مرکز آموزش نیـــرا سیستم
nirasystem.com

Embedded Systems Introduction

Ali Mirghasemi

Sections

- Embedded Systems
- Microcontrollers
- Microprocessors
- CPU
- System-on-Chip (SoC)
- Microcontroller Structure

Embedded Systems - Introduction

- Embedded Systems are specialized computing systems designed to perform specific tasks or functions within larger systems or devices.
- These systems are tightly integrated into hardware, making them dedicated and efficient for their intended applications.
- Embedded systems can be found in various domains, including automotive, consumer electronics, industrial automation, medical devices, and more.

Embedded Systems - Applications

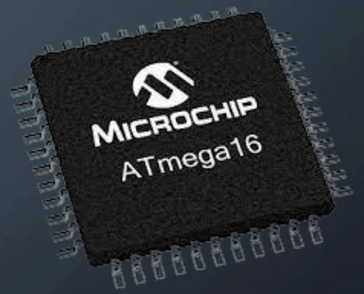
- Automotive:
  - Engine control units (ECUs)
  - Infotainment systems
- Consumer Electronics:
  - Smart TVs and set-top boxes
  - Home automation controllers
- Industrial Automation:
  - Programmable logic controllers (PLCs)
  - SCADA systems for monitoring and control
- Medical Devices:
  - Patient monitoring systems
  - Implantable medical devices

Embedded Systems - Board Examples

- Arduino Boards:
  - Arduino Uno for DIY electronics projects
  - Arduino Nano for compact designs
- Raspberry Pi:
  - Raspberry Pi Zero for low-cost embedded solutions
  - Raspberry Pi Compute Module for industrial applications
- Texas Instruments LaunchPad:
  - MSP430 LaunchPad for ultra-low-power applications
  - Tiva C Series LaunchPad for real-time control

Microcontroller - Introduction

- Microcontrollers are small, self-contained computing devices with integrated memory, I/O ports, and peripherals, commonly used in embedded systems.
- Advantages:
  - Cost-effective solution for specific tasks
  - Low power consumption for extended battery life
  - Compact size and easy integration
- Cons:
  - Limited processing power for complex applications
  - Smaller memory capacity compared to microprocessors
  - Fewer built-in peripherals than some SoCs



Microcontroller - Architecture

- Microcontrollers utilize the Harvard architecture with CPU, RAM, ROM/Flash, and built-in peripherals, allowing efficient control of devices.
- Peripherals:
  - General Purpose Input/Output (GPIO) pins
  - Timers and counters for precise timekeeping
  - Analog-to-Digital Converters (ADC) for sensor interfacing
  - Universal Asynchronous Receiver/Transmitter (UART) for serial communication

Microcontroller - Applications

- Home Automation:
  - Smart thermostats
  - Lighting control systems
- Automotive Systems:
  - Engine control units
  - Airbag systems
- Consumer Electronics:
  - Remote controls
  - Fitness trackers
- Industrial Automation:
  - Programmable logic controllers (PLCs)
  - Process control

- Arduino boards for DIY electronics projects
- Raspberry Pi Pico for low-cost microcontroller applications

Microprocessor - Introduction

- Microprocessors are CPUs designed for high processing power and general-purpose computing, found in devices like computers and laptops.
- Advantages:
  - High computing performance for complex tasks
  - Ability to handle multitasking and multiple applications
  - Support for various operating systems and software
- Cons:
  - Higher power consumption compared to microcontrollers
  - Larger physical size and higher cost
  - Limited on-chip peripherals, requiring external components

Microprocessor - Architecture

- Microprocessors follow the Von Neumann architecture, featuring CPU, ALU, Control Unit, and Registers. External memory chips are required for data storage.
- Peripherals:
  - Microprocessors usually require external components for peripherals such as keyboard, mouse, and display interface.

Microprocessor - Applications

- Personal Computers:
  - Desktops
  - Laptops
  - 2-in-1 devices
- Servers:
  - Data centers
  - Cloud computing
- Gaming Consoles:
  - PlayStation
  - Xbox
  - Nintendo Switch
- CISC
  - Intel 386
  - Intel 486
  - Pentium
  - Pentium Pro
  - Pentium II
- RISC
  - IBM RS6000
  - DEC Alpha 21064
  - DEC Alpha 21164

CPU - Introduction

- CPUs (Central Processing Units) are the heart of any computing device, responsible for executing instructions and performing calculations.
- Advantages:
  - High processing power for complex computations
  - Versatility to handle various tasks and workloads
  - Critical component in modern technology
- Cons:
  - Higher power consumption compared to microcontrollers
  - Requires additional components for a complete system
  - Complexity can lead to challenging debugging and design.

CPU - Architecture

- CPUs follow different architectures like x86, ARM, and RISC-V, with various components such as ALU, Control Unit, and Registers.
- Peripherals:
  - CPUs require external components like memory, I/O devices, and graphics cards for complete system functionality.

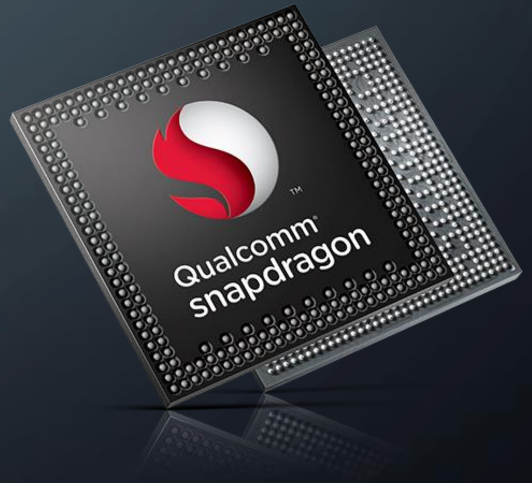
CPU - Applications

- Personal Computers:
  - Desktops
  - Laptops
  - Workstations
- Servers:
  - Data centers
  - Cloud computing
- Supercomputers:
  - Scientific research
  - Simulation and modeling

- Intel Core i9 for high-end desktop computing
- AMD EPYC processors for server applications

System-on-Chip (SoC) - Introduction

- System-on-Chip (SoC) integrates CPU (microcontroller/microprocessor), memory, peripherals, and other components on a single chip, enhancing device functionality.
- Advantages:
  - Compact size and reduced power consumption
  - Simplified design and reduced PCB space
  - Improved performance with integrated components
- Cons:
  - Higher complexity may lead to more challenging debugging
  - Customization options limited by integrated components



SoC - Architecture

- SoCs combine CPU, memory, graphics, and audio controllers on a single chip, commonly used in smartphones, IoT devices, and wearables.
- Peripherals:
  - Graphics Processing Unit (GPU) for enhanced graphics performance
  - Audio Codec for audio processing and playback
  - Connectivity options such as Wi-Fi, Bluetooth, and NFC

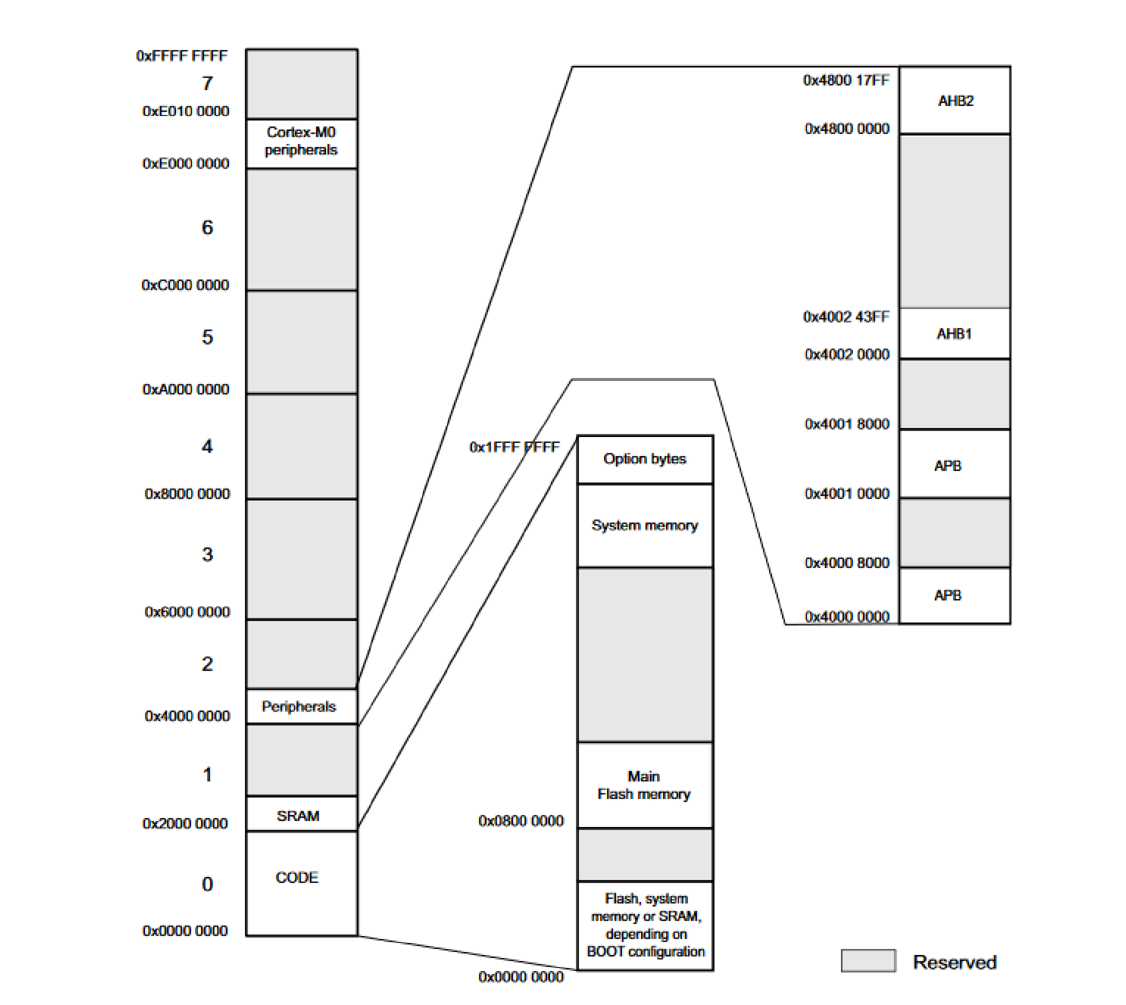
SoC - Applications

- Smartphones and Tablets:
  - Apple iPhone
  - Samsung Galaxy
- Internet of Things (IoT) Devices:
  - Smart home devices
  - Wearables
- Wearable Technology:
  - Smartwatches
  - Fitness trackers

- Qualcomm Snapdragon SoCs for smartphones and IoT devices
- Apple A-series SoCs for iPhones and iPads

Memory Map

- Memory Map organizes a microcontroller's or SoC's memory, including RAM, ROM/Flash, I/O ports, and special function registers, essential for efficient programming.



Flash Memory

- Flash memory is used in microcontrollers for non-volatile data storage.
- Types of Flash Memory:
  - NOR Flash: Commonly used for firmware storage and can execute code directly from memory.
  - NAND Flash: Typically used for data storage and requires a separate RAM buffer for code execution.

Register Map

- Special Function Registers (SFRs) control various hardware functionalities in microcontrollers.
- Examples:
  - GPIO Configuration Registers: Set pins as input or output.
  - Timer Control Registers: Configure timer settings for precise timekeeping.

RAM

- Microcontrollers have various areas in RAM to store data during program execution.
- Global Area:
  - Hold Static and global variables that must remain in entire of process
- Heap Area:
  - Dynamically allocated memory for variables during runtime.
- Stack Area:
  - Holds return addresses and local variables during function calls.