SNS COLLEGE OF TECHNOLOGY

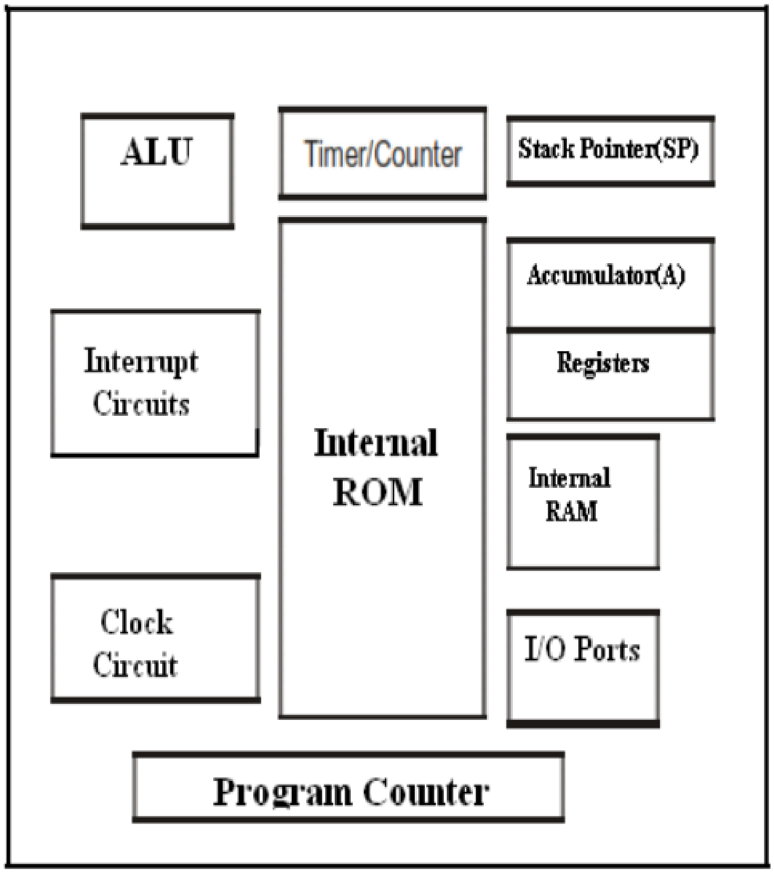
(An Autonomous Institution)

COIMBATORE – 641035

19MCE304- DESIGN OF EMBEDDED SYSTEMS

Functional building block of embedded system

The functional building blocks of an embedded system can be broadly categorized into hardware and software components. Here is a detailed breakdown of these building blocks:



Hardware Components

1. **Microcontroller/Microprocessor**:
   - **Microcontroller (MCU)**: An integrated circuit designed to perform specific control functions. It typically includes a CPU, memory, and I/O peripherals.
   - **Microprocessor (MPU)**: A central processing unit (CPU) used for general-purpose computation, requiring external components like memory and I/O interfaces.
2. **Memory**:
   - **Read-Only Memory (ROM)**: Stores firmware and application code. Types include EEPROM, Flash, and Mask ROM.

   - **Random Access Memory (RAM)**: Provides temporary storage for data and variables during operation.
3. **Input/Output (I/O) Interfaces**:
   - **Digital I/O**: Interfaces for digital signals, such as switches and LEDs.
   - **Analog I/O**: Interfaces for analog signals, such as sensors and actuators.
   - **Communication Interfaces**: Serial (UART, SPI, I2C), USB, Ethernet, CAN, etc.
4. **Timers and Counters**:
   - Used for timing operations, generating delays, and event counting.
5. **Power Supply**:
   - Converts and regulates power from external sources to the appropriate voltage levels required by the system.
6. **Sensors and Actuators**:
   - **Sensors**: Measure physical quantities like temperature, pressure, light, and motion.
   - **Actuators**: Perform actions based on control signals, such as motors, relays, and displays.
7. **System Clock**:
   - Provides the timing signals that synchronize operations within the system.

Software Components

1. **Firmware**:
   - Low-level software permanently stored in ROM, responsible for booting the system and basic operations.
2. **Real-Time Operating System (RTOS)**:
   - Manages hardware resources and provides an environment for executing tasks with real-time constraints.
3. **Device Drivers**:
   - Software modules that control and interact with hardware peripherals.
4. **Application Software**:
   - High-level code that performs the specific functions for which the embedded system is designed.
5. **Middleware**:
   - Provides common services and capabilities to applications beyond those offered by the operating system, such as communication protocols, data management, and graphics.

Functional Blocks in an Example Embedded System

1. **Input Processing Block**:
   - Interfaces with sensors to acquire data.
   - Converts and processes raw sensor data into usable information.
2. **Control Processing Block**:
   - Executes algorithms and control logic based on the processed input data.
   - Makes decisions and generates control signals.

3. **Output Processing Block**:
   - Interfaces with actuators to perform actions based on control signals.
   - Provides feedback and status information.
4. **Communication Block**:
   - Handles data exchange with other systems or devices.
   - Manages protocols and interfaces for wired or wireless communication.
5. **Power Management Block**:
   - Manages power distribution and consumption.
   - Implements power-saving modes and battery management.
6. **User Interface Block**:
   - Provides interfaces for user interaction, such as displays, buttons, and touchscreens.
   - Manages user inputs and presents system status and outputs.

These functional building blocks work together to perform the specific tasks and operations required by the embedded system. The design and integration of these blocks depend on the application's requirements and constraints.