrupt request (INT)** to CPU; waits for **INTA (interrupt acknowledge).**
- On INTA, it provides the **vector number** to CPU, pointing to the interrupt service routine.
- Widely used with Intel 8085/8086 systems.

(b) Capacitive Transducer

- A capacitive transducer works on the principle:

$$C = \frac{\varepsilon A}{d}$$

  where C = capacitance, A = plate area, d = distance, ε = permittivity.
- Any change in displacement, pressure, or thickness alters capacitance.
- Types:
  1. **Variable distance (d)** – e.g., diaphragm pressure sensor.
  2. **Variable area (A)** – overlapping plates change.
  3. **Variable dielectric (ε)** – dielectric material moves between plates.
- Applications: pressure sensors, displacement measurement, liquid level detection.