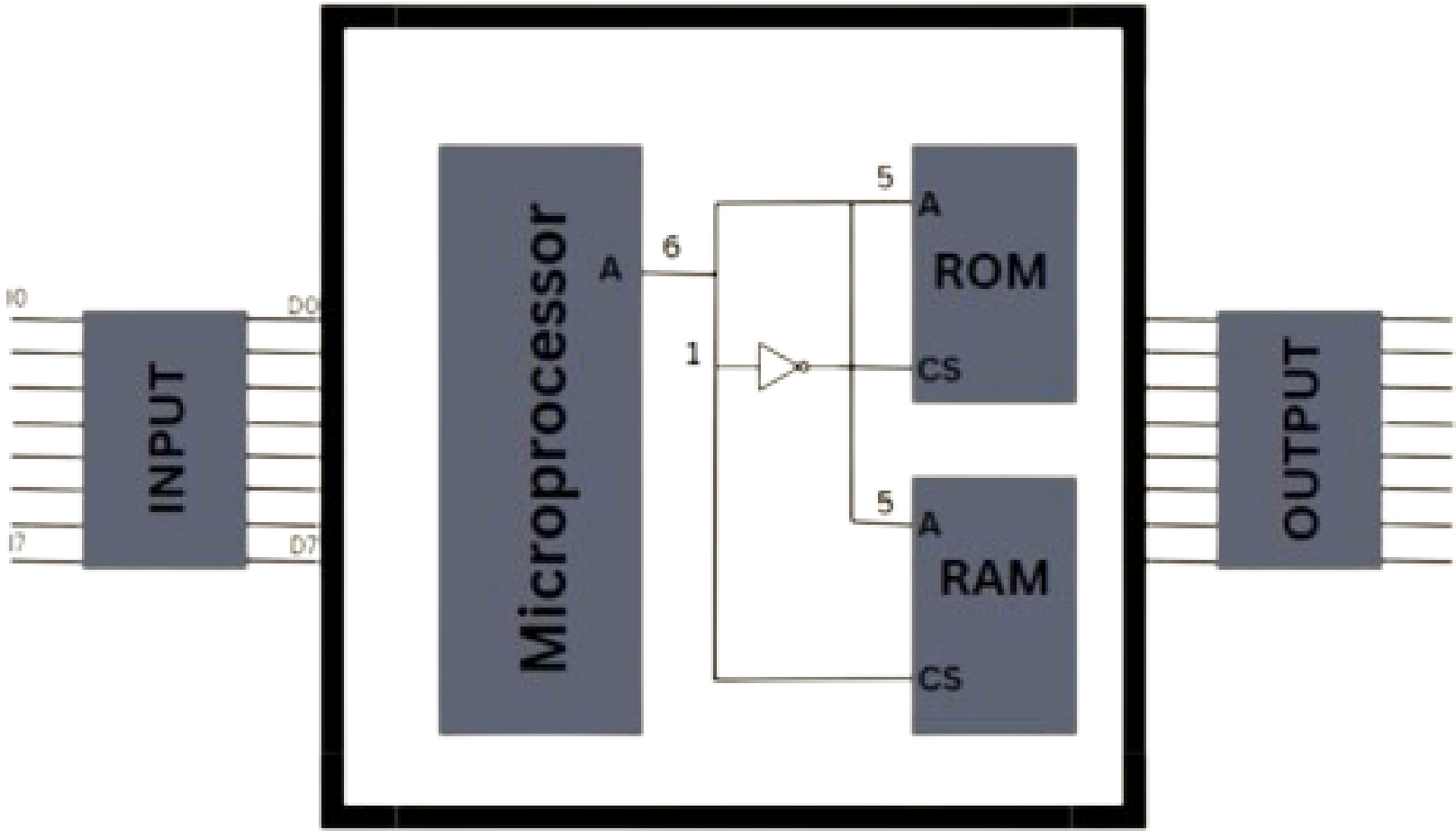
@Sree Vishnu Varthini

Day - 18

Embedded Systems Programming

MICROPROCESSOR COMPONENTS



Pic Credits: Manikanta Sai Vallamkonda

MICROPROCESSOR COMPONENTS

1. ROM (READ-ONLY MEMORY)

- **Purpose:** ROM stores permanent data and instructions essential for the system's basic functions.

- **Non-Volatile Memory:** ROM keeps its data even when power is off, making it ideal for firmware, boot instructions, and critical system code.

- **How it Works:** The microprocessor reads data from ROM but usually cannot write to it. ROM typically holds the bootloader and startup instructions needed to initialize the system.

MICROPROCESSOR COMPONENTS

2. RAM (RANDOM ACCESS MEMORY)

- **Purpose:** RAM provides temporary storage for data and instructions that the microprocessor is currently working on.

- **Volatile Memory:** RAM loses all data when the power is turned off, making it suitable for quick, temporary data access.

- **How it Works:** The microprocessor reads from and writes to RAM during operation. It can directly access any data location by specifying the memory address.

MICROPROCESSOR COMPONENTS

3. INPUT DEVICES

- **Purpose:** Allow data and commands to enter the system from external sources.

- **Examples:** Keyboards, mouse, sensors, touchscreens, or any device that provides input to the processor.

- **How it Works:** Input devices send data to the microprocessor through input ports. The microprocessor then processes this data according to the instructions it has been programmed to execute.

MICROPROCESSOR COMPONENTS

4. OUTPUT DEVICES

- **Purpose:** Display or convey the results of the microprocessor's operations to the user.

- **Examples:** Monitors, LEDs, printers, speakers, or any component that outputs information to the user.

- **How it Works:** The microprocessor sends data to the output device through output ports, allowing users to view or receive the processed data.

MICROPROCESSOR COMPONENTS

INTEGRATION IN A MICROPROCESSOR SYSTEM

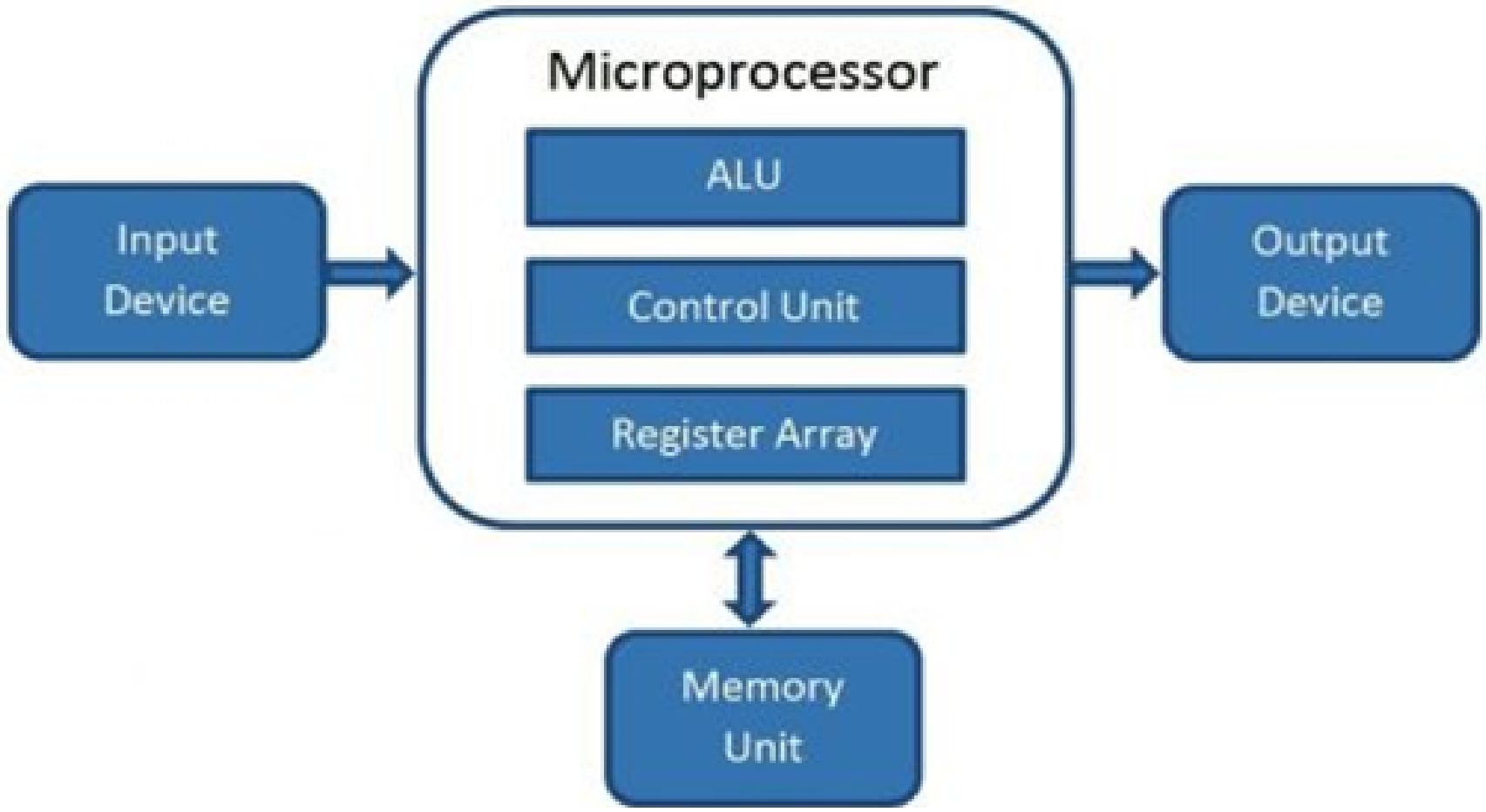
ROM: Holds primary programs like start-up and operational code.

RAM: Temporarily stores data, variables, and instructions during execution.

Input Devices: Send signals or data for processing.

Output Devices: Display or act on processed data, allowing user or environmental interaction

BLOCK DIAGRAM



@Sree Vishnu Varthini

Did you like the post?

follow for more!

P.S. You can access the **Embedded Systems Programming** course by contacting **Balajee Seshadri** Sir through the WhatsApp number provided on his linkedin profile.