Subject: Embedded Computing Systems | **Theoretical: 2 hrs/wk**
Code: CS310 | **Practical: ---**
Class: 3rd Year | **Tutorial: 1 hrs/wk**
Pre-requisite: EE201 | **Units: 2**

==

1- **Embedded ARM microcontrollers:** ARM processor architecture, Software model, Addressing modes, programming instructions, Fundamental concepts of assembly language and linking: labels, address management.
2- **Microcontroller Hardware:** Microcontroller I/O pins, I/O programming and the direction register, Phased-lock loop, SysTick timer, Measurement of dynamic efficiency, Power management.
3- **Real-time operating systems:** Fundamental concepts.
4- **Interfacing and Communication:** Introduction to interfacing, Synchronous serial interface SSI, LCD interface, Scanned keyboard, I^2C interface, USB interface, High speed interfacing: Hardware FIFO, Dual-port memory, DMA controllers.
5- **Interrupt programming and real-time systems:** I/O synchronization, Interrupt concepts, Polled I/O vs. interrupt-driven I/O, Timer periodic interrupt, Ballast code timing, Multithreading.
6- **Analog I/O Interfacing:** Real-time data acquisition, 4~20mA signal standards.
7- **Networked embedded systems:** Networked embedded systems, Network topologies: ring, bus, multi-hop. Wireless communication, Internet-enabled embedded systems.
8- **High speed networks:** Fundamentals, CAN, Ethernet, Internet of Things.

Lecture One: Introduction

1. Embedded Systems – Overview

System

A system is an arrangement in which all its unit assemble work together according to a set of rules. For example, a watch is a time displaying system. Its components follow a set of rules to show time. If one of its parts fails, the watch will stop working. So we can say, in a system, all its subcomponents depend on each other.

Embedded System

An embedded system can be an independent system or it can be a part of a large system. *An embedded system is a microcontroller or microprocessor based system which is designed to perform a specific task.* For example, a fire alarm is an embedded system; it will sense only smoke.

An embedded system has three components:

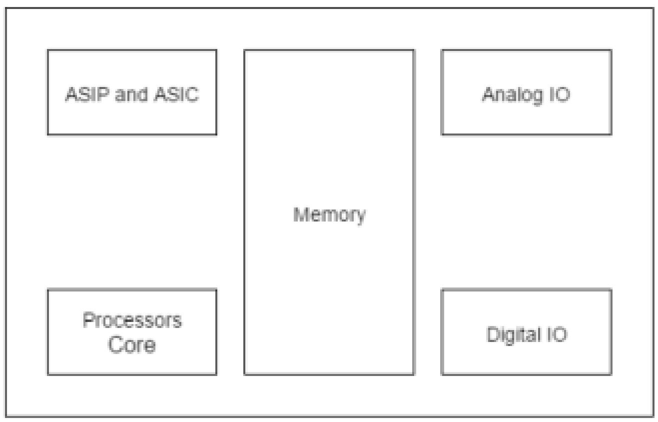
- It has hardware.
- It has application software.
- It has Real Time Operating system (RTOS) that supervises the application software and provide mechanism to let the processor run a process. *RTOS defines the way the system works.* A small scale embedded system may not have RTOS.

Characteristics of an Embedded System

- **Single-functioned** – An embedded system usually performs a specialized operation and does the same repeatedly.
- **Tightly constrained** – All computing systems have constraints on design metrics such as its cost, size, power, and performance, but those on an embedded system can be especially tight. It must be of a size to fit on a single chip, must perform fast enough to process data in real time and consume minimum power to extend battery life.
- **Reactive and real time** – Many embedded systems must continually react to changes in the system's environment and must compute certain results in real time without any delay. Consider an example of a car cruise controller; it continually monitors and reacts

to speed and brake sensors. It must compute acceleration or de-accelerations repeatedly within a limited time; a delayed computation can result in failure to control of the car.

- **Microprocessors based** – It must be microprocessor or microcontroller based.
- **Memory** – It must have a memory, as its software usually embeds in ROM. It does not need any secondary memories in the computer.
- **Connected** – It must have connected peripherals to connect input and output devices.
- **HW-SW systems** – Software is used for more features and flexibility. Hardware is used for performance and security.

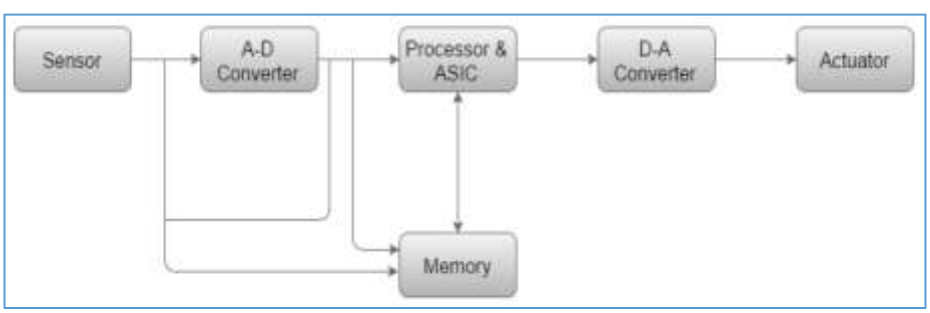


Advantages

- Easily Customizable
- Low power consumption
- Low cost
- Enhanced performance

Basic Structure of an Embedded System

The following illustration shows the basic structure of an embedded system:



- **Sensor** – It measures the physical quantity and converts it to an electrical signal which can be read by an observer or by any electronic instrument like an A2D converter. A sensor stores the measured quantity to the memory.
- **A-D Converter** – An analog-to-digital converter converts the analog signal sent by the sensor into a digital signal.
- **Processor & ASICs (**Application Specific Integrated Circuit) – Processors process the data to measure the input and store it to the memory.
- **D-A Converter** – A digital-to-analog converter converts the digital data fed by the processor to analog data.
- **Actuator** – An actuator compares the output given by the D-A Converter to the actual (expected) output stored in it and stores the approved output.

2. Embedded Systems – Processors

Processor is the heart of an embedded system. It is the basic unit that takes inputs and produces an output after processing the data. For an embedded system designer, it is necessary to have the knowledge of both microprocessors and microcontrollers.

Processors in a System

A processor has two essential units:

- Program Flow Control Unit (CU)
- Execution Unit (EU)

The CU includes a fetch unit for fetching instructions from the memory. The EU includes the Arithmetic and Logical Unit (ALU) and also the circuits that execute instructions for a program control task such as interrupt, or jump to another set of instructions. A processor runs the cycles of fetch and executes the instructions in the same sequence as they are fetched from memory.

Types of Processors

Processors can be of the following categories:

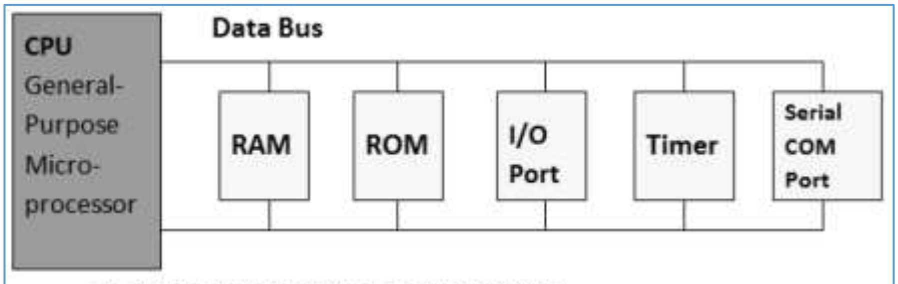
- General Purpose Processor (GPP)
    - Microprocessor
    - Microcontroller
    - Embedded Processor
    - Digital Signal Processor
    - Media Processor

- Application Specific System Processor (ASSP)
- Application Specific Instruction Processors (ASIPs)

Microprocessor

A microprocessor is a single VLSI chip having a CPU. In addition, it may also have other units such as coaches, floating point processing arithmetic unit, and pipelining units that help in faster processing of instructions.

Earlier generation microprocessors' fetch-and-execute cycle was guided by a clock frequency of order of ~1 MHz. Processors now operate at a clock frequency of 2GHz and more.



A SIMPLE BLOCK DIAGRAM OF A MICROPROCESSOR

Microcontroller

A microcontroller is a single-chip VLSI unit (also called microcomputer) which, although having limited computational capabilities, have enhanced input/output capability and a number of on-chip functional units. Microcontrollers are particularly used in embedded systems for real-time control applications with on-chip program memory and devices.

Microcontroller Chip

Microprocessor vs Microcontroller

Microprocessor	Microcontroller
Microprocessors are multitasking in nature. Can perform multiple tasks at a time. For example, on computer we can play music while writing text in text editor.	Single task oriented. For example, a washing machine is designed for washing clothes only.
RAM, ROM, I/O Ports, and Timers can be added externally and can vary in numbers.	RAM, ROM, I/O Ports, and Timers cannot be added externally. These components are to be embedded together on a chip and are fixed in numbers.
Designers can decide the number of memory or I/O ports needed.	Fixed number for memory or I/O makes a microcontroller ideal for a limited but specific task.
External support of external memory and I/O ports makes a microprocessor-based system heavier and costlier.	Microcontrollers are lightweight and cheaper than a microprocessor.
External devices require more space and their power consumption is higher.	A microcontroller-based system consumes less power and takes less space.

3. Embedded Systems – Architecture

When data and code lie in different memory blocks, then the architecture is referred as Harvard architecture. In case data and code lie in the same memory block, then the architecture is referred as Von Neumann architecture.

Von Neumann Architecture

The Von Neumann architecture was first proposed by a computer scientist John von Neumann. In this architecture, one data path or bus exists for both instruction and data. As a result, the CPU does one operation at a time. It either fetches an instruction from memory, or performs read/write operation on data. So an instruction fetch and a data operation cannot occur simultaneously, sharing a common bus.

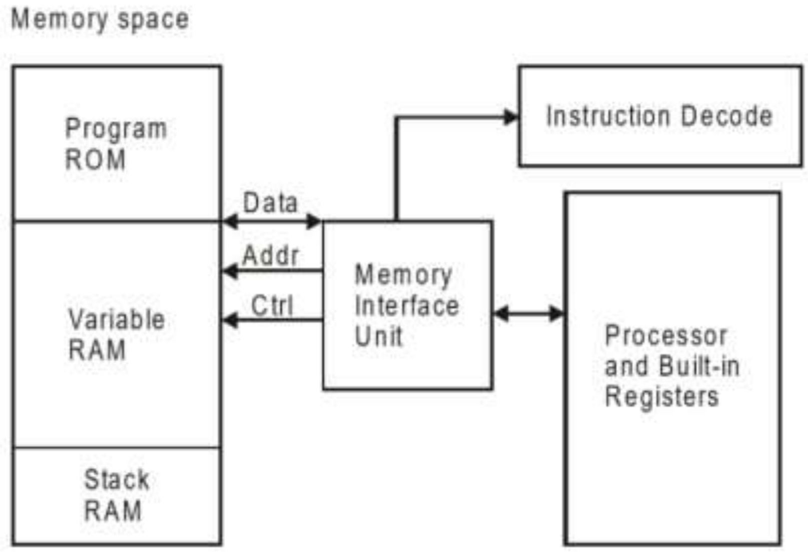


Figure: Von-Neumann Architecture

Von-Neumann architecture supports simple hardware. It allows the use of a single, sequential memory. Today's processing speeds vastly outpace memory access times, and we employ a very fast but small amount of memory (cache) local to the processor.

Harvard Architecture

The Harvard architecture offers separate storage and signal buses for instructions and data. Computers have separate memory areas for program instructions and data using internal data buses, allowing simultaneous access to both instructions and data.

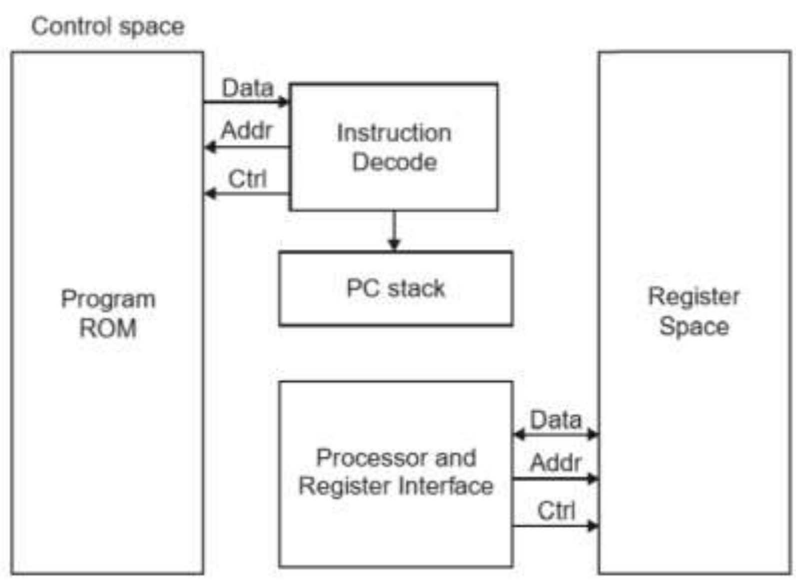


Figure: Harvard Architecture

Von-Neumann Architecture vs Harvard Architecture

Von-Neumann Architecture	Harvard Architecture
Single memory to be shared by both code and data.	Separate memories for code and data.
Processor needs to fetch code in a separate clock cycle and data in another clock cycle. So it requires two clock cycles.	Single clock cycle is sufficient, as separate buses are used to access code and data.
Slower in speed, thus more time-consuming.	Higher in speed, thus less time-consuming.
Simple in design.	Complex in design.
Used in all machines from desktop computers, notebooks, high performance computers to workstations.	It is a new concept used specifically in microcontrollers and digital signal processing (DSP).

CISC (Complex Instruction Set Computer)

It is a computer that can address a large number of instructions. In the early 1980s, computer designers recommended that computers should use fewer instructions with simple constructs so that they can be executed much faster within the CPU without having to use memory. Such computers are classified as Reduced Instruction Set Computer or RISC.