# Embedded Systems- BEC601

## CREDITS: 4

# Course Objectives:

**This course will enable students to:**

1. Identify various components, their purpose, and their application to embedded systems.
2. Program various embedded components using Embedded C.
3. Understand real-time operating systems (RTOS) and their application in IoT.
4. Learn ARM-based system fundamentals, including architecture and key units.
5. Utilize ARM instructions for programming.

# Module 1 - Introduction to Embedded Systems

1. What is an Embedded System?
2. Embedded systems vs. General computing systems
3. History and Classification of Embedded Systems
4. Major Application Areas and Purpose of Embedded Systems
5. Microprocessor vs. Microcontroller

# Module 1 - Introduction to Embedded Systems

6. RISC vs. CISC Architecture
7. Harvard vs. Von Neumann Architecture
8. Big-endian vs. Little-endian Processors
9. Memory (ROM & RAM types), Sensors & Actuators
10. I/O Subsystem & Communication Interfaces
11. Embedded Firmware and Other System Components

# Module 2 - Embedded System Design Concepts

1. Characteristics and Quality Attributes of Embedded Systems
2. Operational and Non-operational Quality Attributes
3. Domain-Specific Applications of Embedded Systems
4. Hardware-Software Co-Design
5. Program Modeling (excluding UML)
6. Embedded Firmware Design & Development (excluding C language)

# Module 3 - RTOS and IDE for Embedded System Design

1. Basics of Operating Systems
2. Types of Operating Systems
3. Task, Process, and Threads
   (Only POSIX Threads with an example program)
4. Thread Preemption and Preemptive Task Scheduling Techniques
5. Task Communication and Synchronization Issues (Racing & Deadlock)
6. Choosing an RTOS
7. Integration and Testing of Embedded Hardware & Firmware
8. Embedded System Development Environment – Block Diagram

# Module 4 - ARM Embedded Systems

1. Introduction to ARM Embedded Systems
2. RISC and ARM Design Philosophy
3. Embedded System Hardware (AMBA Bus Protocol, ARM Bus Technology, Memory, Peripherals)
4. Embedded System Software (Boot Code, OS, Applications)
5. ARM Processor Fundamentals (Dataflow Model, Registers, CPSR, Pipeline, Interrupts, Vector Table, Core Extensions)

# Module 5 - ARM Instruction Set

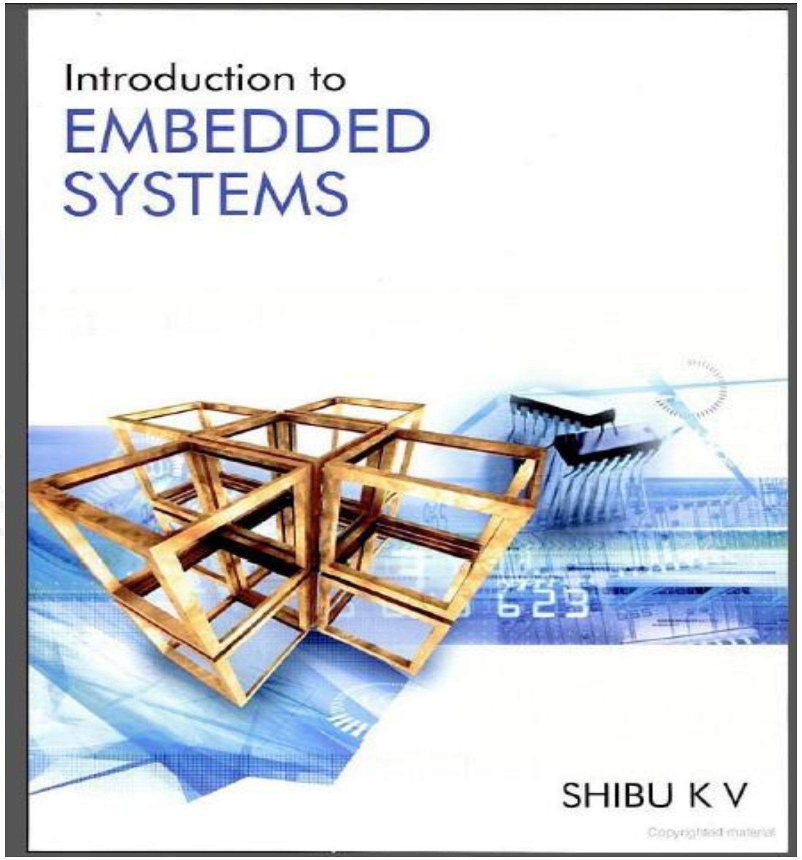
1. Introduction to ARM Instruction Set
2. Data Processing Instructions
3. Load-Store Instructions
4. Software Interrupt Instructions
5. Program Status Register Instructions
6. Loading Constants
7. ARMv5E Extensions
8. Conditional Execution

# Course Outcomes

1. Understand the selection of basic ***hardware components*** for embedded systems.
2. Explain *hardware-software co-design and firmware design* approaches.
3. Justify the need for ***real-time operating systems*** in embedded applications.
4. Describe ***ARM Cortex M3 architecture and instructions***.
5. Apply ***ARM Cortex M3 programming*** knowledge for different applications.

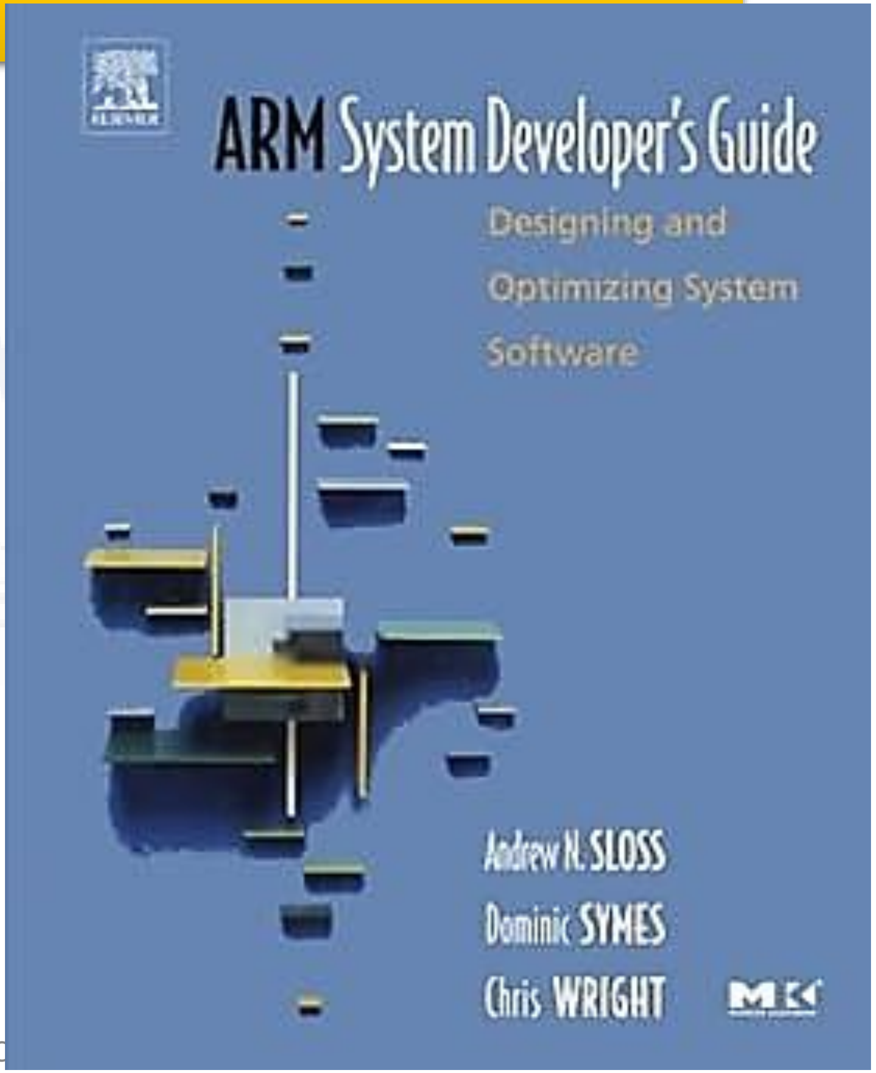
# Recommended Textbooks

***Shibu K V***, **"Introduction to Embedded Systems"**
Tata McGraw Hill Education Private Limited, 2nd Edition



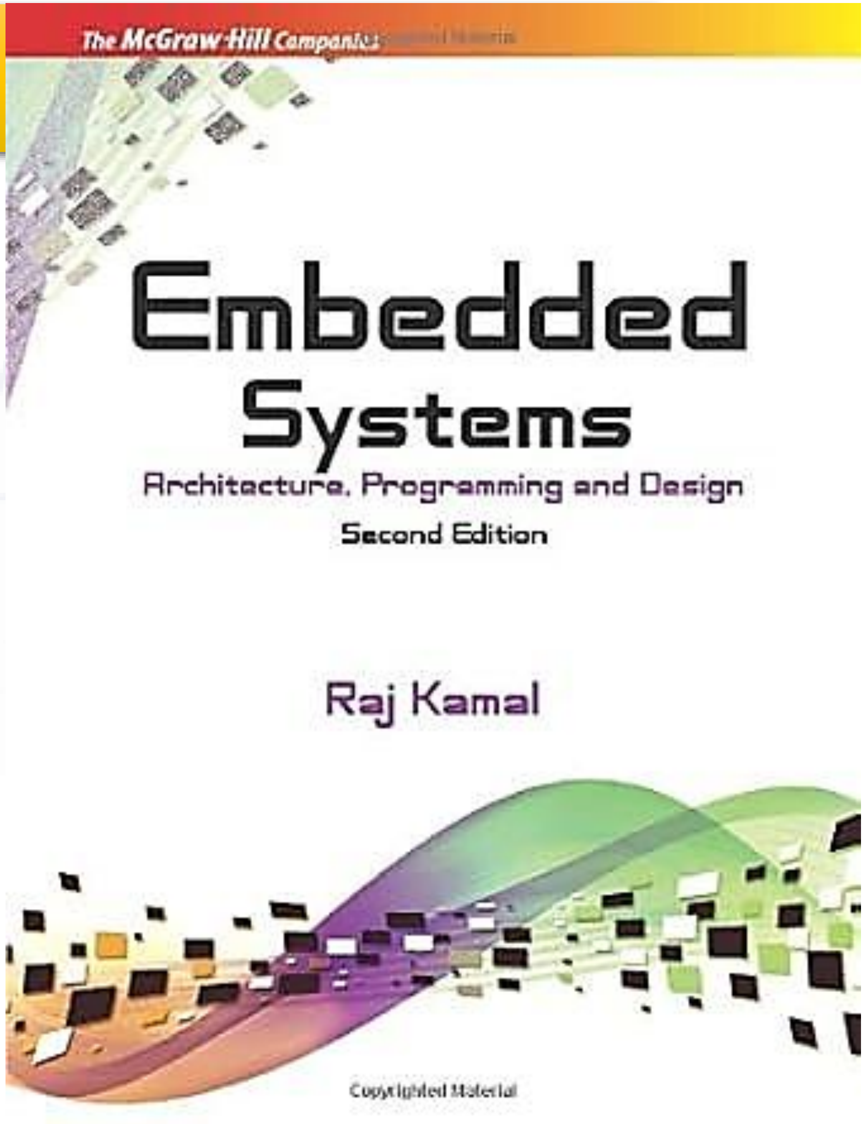
# Recommended Textbooks

***Andrew N Sloss*, “Dominic System, Chris Wright, 'ARM System Developer’s Guide”,** Elsevier, 2008.



# Reference Book

***Raj Kamal,*** **"Embedded Systems: Architecture and Programming"**, Tata McGraw Hill, 2008.



# URLs

- NPTEL Course on Embedded Systems
- ARM Instruction Reference
- Udemy: Introduction to ARM Cortex M3 and M4
- Nuvoton Advanced ARM Cortex Processors
- Alison Embedded Systems Courses

Embedded Systems-BEC601

## Module-1

# Introduction to Embedded System

Pradeep Kumar Y
Assistant Professor
Dept.of ECE

# Topics

1. Embedded v/s General computing system
2. Classification of Embedded Systems
3. Major applications and purpose of ES
4. Elements of an Embedded System

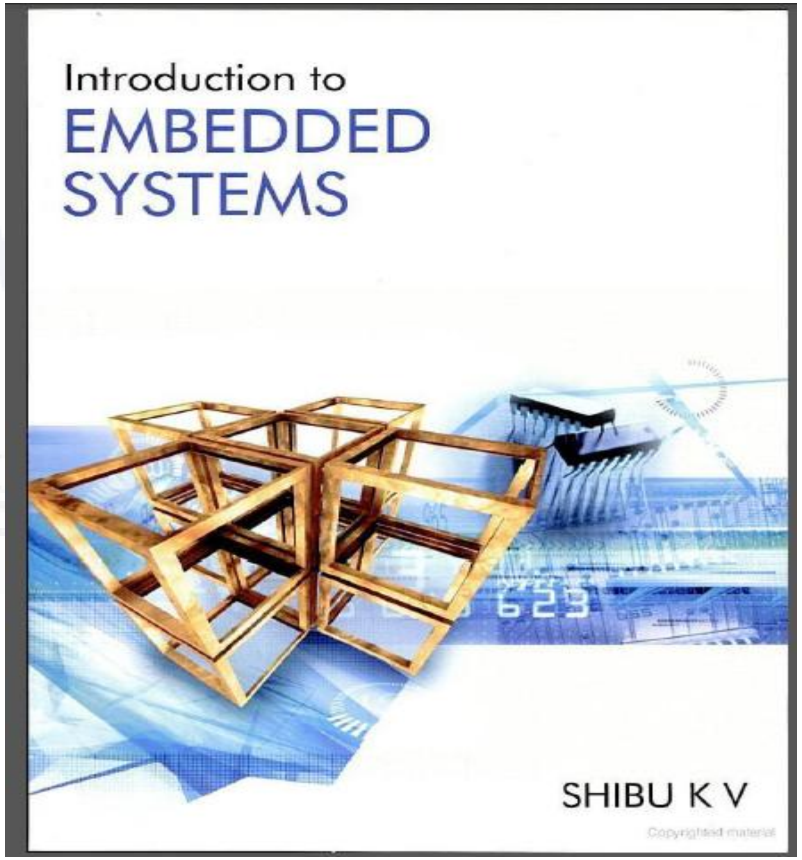
# Topics

5. Differences between RISC and CISC
6. Harvard and Von-Neumann
7. Big and Little Endian formats,
8. Memory (ROM and RAM types)
9. Sensors, Actuators, Optocoupler
10. Communication Interfaces (I2C, SPI, IrDA, Bluetooth, Wi-Fi, Zigbee)

# Text Book

***Shibu K V*, “Introduction to Embedded Systems”**

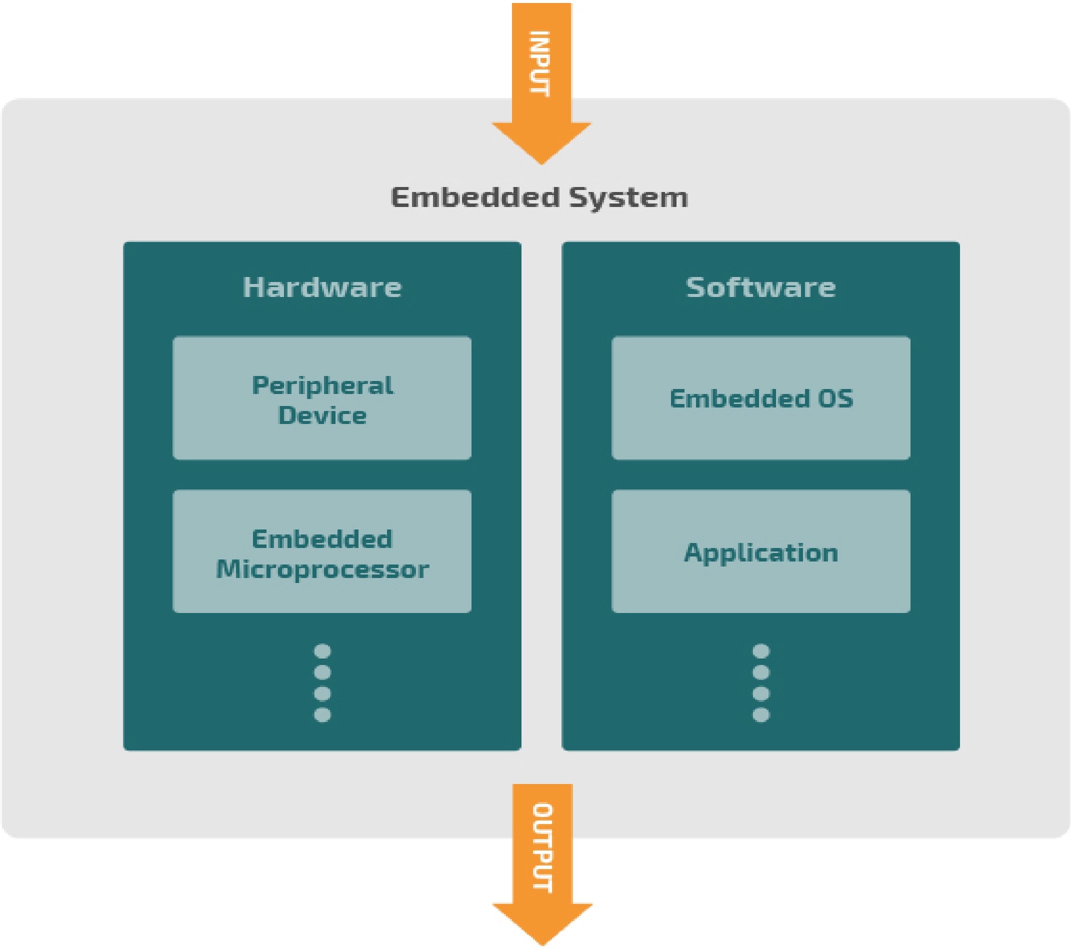
Tata McGraw Hill Education Private Limited, 2nd Edition



# Introduction to Embedded System

## What is Embedded System?

An Electronic/Electro mechanical system which is designed to perform a specific function and is a combination of both hardware and firmware (Software).




**Embedded Systems are:**

- Unique in character and behavior
- With specialized hardware and software

# Introduction to Embedded System

## Examples of Embedded System

- Electronic Toys
- Mobile Handsets
- Washing Machines
- Air Conditioners
- Automotive Control Units
- Set Top Box
- DVD Player etc…












# Introduction to Embedded System

## Examples of Embedded System


# General Computing Systems v/s Embedded Systems

## General Computing Systems


## Embedded Systems

# General Computing Systems v/s Embedded Systems

| Criteria | General Purpose Computing System | Embedded System |
|---|---|---|
| **Components** | A system which is a combination of a Generic Hardware and a General Purpose Operating System for executing a variety of applications. | A system which is a combination of special purpose hardware and embedded OS for executing a specific set of applications. |
| **OS** | It contains a general purpose operating system (GPOS). | It may or not contain an operating system for functioning. |
| **Application Alteration** | Applications are alterable (programmable) by the user. (It is possible for the end user to re- install the OS and also add or remove user applications.) | The firmware of the embedded system is pre-programmed and it is non-alterable by the end-user. |

# General Computing Systems v/s Embedded Systems

| Criteria | General Purpose Computing System | Embedded System |
|---|---|---|
| **Key factor** | Performance is the key deciding factor in the selection of the system. Faster is better. | Application specific requirements (like performance, power requirements, memory usage, etc.) are key deciding factors. |
| **Power Consumption** | More | Less |
| **Response Time** | Not critical | Critical for some applications |
| **Execution** | Need not be deterministic | Deterministic for certain types of ES like 'Hard Real Time' systems. |

# Classification of Embedded Systems

- Based on Generation
- Based on Complexity & Performance Requirements
- Based on deterministic behavior
- Based on Triggering

## Classification of Embedded Systems

### Based on Generation

I. **First Generation:** The early embedded systems built around 8bit microprocessors like 8085 and Z80 and 4bit microcontrollers.

II. **Second Generation**: Embedded Systems built around 16bit microprocessors and 8 or 16bit microcontrollers, following the first generation embedded systems.

III. **Third Generation:** Embedded Systems built around high performance 16/32 bit Microprocessors/controllers, Application Specific Instruction set processors like Digital Signal Processors (DSPs), and Application Specific Integrated Circuits (ASICs).

IV. **Fourth Generation:** Embedded Systems built around System on Chips (SoCs), Re-configurable processors and multicore processors.

# Classification of Embedded Systems

## Based on Complexity & Performance

- **Small Scale:** Simple in application needs and performance requirement are not time critical. Built around 8/16 bit MP/MC.

- **Medium Scale**: Slightly complex in Hw and Fw. Built around low cost 16/32 bit MP/MC/DSP. Contain embedded OS.

- **Large Scale/Complex:** Highly complex Hw and Fw. Used in mission critical application demanding high performance. Built around high performance 32/64 bit RISC MP/MC or SoC or Multi core processors or PLDs. Contains high performance RTOS.

ATME College of Engineering

## Major Application Areas of Embedded System

1. **Consumer Electronics** : Camcorders, Cameras etc.

2. **Household Appliances**: Television, DVD players, Washing machine, Fridge, Microwave Oven etc.

3. **Home Automation and Security Systems**: Air conditioners, sprinklers, Intruder detection alarms, Closed Circuit Television Cameras, Fire alarms etc.

4. **Automotive Industry:** Anti-lock breaking systems (ABS), Engine Control, Ignition Systems, Automatic Navigation Systems etc.

5. **Telecom:** Cellular Telephones, Telephone switches, Handset Multimedia Applications etc

# Major Application Areas of Embedded System


6. **Computer Peripherals**: Printers, Scanners, Fax machines etc.


7. **Computer Networking Systems:** Network Routers, Switches, Hubs, Firewalls etc.

8. **Health Care**: Different Kinds of Scanners, EEG, ECG Machines etc.

9. **Measurement & Instrumentation**: Digital multi meters, Digital CROs, Logic Analyzers PLC systems etc.


10. **Banking & Retail**: Automatic Teller Machines (ATM) and Currency counters, Point of Sales (POS)

11. **Card Readers:** Barcode, Smart Card Readers, Hand held Devices etc.

# Purpose of Embedded Systems

Each Embedded Systems is designed to serve the purpose of any one or a combination of the following tasks.

- ❑Data Collection/Storage/Representation
- ❑Data Communication
- ❑Data (Signal) Processing
- ❑Monitoring
- ❑Control
- ❑Application Specific User Interface

## Purpose of Embedded Systems

### Data Collection/Storage/Representation

✓Performs acquisition of data from the external world.

✓The collected data can be either analog or digital

✓Data collection is usually done for storage, analysis, manipulation and transmission

✓The collected data may be stored directly in the system or may be transmitted to some other systems or it may be processed by the system or it may be deleted instantly after giving a meaningful representation


Digital Camera for Image capturing/storage/display

## Purpose of Embedded Systems

### Data Communication

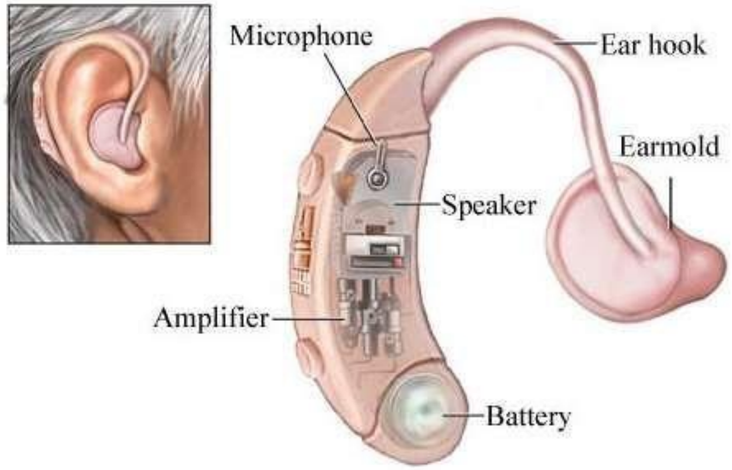
- ✓Embedded Data communication systems are deployed in applications ranging from complex satellite communication systems to simple home networking systems
- ✓Embedded Data communication systems are dedicated for data communication
- ✓The data communication can happen through a wired interface (like Ethernet, RS-232C/USB etc) or wireless interface (like Wi-Fi, GSM,/GPRS, Bluetooth, ZigBee etc)
- ✓Network hubs, Routers, switches, Modems etc are typical examples for dedicated data transmission embedded systems


Wireless Network Router for Data Communication

# Purpose of Embedded Systems

## Data (Signal) Processing

- ✓Embedded systems with Signal processing functionalities are employed in applications demanding signal processing like Speech coding, synthesis, audio video codec, transmission applications etc
- ✓Computational intensive systems
- ✓Employs Digital Signal Processors (DSPs)

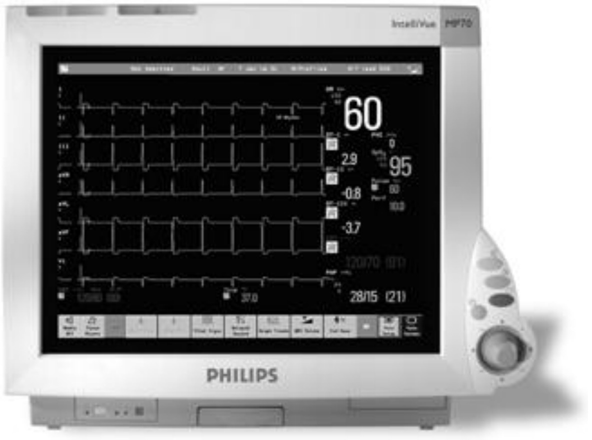



Digital hearing Aid employing Signal Processing Technique

# Purpose of Embedded Systems

## Monitoring

- ✓Embedded systems coming under this category are specifically designed for monitoring purpose
- ✓They are used for determining the state of some variables using input sensors
- ✓They cannot impose control over variables.
- ✓Electro Cardiogram (ECG) machine for monitoring the heart beat of a patient is a typical example for this
- ✓The sensors used in ECG are the different Electrodes connected to the patient's body
- ✓Measuring instruments like Digital CRO, Digital Multi meter, Logic Analyzer etc used in Control & Instrumentation applications are also examples of embedded systems for monitoring purpose




Patient Monitoring system

## Purpose of Embedded Systems

### Control

- ✓Embedded systems with control functionalities are used for imposing control over some variables according to the changes in input variables
- ✓Embedded system with control functionality contains both sensors and actuators
- ✓Sensors are connected to the input port for capturing the changes in environmental variable or measuring variable
- ✓The actuators connected to the output port are controlled according to the changes in input variable to put an impact on the controlling variable to bring the controlled variable to the specified range

## Purpose of Embedded Systems

### Control


Air Conditioner for controlling room temperature

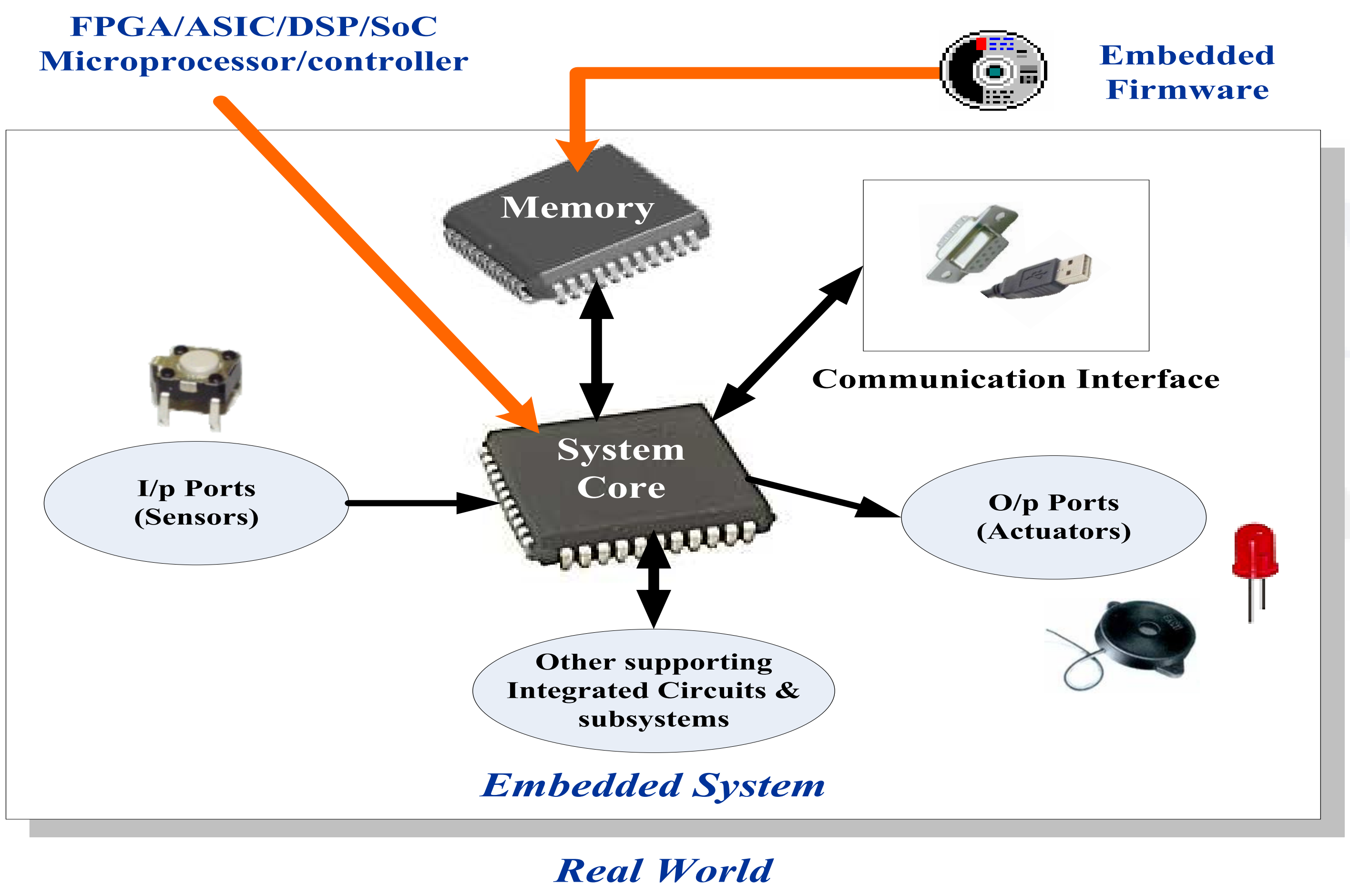
- ✓Air conditioner for controlling room temperature is a typical example for embedded system with ‘Control’ functionality
- ✓Air conditioner contains a room temperature sensing element (sensor) which may be a thermistor and a handheld unit for setting up (feeding) the desired temperature
- ✓The air compressor unit acts as the actuator. The compressor is controlled according to the current room temperature and the desired temperature set by the end user.

# Purpose of Embedded Systems

## Application Specific User Interface

- ✓Embedded systems which are designed for a specific application
- ✓Contains Application Specific User interface (rather than general standard UI ) like key board, Display units etc
- ✓Aimed at a specific target group of users
- ✓Mobile handsets, Control units in industrial applications etc are examples for this

# Elements of an Embedded System



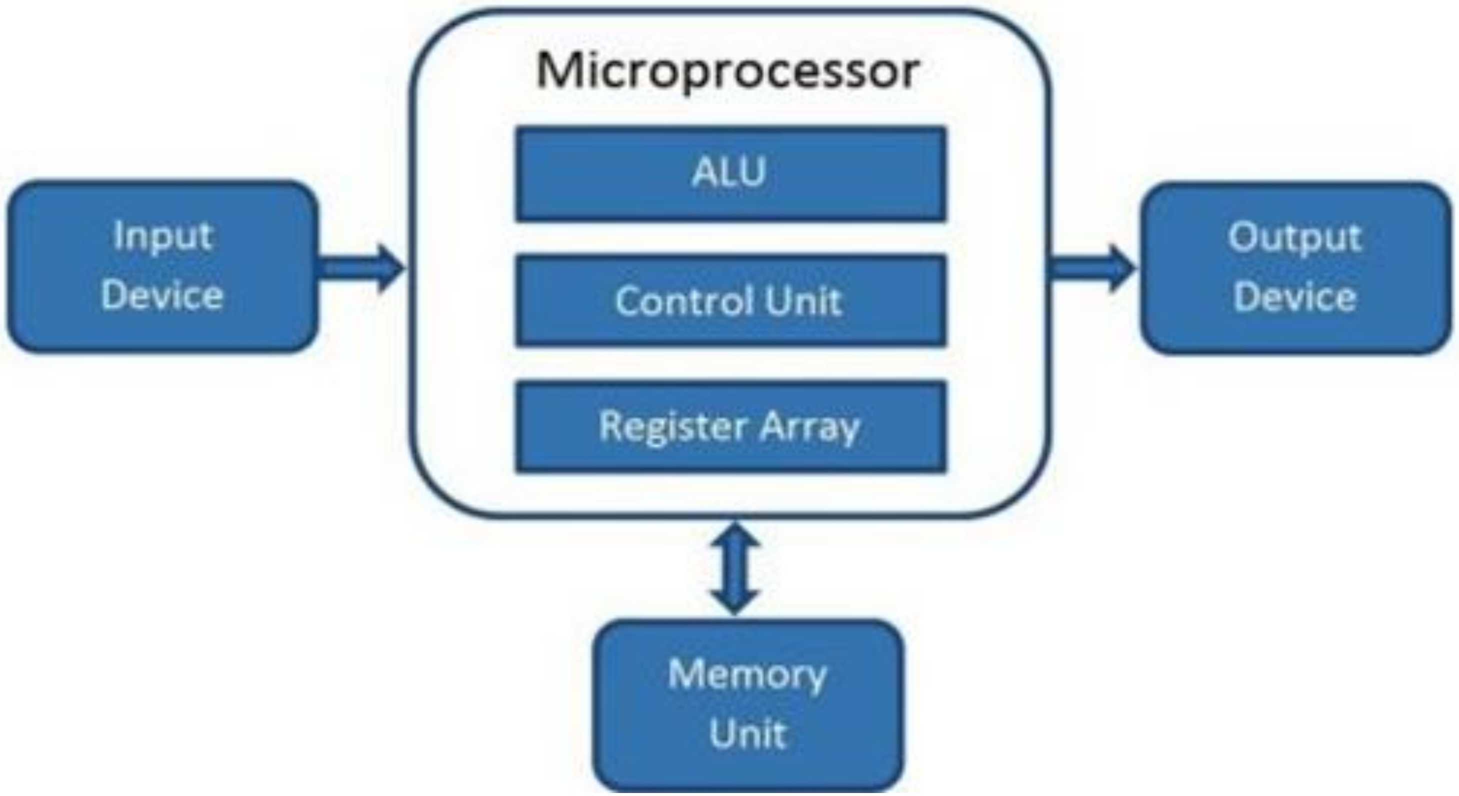

- ✓ **System Core:** can be 8051,8086, FPGA or DSP.
- ✓ **Embedded Firmware**: Designed to regulate a state of device through control signals.
- ✓ **Memory** :For holding the algorithm & other configuration details.
- ✓ **I/O devices:** Sensors and actuators.

# The Core of the Embedded System

1. General Purpose and Domain Specific Processors
   - Microprocessors
   - Microcontrollers
   - Digital Signal Processors
2. Programmable Logic Devices (PLDs)
3. Application Specific Integrated Circuits (ASICs)
4. Commercial off the shelf Components (COTS)

# The Core of the Embedded System

## Microprocessor



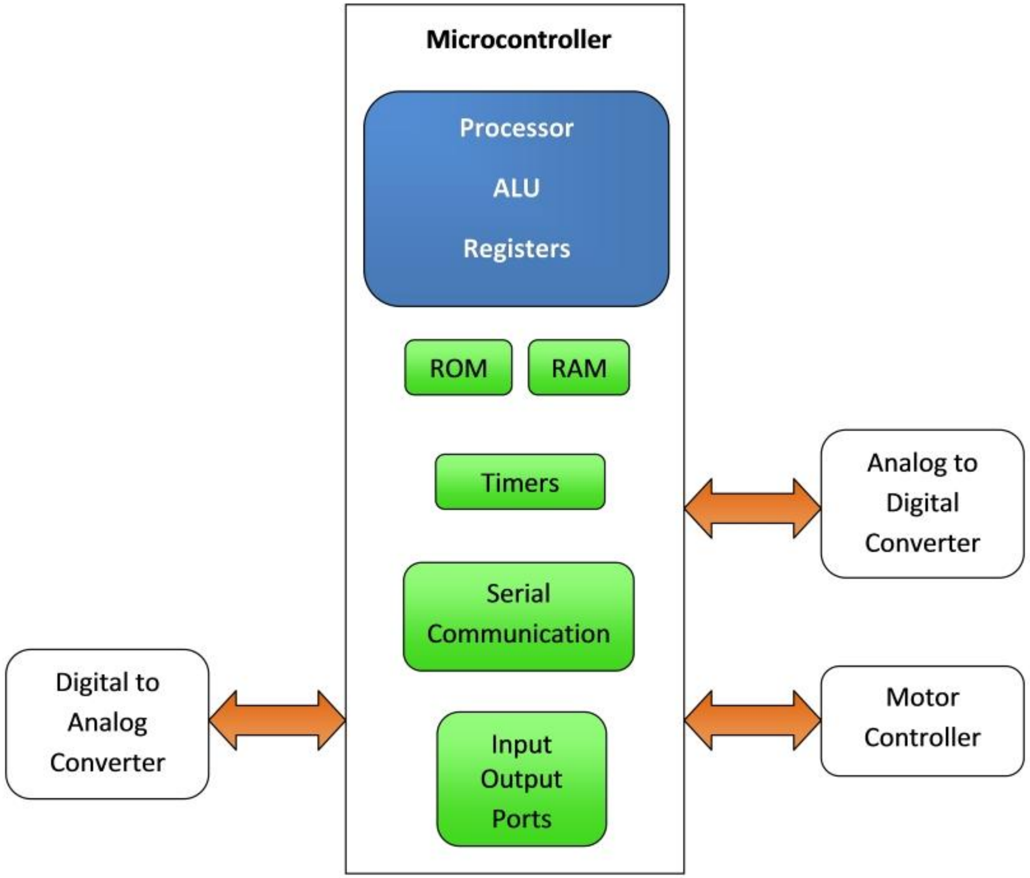
# The Core of the Embedded System

## Microprocessor

- ✓A silicon chip representing a Central Processing Unit (CPU), which is capable of performing arithmetic as well as logical operations according to a pre-defined set of Instructions, which is specific to the manufacturer.
- ✓In general the CPU contains the Arithmetic and Logic Unit (ALU), Control Unit and Working registers.
- ✓Microprocessor is a dependent unit and it requires the combination of other hardware like Memory, Timer Unit, and Interrupt Controller etc for proper functioning.
- ✓Intel claims the credit for developing the first Microprocessor unit Intel 4004, a 4 bit processor which was released in Nov 1971.

# The Core of the Embedded System

## Microcontroller



# The Core of the Embedded System

## Microcontroller

- ✓A highly integrated silicon chip containing a CPU, scratch pad RAM, Special and General purpose Register Arrays, On Chip ROM/FLASH memory for program storage, Timer and Interrupt control units and dedicated I/O ports
- ✓Microcontrollers can be considered as a super set of Microprocessors
- ✓Microcontroller can be general purpose (like Intel 8051, designed for generic applications and domains) or application specific (Like Automotive AVR from Atmel Corporation. Designed specifically for automotive applications)

# The Core of the Embedded System

## Microcontroller

- ✓Since a microcontroller contains all the necessary functional blocks for independent working, they found greater place in the embedded domain in place of microprocessors
- ✓Microcontrollers are cheap, cost effective and are readily available in the market
- ✓Texas Instruments TMS 1000 is considered as the world's first microcontroller

# Microprocessor Vs Microcontroller

| Microprocessor | Microcontroller |
| --- | --- |
| A silicon chip representing a Central Processing Unit (CPU), which is capable of performing arithmetic as well as logical operations according to a pre-defined set of Instructions | A microcontroller is a highly integrated chip that contains a CPU, scratch pad RAM, Special and General purpose Register Arrays, On Chip ROM/FLASH memory for program storage, Timer and Interrupt control units and dedicated I/O ports |
| It is a dependent unit. It requires the combination of other chips like Timers, Program and data memory chips, Interrupt controllers etc for functioning | It is a self contained unit and it doesn't require external Interrupt Controller, Timer, UART etc for its functioning |
| Most of the time general purpose in design and operation | Mostly application oriented or domain specific |
| Doesn't contain a built in I/O port. The I/O Port functionality needs to be implemented with the help of external Programmable Peripheral Interface Chips like 8255 | Most of the processors contain multiple built-in I/O ports which can be operated as a single 8 or 16 or 32 bit Port or as individual port pins |
| Targeted for high end market where performance is important | Targeted for embedded market where performance is not so critical (At present this demarcation is invalid) |
| Limited power saving options compared to microcontrollers | Includes lot of power saving features |

# The Core of the Embedded System

## Digital Signal Processors (DSPs)

- ✓Powerful special purpose 8/16/32 bit microprocessors designed specifically to meet the computational demands and power constraints of today's embedded audio, video, and communications applications
- ✓ 2 to 3 times faster than the general purpose microprocessors in signal processing applications
- ✓Implement algorithms in hardware which speeds up the execution whereas general purpose processors implement the algorithm in firmware and the speed of execution depends primarily on the clock for the processors

# The Core of the Embedded System

## Digital Signal Processors (DSPs)

- ✓DSP can be viewed as a microchip designed for performing high speed computational operations for 'addition', 'subtraction', 'multiplication' and 'division'
- ✓A typical Digital Signal Processor incorporates the following key units
  - ✓Program Memory
  - ✓Data Memory
  - ✓Computational Engine
  - ✓I/O Unit
- ✓Audio video signal processing, telecommunication and multimedia applications are typical examples where DSP is employed

# The Core of the Embedded System

**General Purpose Processor (GPP) Vs Application Specific Instruction Set Processor (ASIP)**

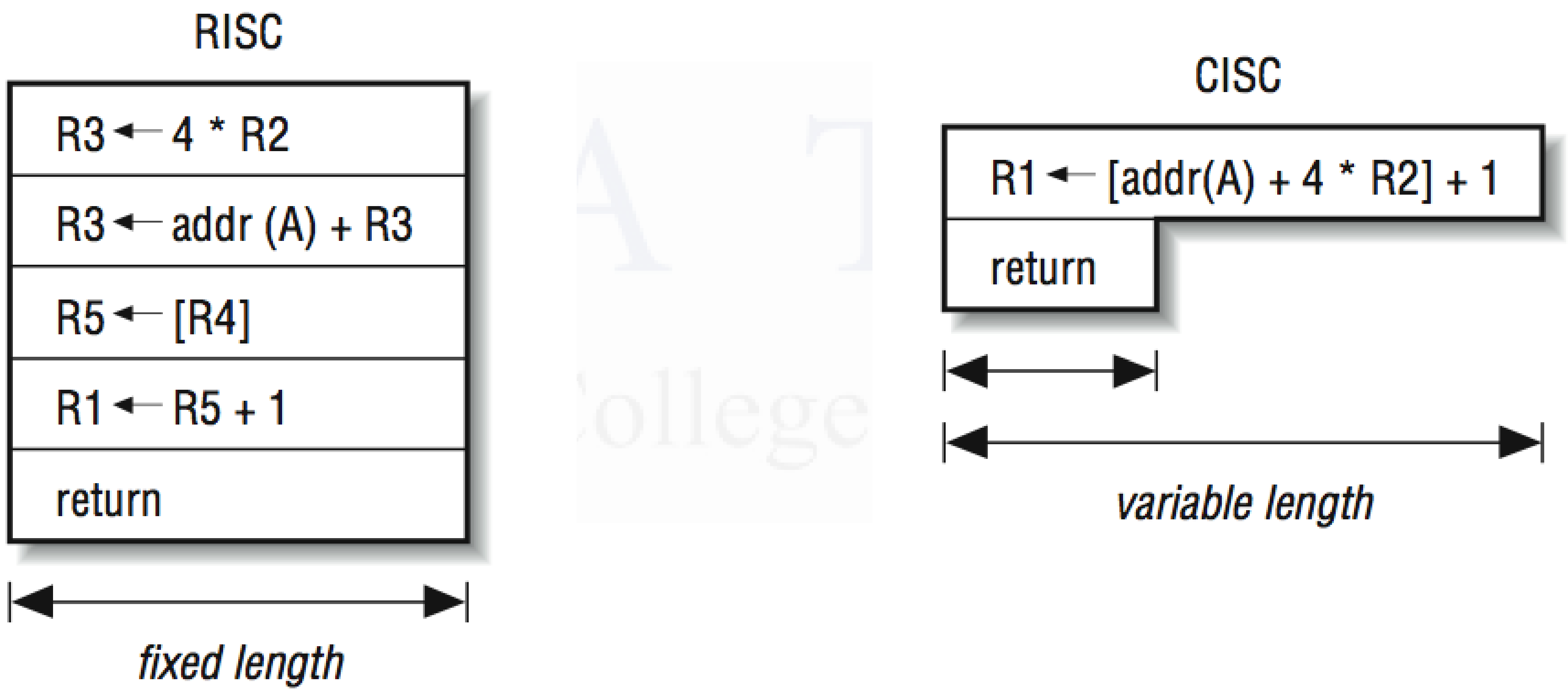
- ✓ General Purpose Processor or GPP is a processor designed for general computational tasks
- ✓ GPPs are produced in large volumes and targeting the general market. Due to the high volume production, the per unit cost for a chip is low compared to ASIC or other specific Ics.
- ✓ A typical general purpose processor contains an Arithmetic and Logic Unit (ALU) and Control Unit (CU)

# The Core of the Embedded System

## General Purpose Processor (GPP) Vs Application Specific Instruction Set Processor (ASIP)

- ✓ Application Specific Instruction Set processors (ASIPs) are processors with architecture and instruction set optimized to specific domain/application requirements like Network processing, Automotive, Telecom, media applications, digital signal processing, control applications etc.
- ✓ ASIPs fill the architectural spectrum between General Purpose Processors and Application Specific Integrated Circuits (ASICs)
- ✓ The need for an ASIP arises when the traditional general purpose processor are unable to meet the increasing application needs
- ✓ Some Microcontrollers (like Automotive AVR, USB AVR from Atmel), System on Chips, Digital Signal Processors etc are examples of Application Specific Instruction Set Processors (ASIPs)
- ✓ ASIPs incorporate a processor and on-chip peripherals, demanded by the application requirement, program and data memory

## RISC V/s CISC Processors/Controllers



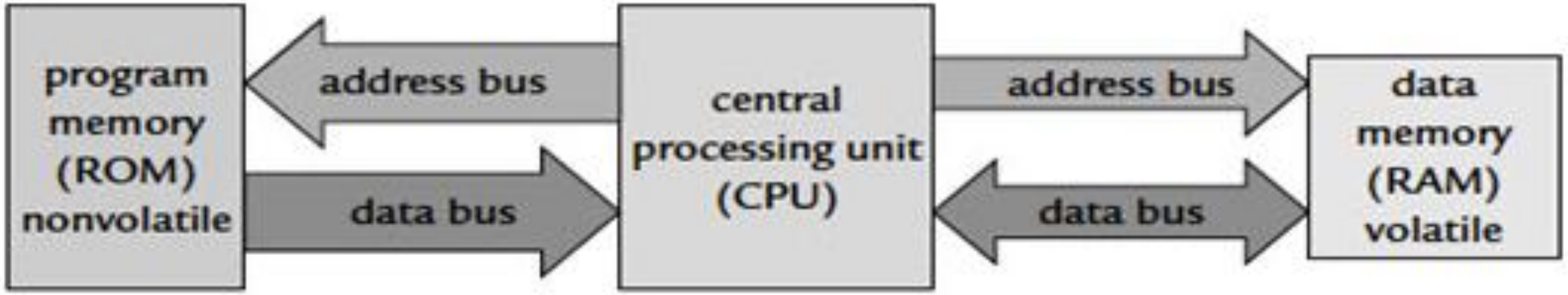
## RISC V/s CISC Processors/Controllers

| RISC | CISC |
|---|---|
| Lesser no. of instructions | Greater no. of Instructions |
| Instruction Pipelining and increased execution speed | Generally no instruction pipelining feature |
| Orthogonal Instruction Set (Allows each instruction to operate on any register and use any addressing mode) | Non Orthogonal Instruction Set (All instructions are not allowed to operate on any register and use any addressing mode. It is instruction specific) |
| Operations are performed on registers only, the only memory operations are load and store | Operations are performed on registers or memory depending on the instruction |
| Large number of registers are available | Limited no. of general purpose registers |

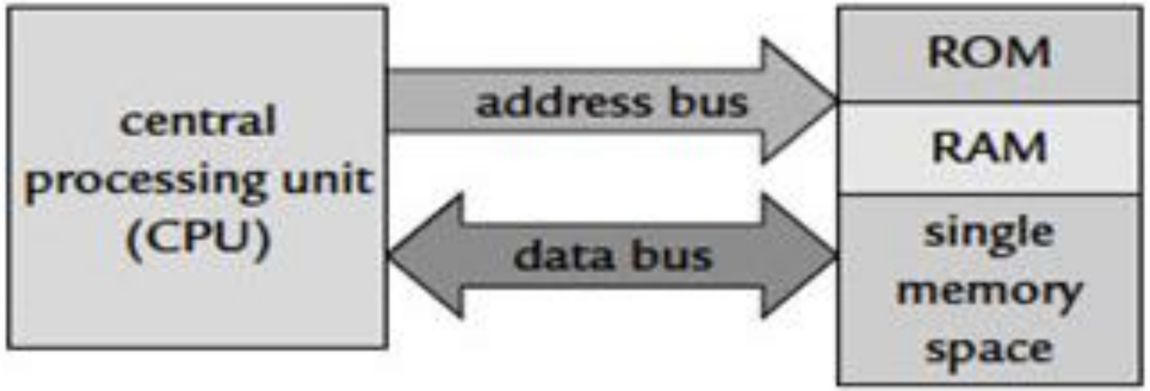
## RISC V/s CISC Processors/Controllers

| RISC | CISC |
|---|---|
| Programmer needs to write more code to execute a task since the instructions are simpler ones | Instructions are like macros in C language. A programmer can achieve the desired functionality with a single instruction which in turn provides the effect of using more simpler single instructions in RISC |
| Single, Fixed length Instructions | Variable length Instructions |
| Less Silicon usage and pin count | More silicon usage since more additional decoder logic is required to implement the complex instruction decoding. |
| With Harvard Architecture | Can be Harvard or Von-Neumann Architecture |

# Harvard V/s Von-Neumann Processor/Controller Architecture




Harvard Architecture




Von-Neumann (Princeton) Architecture

# Harvard V/s Von-Neumann Processor/Controller Architecture

| Harvard Architecture | Von-Neumann Architecture |
|---|---|
| Separate buses for Instruction and Data fetching | Single shared bus for Instruction and Data fetching |
| Easier to Pipeline, so high performance can be achieved | Low performance Compared to Harvard Architecture |
| Comparatively high cost | Cheaper |
| Since data memory and program memory are stored physically in different locations, no chances for accidental corruption of program memory | Since data memory and program memory are stored physically in same chip, chances for accidental corruption of program memory |

# Big-endian V/s Little-endian processors

Endianness specifies the order in which the data is stored in the memory by processor operations in a multi byte system.

Example : **90AB12CD** (Hexadecimal)

| | | | |
|---|---|---|---|
| Base Address + 0 | Byte 0 | Byte 0 | 0x20000 (Base Address) |
| Base Address + 1 | Byte 1 | Byte 1 | 0x20001 (Base Address + 1) |
| Base Address + 2 | Byte 2 | Byte 2 | 0x20002 (Base Address + 2) |
| Base Address + 3 | Byte 3 | Byte 3 | 0x20003 (Base Address + 3) |

| Address | Value |
|---|---|
| 1000 | CD |
| 1001 | 12 |
| 1002 | AB |
| 1003 | 90 |

**Little-endian Operation:** lower-order byte of the data is stored in memory at the lowest address, and the higher-order byte at the highest address

## Big-endian V/s Little-endian processors

**Big-endian Operation** : higher-order byte of the data is stored in memory at the lowest address, and the lower-order byte at the highest address

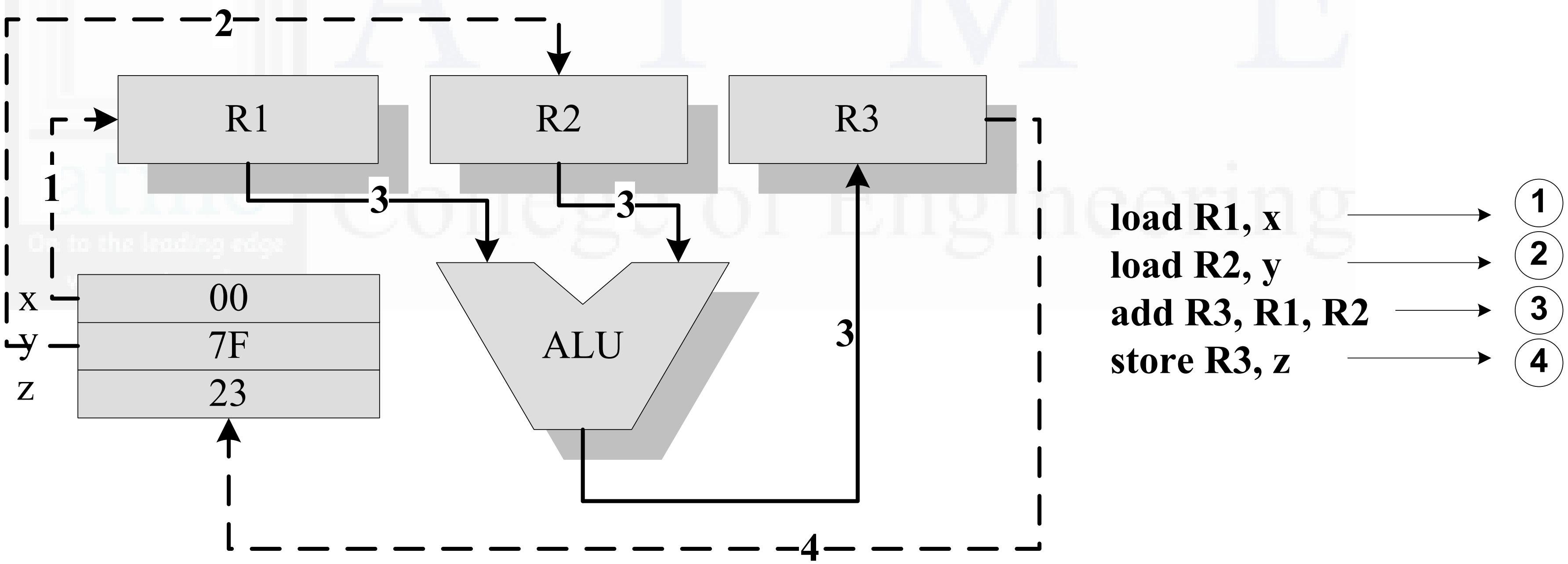
Example : **90AB12CD** (Hexadecimal)

| | | | |
|---|---|---|---|
| Base Address + 0 | Byte 3 | Byte 3 | 0x20000 (Base Address) |
| Base Address + 1 | Byte 2 | Byte 2 | 0x20001 (Base Address + 1) |
| Base Address + 2 | Byte 1 | Byte 1 | 0x20002 (Base Address + 2) |
| Base Address + 3 | Byte 0 | Byte 0 | 0x20003 (Base Address + 3) |

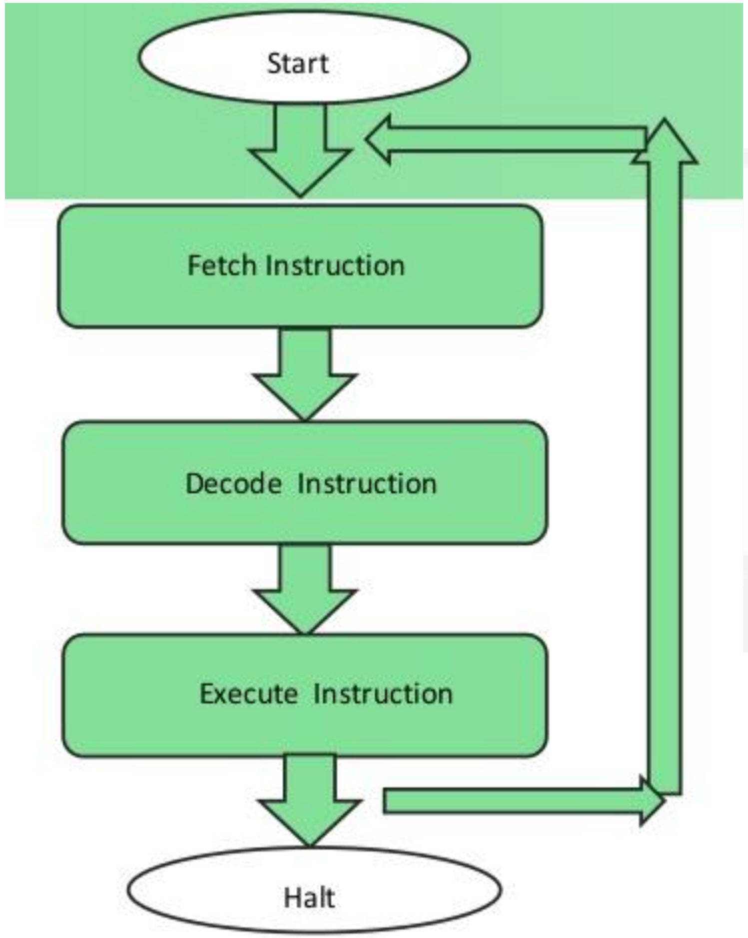
| Address | Value |
|---|---|
| 1000 | 90 |
| 1001 | AB |
| 1002 | 12 |
| 1003 | CD |

# Load Store Operation

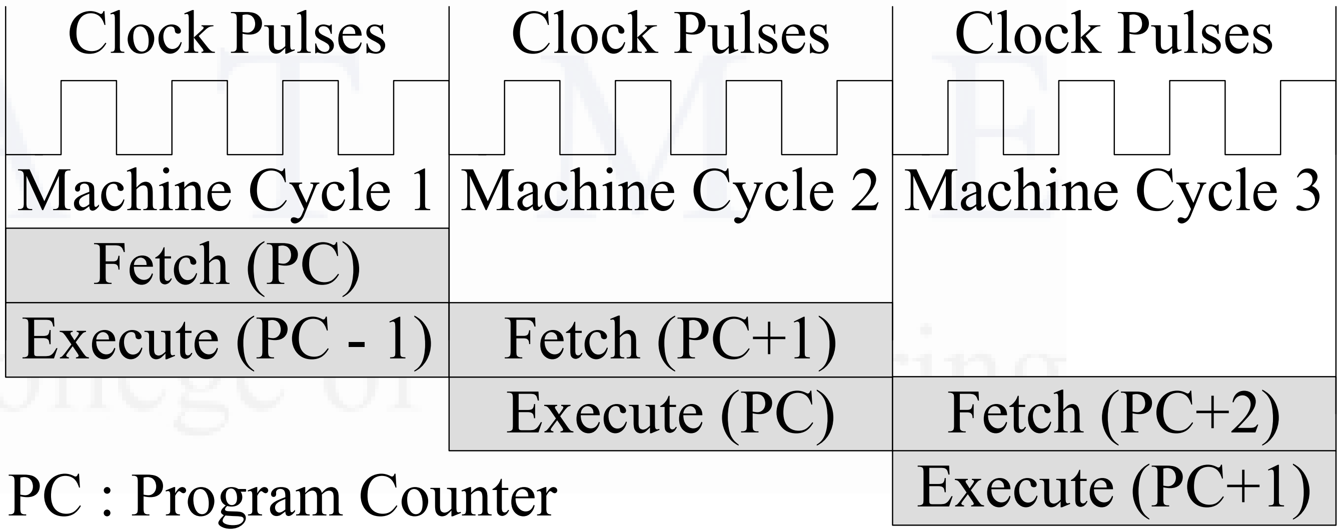
- The RISC processor instruction set is orthogonal and it operates on registers.
- The memory access related operations are performed by the special instructions ***load*** and ***store***



# Instruction Pipelining




Overlapped execution of instructions




**The Single stage pipelining concept**

## Application Specific Integrated Circuits (ASICs)

- Microchip designed to perform a specific or unique applications
- It integrates several functions in to single chip
- Reduces system development cost
- Consumes small area in the total system → Design of Smaller systems
- Can be pre- fabricated or custom fabricated
- Application Specific Standard Product (ASSP)

## Programmable Logic Devices (PLDs)

✓Logic devices provide specific functions, including *device-to-device interfacing*, *data communication, signal processing, data display, timing and control operations*, and almost every other function a system must perform.

✓Logic devices can be classified into two broad categories –

**1. Fixed :** The circuits in a fixed logic device are permanent, they perform one function or set of functions - once manufactured, they cannot be changed

**2.Programmable logic devices (PLDs):** offer customers a wide range of logic capacity, features, speed, and voltage characteristics - and these devices can be re-configured to perform any number of functions at any time

## Programmable Logic Devices (PLDs)

- ✓Designers can use inexpensive software tools to quickly develop, simulate, and test their logic designs in PLD based design
- ✓PLDs are based on re-writable memory technology and the device is reprogrammed to change the design
- ✓The two major types of programmable logic devices

  1. Field Programmable Gate Arrays (FPGAs)
  2. Complex Programmable Logic Devices (CPLDs)

## Programmable Logic Devices (PLDs)-FPGA

- ✓FPGAs offer the highest amount of logic density, the most features, and the highest performance.
- ✓These advanced FPGA devices also offer features such as built-in hardwired processors (such as the IBM Power PC), substantial amounts of memory, clock management systems, and support for many of the latest, very fast device-to-device signaling technologies
- ✓FPGAs are used in a wide variety of applications ranging from data processing and storage, to instrumentation, telecommunications, and digital signal processing

## Programmable Logic Devices (PLDs)-CPLDs

- ✓CPLDs, by contrast, offer much smaller amounts of logic - up to about 10,000 gates
- ✓CPLDs offer very predictable timing characteristics and are therefore ideal for critical control applications
- ✓CPLDs such as the Xilinx **CoolRunner** series also require extremely low amounts of power and are very inexpensive, making them ideal for cost-sensitive, battery-operated, portable applications such as mobile phones and digital handheld assistants

# Programmable Logic Devices (PLDs)-Advantages

- ✓Flexibility in design
- ✓Do not require long time for prototype or production parts
- ✓Do not include large NRE cost
- ✓Customers can order just the number of parts they need
- ✓Reprogrammable

# Commercial off the Shelf Component (COTS)

- ✓A Commercial off-the-shelf (COTS) product is one which is used '*as-is*'
- ✓Designed in such a way to provide easy integration and interoperability with existing system components
- ✓Typical examples for the COTS hardware unit are
  - Remote Controlled Toy Car control unit including the RF Circuitry part
  - High performance, high frequency microwave electronics (2 to 200 GHz),
  - High bandwidth analog-to-digital converters,
  - Devices and components for operation at very high temperatures
  - Electro-optic IR imaging arrays, UV/IR Detectors etc.

## Commercial off the Shelf Component (COTS)

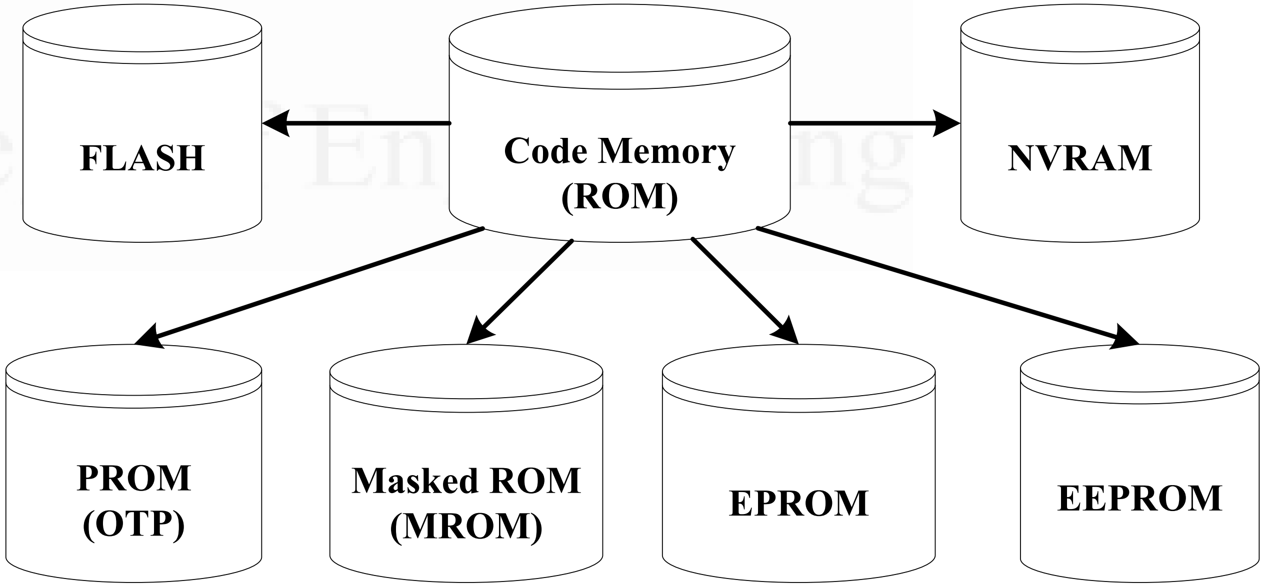
- ✓ A COTS component in turn contains a
  - ❑ General Purpose Processor (GPP)
  - ❑ Application Specific Instruction Set Processor (ASIP)
  - ❑ Application Specific Integrated Chip (ASIC)
  - ❑ Application Specific Standard Product (ASSP)
  - ❑ Programmable Logic Device (PLD)
- ✓ The major advantage of using COTS is that they are readily available in the market, cheap and a developer can cut down his/her development time to a great extend

# MEMORY

- ✓The memory used in embedded system can be
  - ➢Program Storage Memory (ROM)
  - ➢Data memory (RAM)
- ✓Certain Embedded processors/controllers contain built in program memory and data memory and this memory is known as ***on-chip memory***

## MEMORY-Program Storage Memory

- ✓Stores the program instructions
- ✓Retains its contents even after the power to it is turned off. It is generally known as *Non volatile storage* memory
- ✓Depending on the fabrication, erasing and programming techniques they are classified into



## Program Storage Memory- Masked ROM (MROM)

- ✓One-time programmable memory.
- ✓Uses hardwired technology for storing data.
- ✓The device is factory programmed by masking and metallization process according to the data provided by the end user
- ✓The primary advantage of MROM is low cost for high volume production.
- ✓They are the least expensive type of solid state memory
- ✓The limitation is the inability to modify the device firmware against firmware upgrades.

## Program Storage Memory- Programmable Read Only Memory (PROM) / (OTP)

- ✓Unlike MROM it is not pre-programmed by the manufacturer
- ✓PROM/OTP has *nichrome* or *polysilicon* wires arranged in a matrix, these wires can be functionally viewed as fuses
- ✓It is programmed by a PROM programmer which selectively burns the fuses according to the bit pattern to be stored

## Program Storage Memory- Programmable Read Only Memory (PROM) / (OTP)

- ✓It is programmed by a PROM programmer which selectively burns the fuses according to the bit pattern to be stored
- ✓Fuses which are not blown/burned represents a logic “1” where as fuses which are blown/burned represents a logic “0”. The default state is logic “1”
- ✓OTP is widely used for commercial production of embedded systems whose proto-typed versions are proven and the code is finalized
- ✓It is a low cost solution for commercial production. OTPs cannot be reprogrammed

## Program Storage Memory- Erasable Programmable Read Only Memory (EPROM)

- ✓Erasable Programmable Read Only (EPROM) memory gives the flexibility to re-program the same chip
- ✓EPROM stores the bit information by charging the floating gate of an FET
- ✓Bit information is stored by using an EPROM Programmer, which applies high voltage to charge the floating gate
- ✓EPROM contains a quartz crystal window for erasing the stored information. If the window is exposed to Ultra violet rays for a fixed duration, the entire memory will be erased
- ✓Even though the EPROM chip is flexible in terms of re-programmability, it needs to be taken out of the circuit board and needs to be put in a UV eraser device for 20 to 30 minutes

## Program Storage Memory-
## Electrically Erasable Programmable Read Only Memory (EEPROM)

- ✓Erasable Programmable Read Only (EPROM) memory gives the flexibility to re-program the same chip using electrical signals
- ✓The information contained in the EEPROM memory can be altered by using electrical signals at the register/Byte level
- ✓They can be erased and reprogrammed within the circuit
- ✓These chips include a chip erase mode and in this mode they can be erased in a few milliseconds
- ✓It provides greater flexibility for system design
- ✓The only limitation is their capacity is limited when compared with the standard ROM

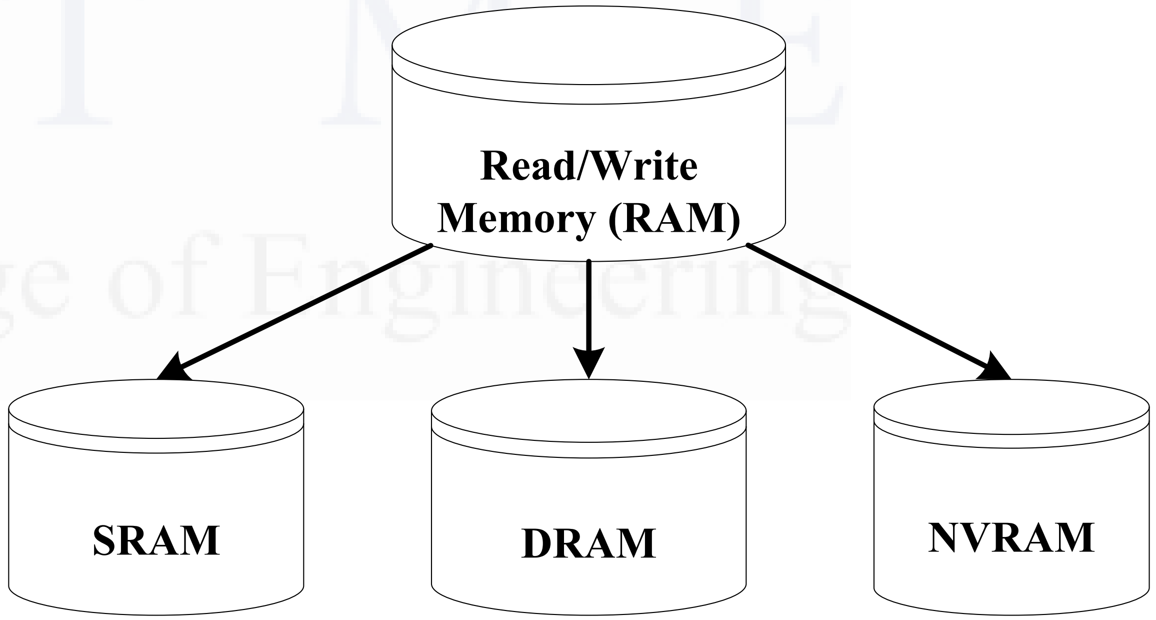
(A few kilobytes)

## Program Storage Memory- FLASH

- ✓FLASH memory is a variation of EEPROM technology
- ✓It combines the re-programmability of EEPROM and the high capacity of standard ROMs
- ✓FLASH memory is organized as sectors (blocks) or pages
- ✓FLASH memory stores information in an array of floating gate MOSFET transistors
- ✓The erasing of memory can be done at sector level or page level without affecting the other sectors or pages
- ✓Each sector/page should be erased before re-programming

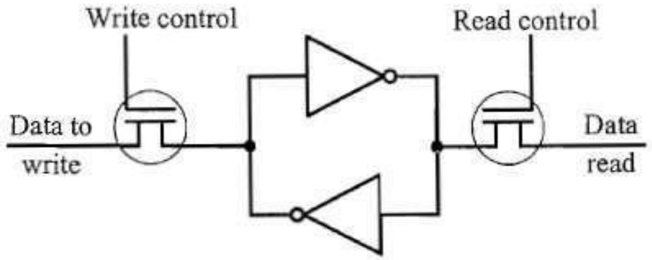
# Memory – Read-Write Memory/Random Access Memory (RAM)

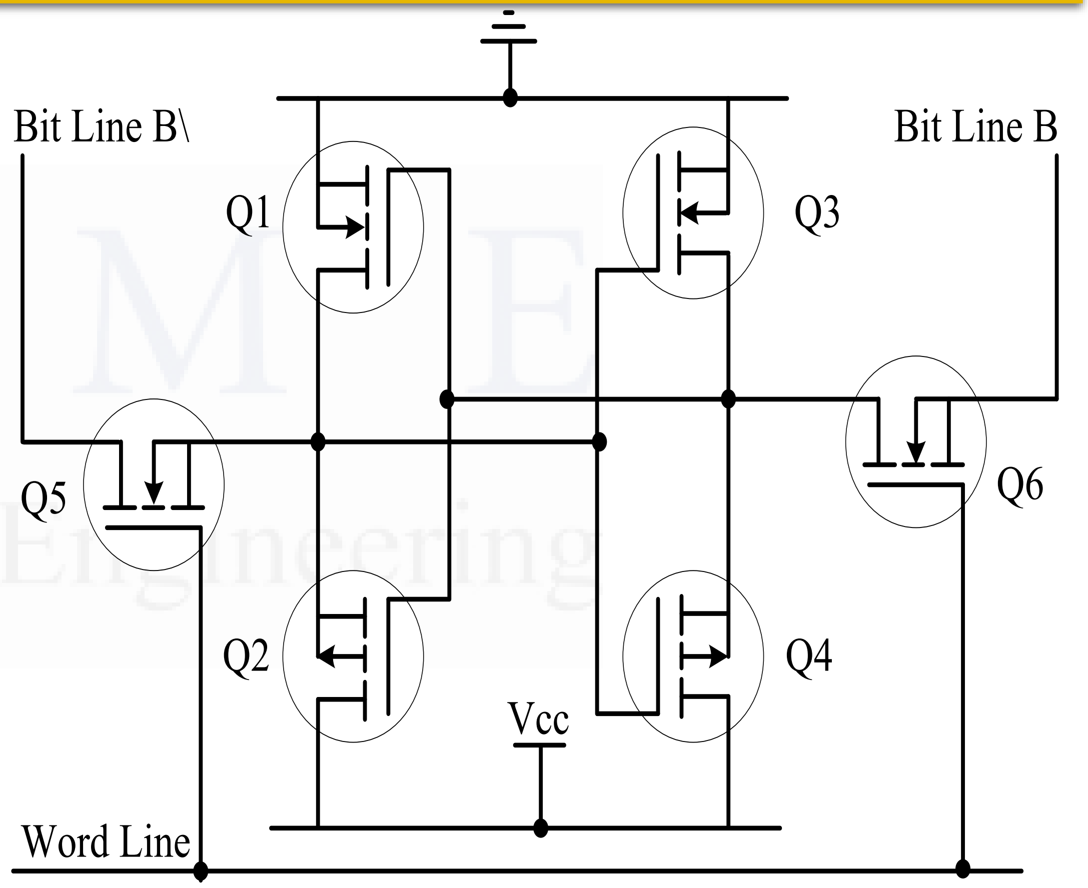
- ✓RAM is the data memory or working memory of the controller/processor
- ✓RAM is volatile, meaning when the power is turned off, all the contents are destroyed
- ✓RAM is a direct access memory



# RAM – Static RAM (SRAM)

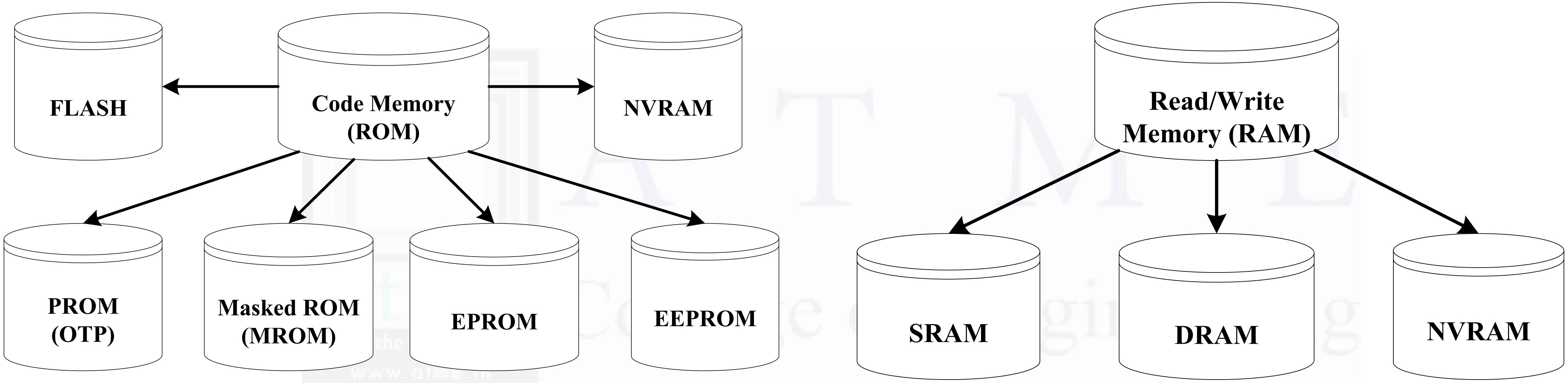
- ✓Static RAM stores data in the form of Voltage.
- ✓They are made up of flip-flops.
- ✓In typical implementation, an SRAM cell (bit) is realized using 6 transistors (or 6 MOSFETs).
- ✓Four of the transistors are used for building the latch (flip-flop) part of the memory cell and 2 for controlling the access.



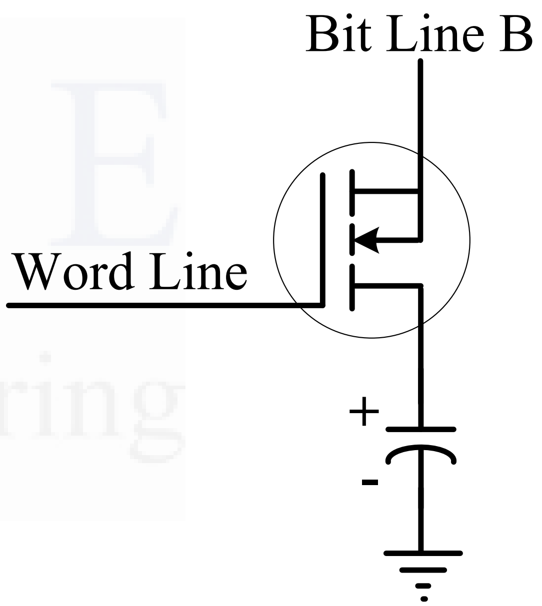

SRAM cell implementation

# MEMORY


FLASH
Code Memory
(ROM)
NVRAM
PROM
(OTP)
Masked ROM
(MROM)
EPROM
EEPROM
Read/Write
Memory (RAM)
SRAM
DRAM
NVRAM

# RAM – Dynamic RAM (DRAM)

- ✓Dynamic RAM stores data in the form of charge. They are made up of MOS transistor gates
- ✓The advantages of DRAM are its high density and low cost compared to SRAM
- ✓The disadvantage is that since the information is stored as charge it gets leaked off with time and to prevent this they need to be refreshed periodically




DRAM cell implementation

# RAM – SRAM Vs DRAM

| SRAM Cell | DRAM Cell |
|---|---|
| Made up of 6 CMOS transistors (MOSFET) | Made up of a MOSFET and a capacitor |
| Doesn’t Require refreshing | Requires refreshing |
| Low capacity (Less dense) | High Capacity (Highly dense) |
| More expensive | Less Expensive |
| Fast in operation. Typical access time is 10ns | Slow in operation due to refresh requirements. Typical access time is 60ns. Write operation is faster than read operation. |

## RAM – Non Volatile RAM (NVRAM)

✓Random access memory with battery backup

✓It contains Static RAM based memory and a minute battery for providing supply to the memory in the absence of external power supply

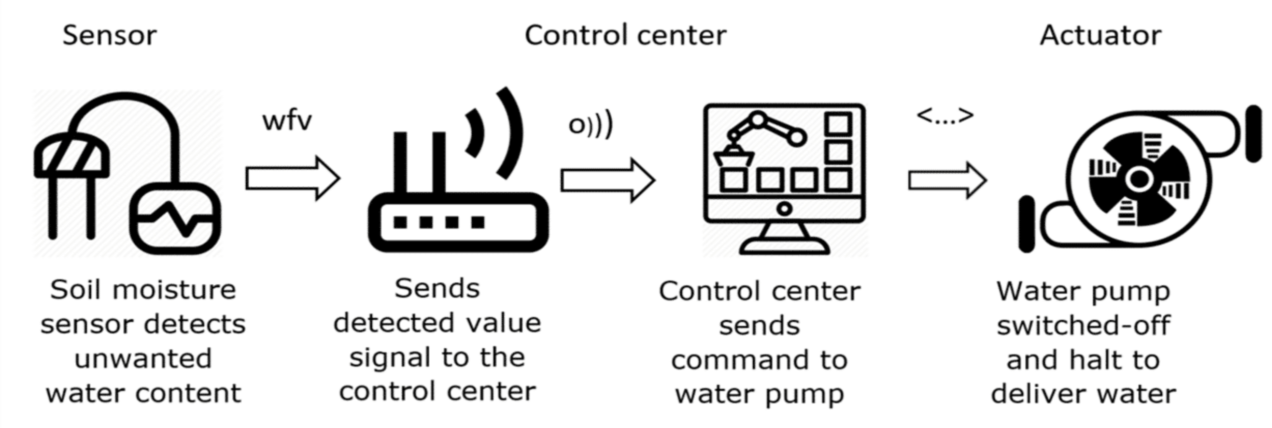
✓The memory and battery are packed together in a single package

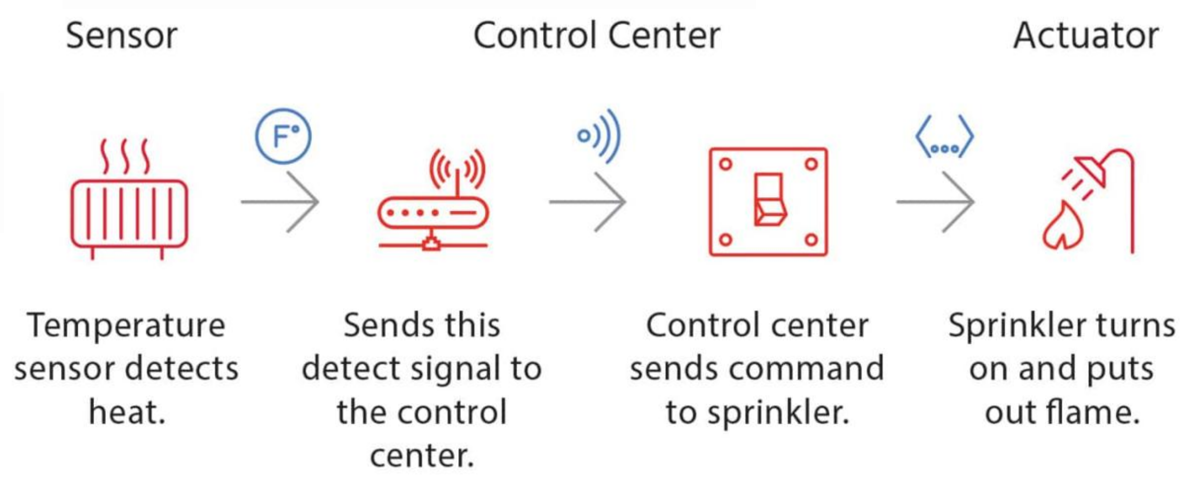
✓NVRAM is used for the non volatile storage of results of operations or for setting up of flags etc

✓The life span of NVRAM is expected to be around 10 years

✓DS1744 from Maxim/Dallas is an example for 32KB NVRAM

# Sensors & Actuators

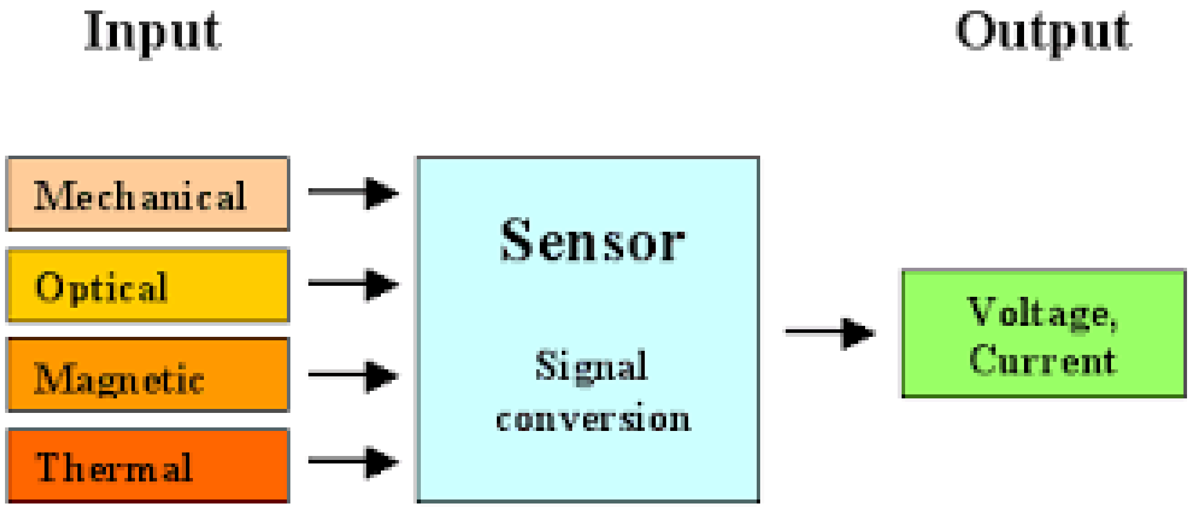


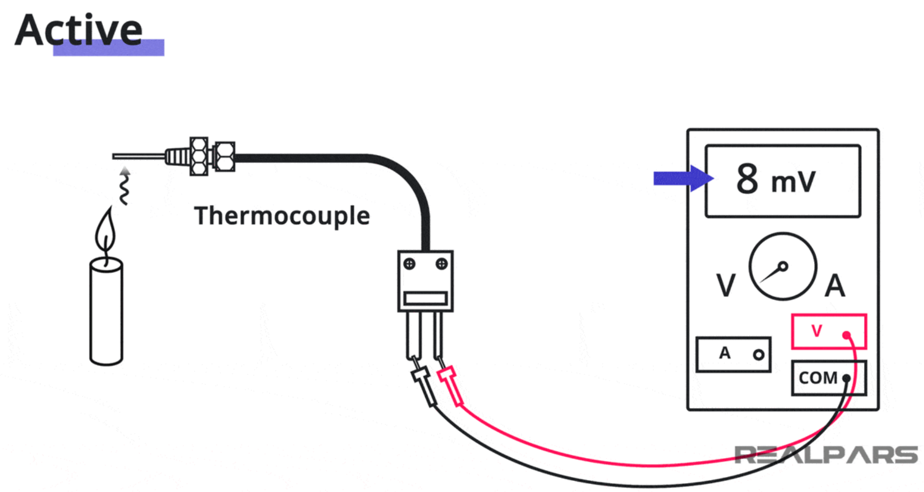



## Sensors & Actuators

# Sensor:

A transducer device which converts energy from one form to another for any measurement or control purpose. Sensors acts as input device



## Sensors & Actuators

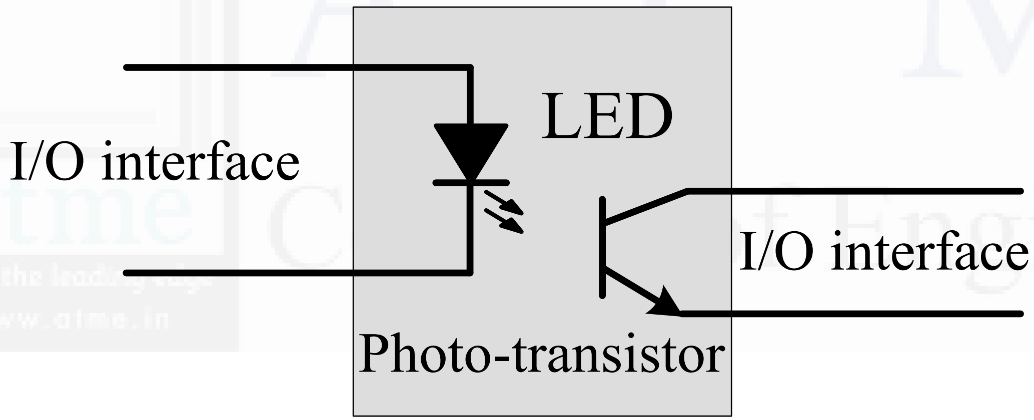
**Actuator:**

- A form of transducer device (mechanical or electrical) which converts signals to corresponding physical action (motion).
- Actuator acts as an output device

# Optocoupler

✓Optocoupler is a solid state device to isolate two parts of a circuit.

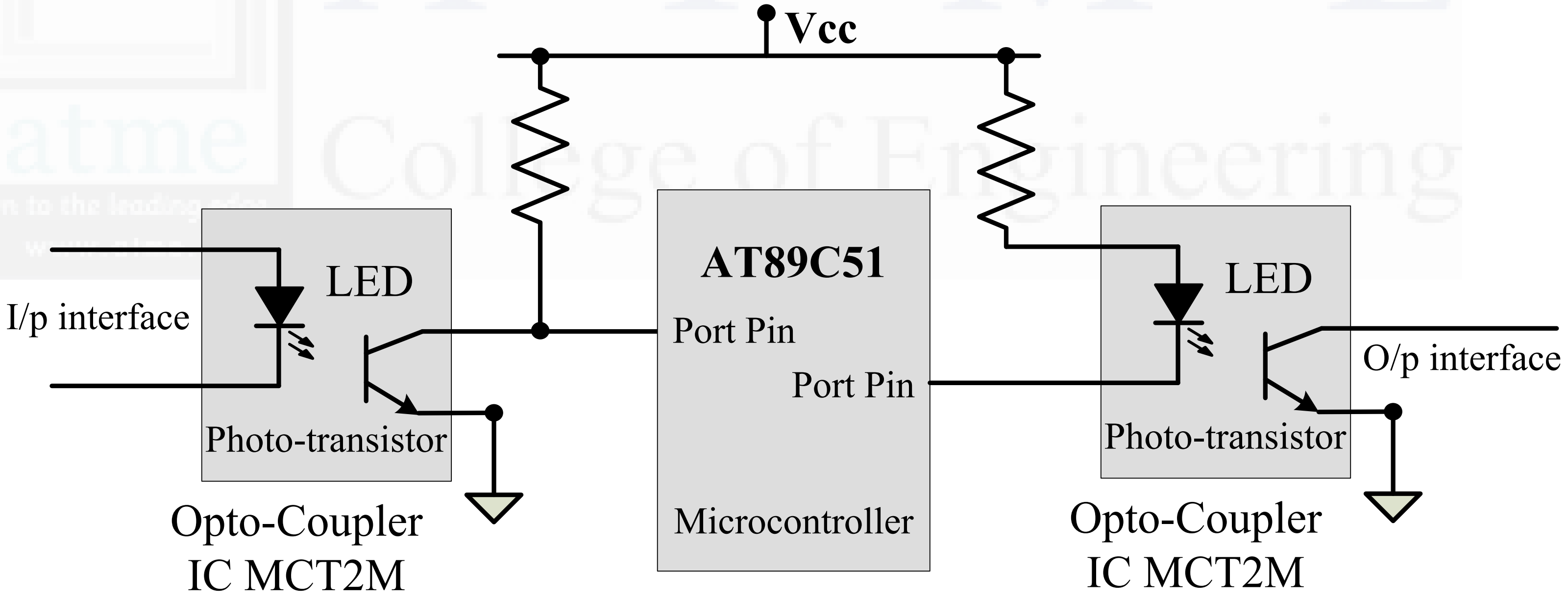
✓Optocoupler combines an LED and a photo-transistor in a single housing (package)



# Optocoupler

✓In electronic circuits, optocoupler is used for suppressing interference in data communication, circuit isolation, High voltage separation, simultaneous separation and intensification signal etc

✓Optocouplers can be used in either input circuits or in output circuits



# The I/O Subsystem

- ✓The I/O subsystem of the embedded system facilitates the interaction of the embedded system with external world
- ✓The interaction happens through the sensors and actuators connected to the Input and output ports respectively of the embedded system
- ✓The sensors may not be directly interfaced to the Input ports, instead they may be interfaced through signal conditioning and translating systems like ADC, Optocouplers etc

# I/O Devices - Light Emitting Diode (LED)

✓Light Emitting Diode (LED) is an output device for visual indication in any embedded system

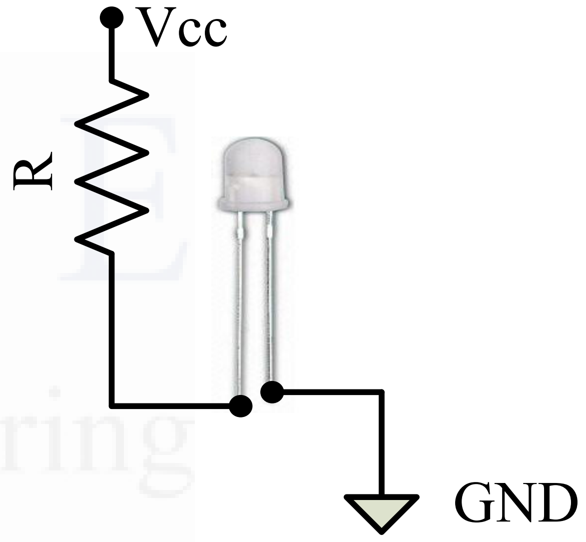
✓LED can be used as an indicator for the status of various signals or situations.

✓Typical examples are indicating the presence of power conditions like ‘Device ON’, ‘Battery low’ or ‘Charging of battery’ for a battery operated handheld embedded devices

# I/O Devices - Light Emitting Diode (LED)

✓LED is a p-n junction diode and it contains an anode and a cathode. For proper functioning of the LED, the anode of it should be connected to +ve terminal of the supply voltage and cathode to the –ve terminal of supply voltage

✓The current flowing through the LED must limited to a value below the maximum current that it can conduct. A resister is used in series between the power supply and the resistor to limit the current through the LED



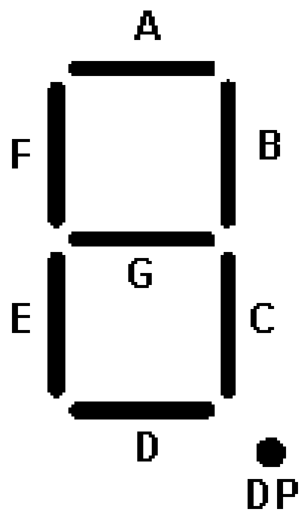
# I/O Devices - 7-Segment LED Display

✓The 7 – segment LED display is an output device for displaying alpha numeric characters

✓It contains 8 light-emitting diode (LED) segments arranged in a special form. Out of the 8 LED segments, 7 are used for displaying alpha numeric characters

✓The LED segments are named A to G and the decimal point LED segment is named as DP

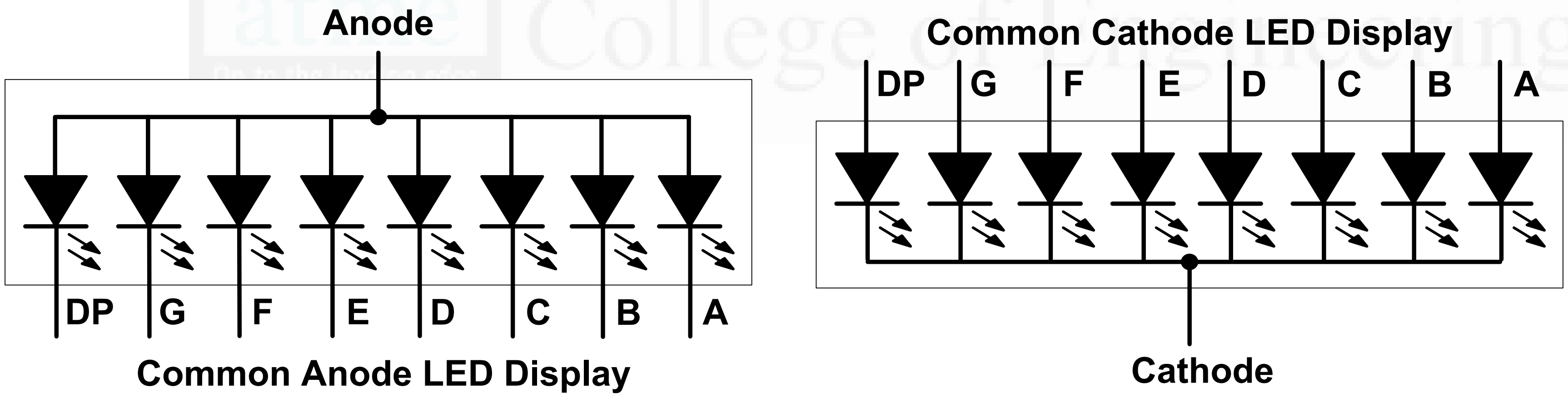
✓The LED Segments A to G and DP should be lit accordingly to display numbers and characters



# I/O Devices - 7-Segment LED Display

✓The 7 – segment LED displays are available in two different configurations, namely; Common anode and Common cathode

✓In the Common anode configuration, the anodes of the 8 segments are connected commonly whereas in the Common cathode configuration, the 8 LED segments share a common cathode line



# Communication Interface

Communication interface is essential for communicating with various subsystems of the embedded system and with the external world

1. Device/board level communication interface (Onboard Communication Interface)
2. Product level communication interface (External Communication Interface)

## Communication Interface- ON BOARD

- ✓ Embedded product is a combination of different types of components (chips/devices) arranged on a Printed Circuit Board (PCB)
- ✓ The communication channel which interconnects the various components within an embedded product is referred as Device/board level communication interface (Onboard Communication Interface)
- ✓ Serial interfaces like I2C, SPI, UART, 1-Wire etc and Parallel bus interface are examples of ‘Onboard Communication Interface’

## Communication Interface – EXTERNAL

✓The ‘Product level communication interface’ (External Communication Interface) is responsible for data transfer between the embedded system and other devices or modules

✓The external communication interface can be either wired media or wireless media and it can be a serial or parallel interface.

✓ Infrared (IR), Bluetooth (BT), Wireless LAN (Wi-Fi), Radio Frequency waves (RF), GPRS etc are examples for wireless communication interface

✓RS-232C/RS-422/RS 485, USB, Ethernet (TCP-IP), IEEE 1394 port, Parallel port, CF-II Slot, SDIO, PCMCIA etc are examples for wired interfaces

## On-board Communication Interface - I2C

✓Inter Integrated Circuit Bus (I2C) is a synchronous bi-directional half duplex two wire serial interface bus

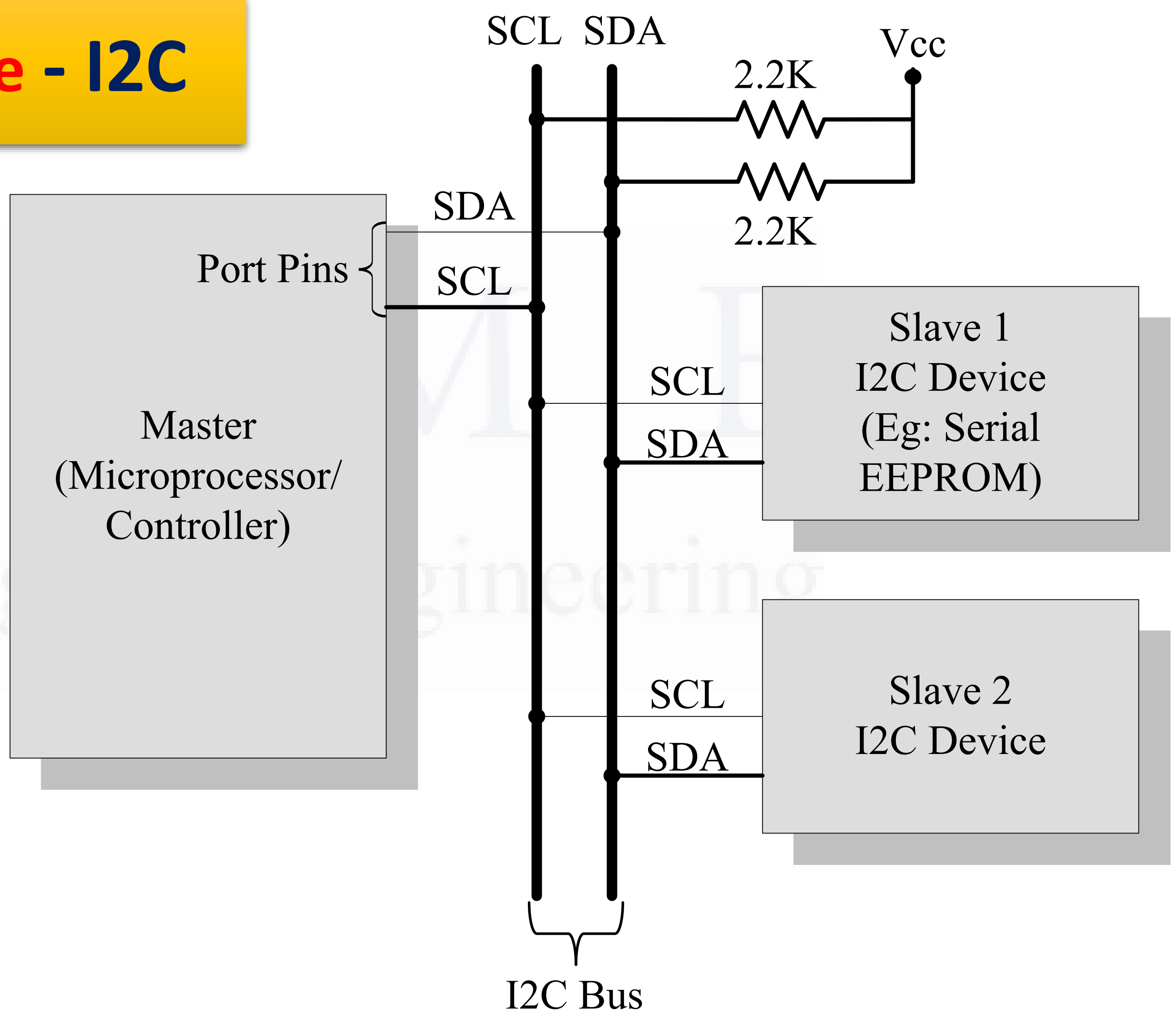
✓The original intention of I2C was to provide an easy way of connection between a microprocessor/microcontroller system and the peripheral chips in Television sets

# On-board Communication Interface - I2C

- Two bus lines:

  Serial Clock – SCL and Serial Data – SDA

- I2C is a Shared bus system.
- Devices connected to the I2C bus can act as either 'Master' device or 'Slave' device
- I2C supports multi masters on the same bus
- For proper operation both bus line should pulled to supply voltage



## On-board Communication Interface - I2C

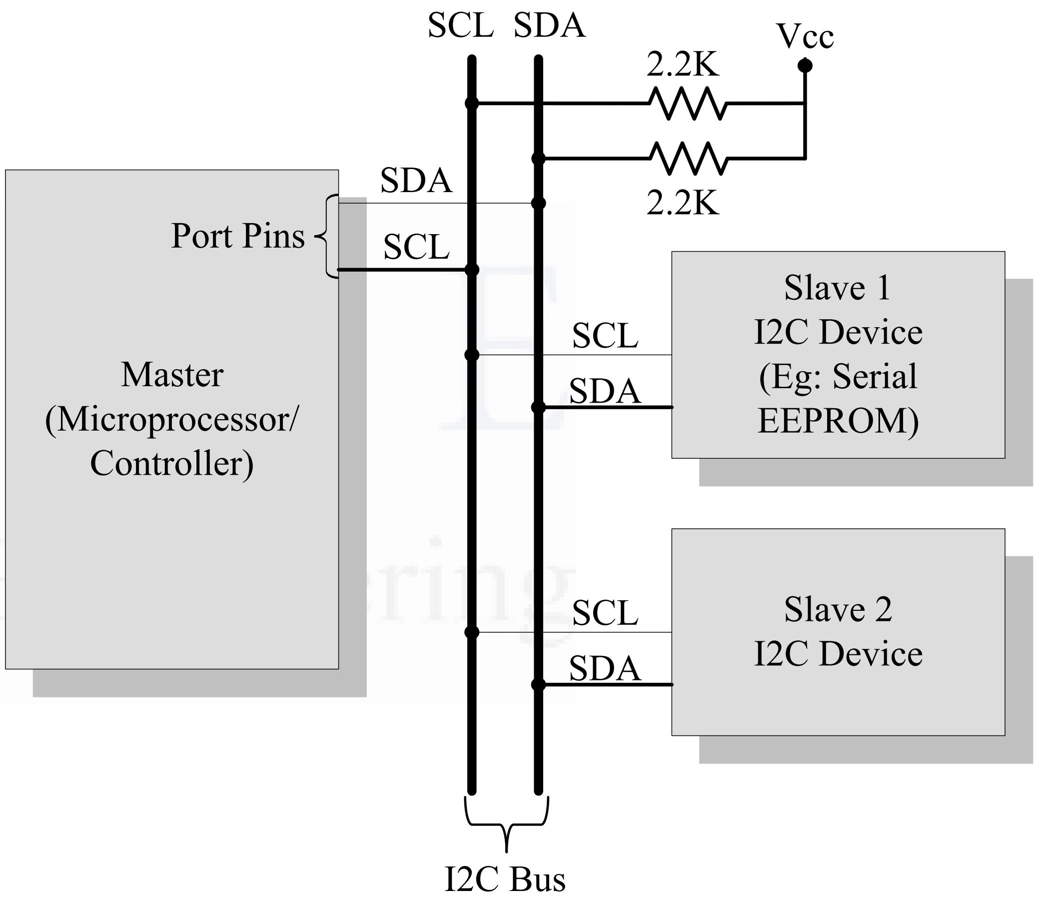
- ✓The ‘Master’ device is responsible for controlling the communication by initiating/terminating data transfer, sending data and generating necessary synchronization clock pulses
- ✓‘Slave’ devices wait for the commands from the master and respond upon receiving the commands
- ✓‘Master’ and ‘Slave’ devices can act as either transmitter or receiver
- ✓Regardless whether a master is acting as transmitter or receiver, the synchronization clock signal is generated by the ‘Master’ device only
- ✓I2C supports multi masters on the same bus

## On-board Communication Interface - I2C

**The sequence of operation for communicating with an I2C slave device is:**

1. Master device pulls the clock line (SCL) of the bus to 'HIGH'
2. Master device pulls the data line (SDA) 'LOW', when the SCL line is at logic 'HIGH' (This is the 'Start' condition for data transfer)
3. Master sends the address of the 'Slave' device to which it wants to communicate, over the SDA line.

- ✓ Clock pulses are generated at the SCL line for synchronizing the bit reception by the slave device.
- ✓ The MSB of the data is always transmitted first..
- ✓ The data in the bus is valid during the 'HIGH' period of the clock signal



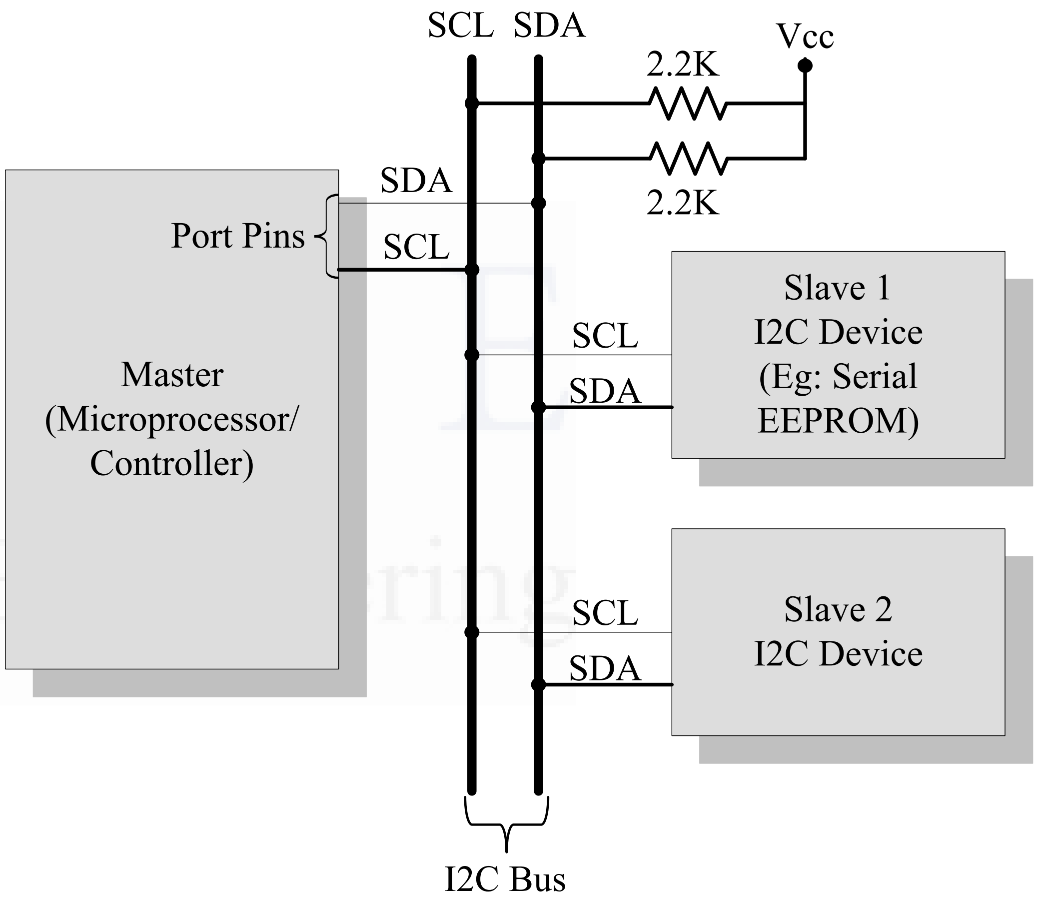
# On-board Communication Interface - I2C

**The sequence of operation for communicating with an I2C slave device is:**

4. Master sends the Read or Write bit according to the requirement.

(Bit value = 1 Read Operation; Bit value = 0 Write Operation)

5. Master waits for the acknowledgement bit from the slave device. Slave devices connected to the bus compares the address received with the address assigned to them.
6. The Slave device with the address requested by the master device responds by sending an acknowledge bit (Bit value =1) over the SDA line
7. Upon receiving the acknowledge bit, master sends the 8bit data to the slave device over SDA line.

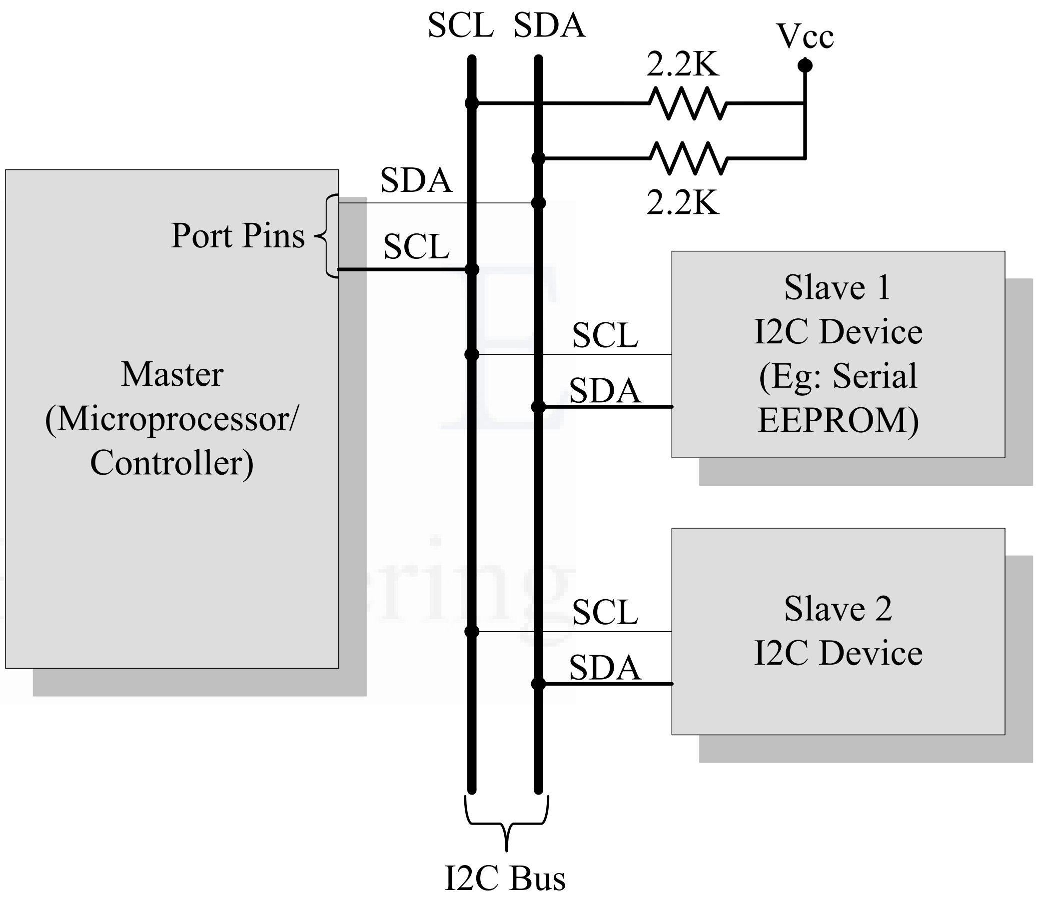


## On-board Communication Interface - I2C

**The sequence of operation for communicating with an I2C slave device is:**

8. Master *waits for the acknowledgement* bit from the device upon byte transfer complete for a *write operation* and *sends an acknowledge* bit to the slave device for a *read operation*.

9. Master terminates the transfer by pulling the SDA line 'HIGH' when the clock line SCL is at logic 'HIGH' (Indicating the 'STOP' condition)



## On-board Communication Interface – Serial Peripheral Interface (SPI) Bus

- The Serial Peripheral Interface Bus (SPI) is a ***synchronous, bi-directional, full duplex, four wire serial interface bus.***
- The concept of SPI is introduced by Motorola.
- SPI is a single master multi-slave system.
- It is possible to have a system where more than one SPI device can be master, provided the condition only one master device is active at any given point of time, is satisfied.

# On-board Communication Interface – Serial Peripheral Interface (SPI) Bus

**Master Out Slave In (MOSI):**

Signal line carrying the data from master to slave device. It is also known as Slave Input/Slave Data In (SI/SDI)

**Master In Slave Out (MISO):**

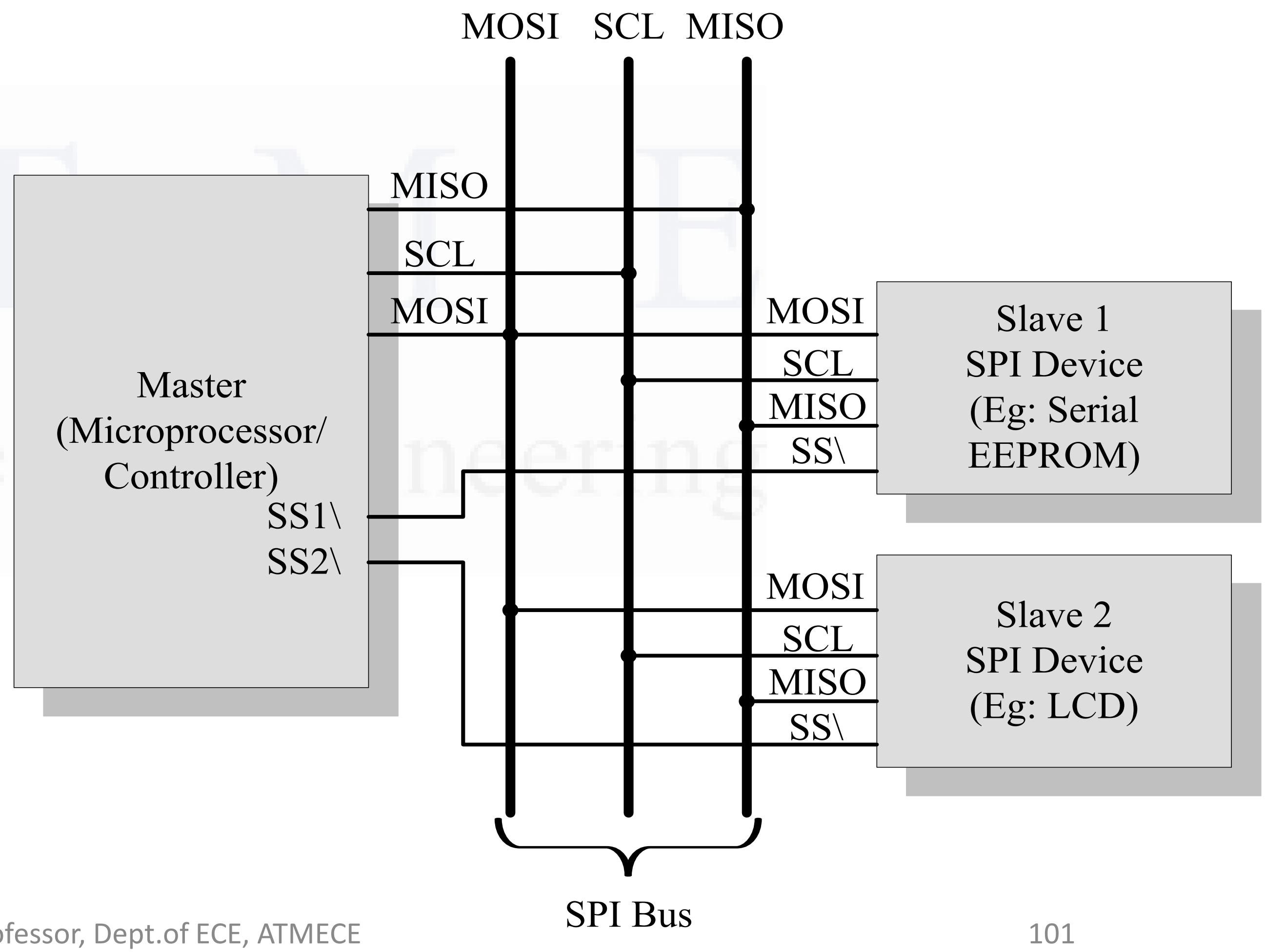
Signal line carrying the data from slave to master device. It is also known as Slave Output (SO/SDO)

**Serial Clock (SCLK):**

Signal line carrying the clock signals

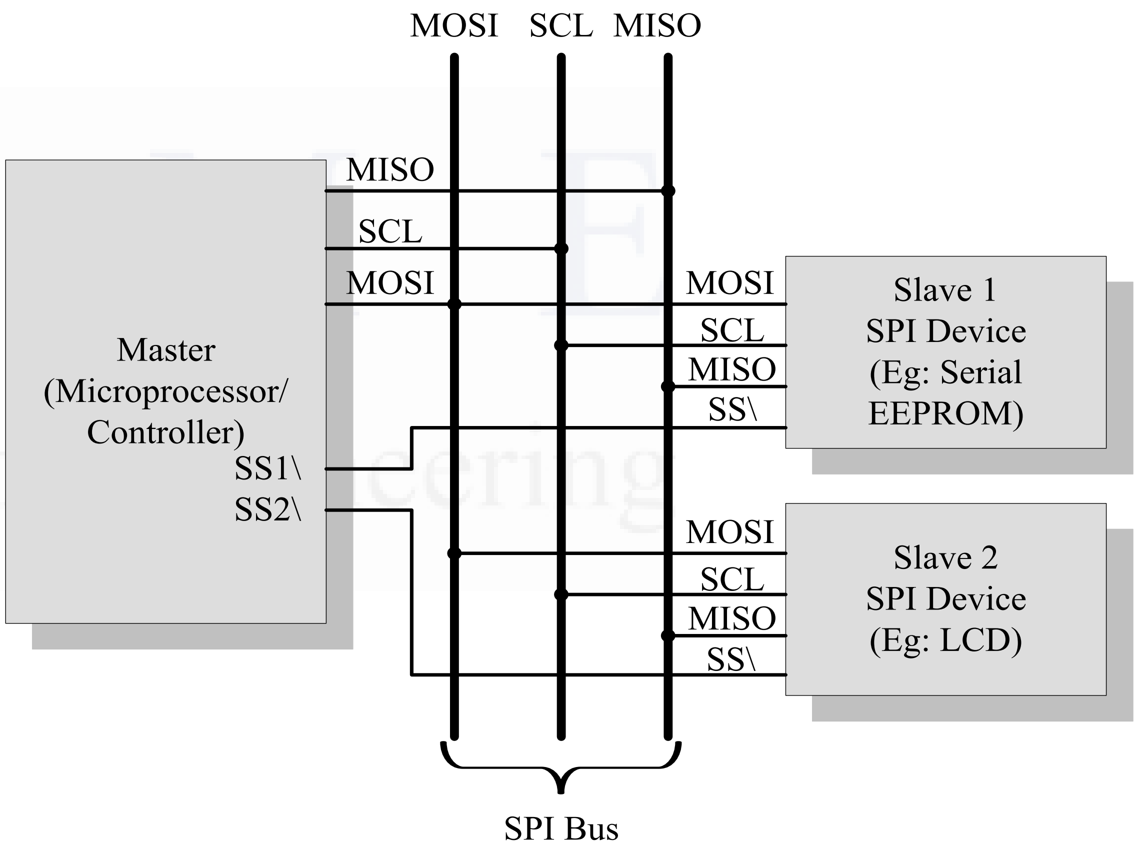
**Slave Select (SS):**

Signal line for slave device select. It is an active low signal



# On-board Communication Interface – Serial Peripheral Interface (SPI) Bus

- ✓The master device is responsible for generating the clock signal.
- ✓Master device selects the required slave device by asserting the corresponding slave device's slave select signal 'LOW'.
- ✓The data out line (MISO) of all the slave devices when not selected floats at high impedance state
- ✓SPI works on the principle of 'Shift Register'. The master and slave devices contain a special shift register for the data to transmit or receive.



# On-board Communication Interface – Serial Peripheral Interface (SPI) Bus

- ✓The serial data transmission through SPI Bus is fully configurable.
- ✓SPI devices contain certain set of registers for holding these configurations.
- ✓The Serial Peripheral Control Register holds the various configuration parameters like master/slave selection for the device, baudrate selection for communication, clock signal control etc.
- ✓The status register holds the status of various conditions for transmission and reception.

# Communication Interface-EXTERNAL

✓The ‘Product level communication interface’ (External Communication Interface) is responsible for data transfer between the embedded system and other devices or modules

✓The external communication interface can be either wired media or wireless media and it can be a serial or parallel interface.

✓ Infrared (IR), Bluetooth (BT), Wireless LAN (Wi-Fi), Radio Frequency waves (RF), GPRS etc are examples for wireless communication interface

✓RS-232C/RS-422/RS 485, USB, Ethernet (TCP-IP), IEEE 1394 port, Parallel port, CF-II Slot, SDIO, PCMCIA etc are examples for wired interfaces

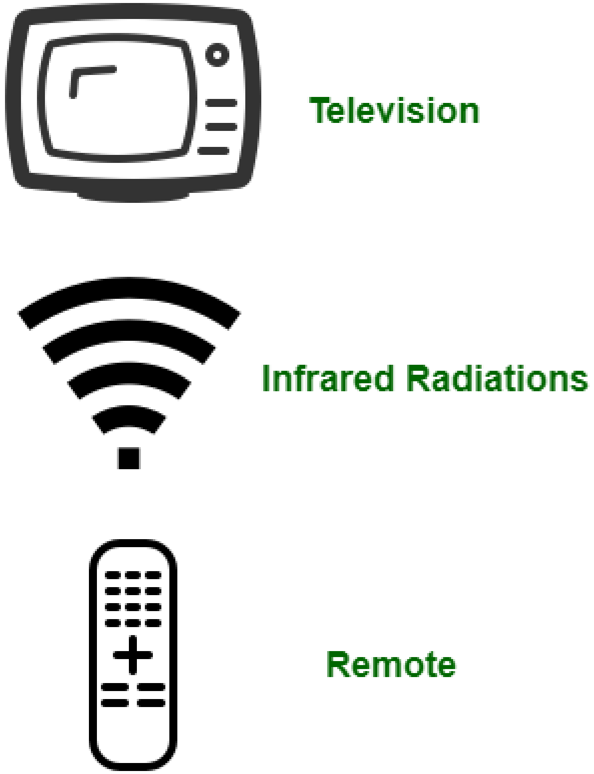
# External Communication Interface – Infrared (IrDA)

✓A serial, half duplex, line of sight based wireless technology for data communication between devices

✓Infrared communication technique makes use of Infrared waves of the electromagnetic spectrum for transmitting the data

✓IrDA supports point-point and point-to-multipoint communication, provided all devices involved in the communication are within the line of sight

✓The typical communication range for IrDA lies in the range 10cm to 1 m



# External Communication Interface – Infrared (IrDA)

✓IR supports data rates ranging from 9600bits/second to 16Mbps.

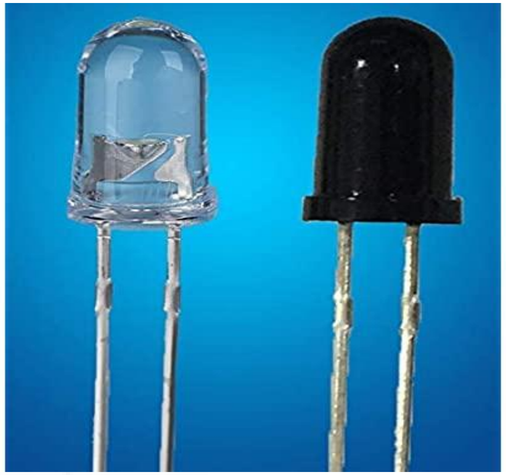
*Depending on the speed of data transmission IR is classified into*

1. Serial IR (SIR)
2. Medium IR (MIR)
3. Fast IR (FIR)
4. Very Fast IR (VFIR)
5. Ultra Fast IR (UFIR)

## External Communication Interface – Infrared (IrDA)

✓IrDA communication involves a transmitter unit for transmitting the data over IR and a receiver for receiving the data.

✓Infrared Light Emitting Diode (LED) is used as the IR source for transmitter and at the receiving end a photodiode is used as the receiver



# External Communication Interface – Bluetooth

- ✓Low cost, low power, short range wireless technology for data and voice communication
- ✓Bluetooth operates at 2.4GHz of the Radio Frequency spectrum and uses the Frequency Hopping Spread Spectrum (FHSS) technique for communication
- ✓Bluetooth supports a theoretical maximum data rate of up to 1Mbps and a range of approximately 30 feet for data communication

# External Communication Interface – Bluetooth

✓Bluetooth communication has two essential parts

1. Physical link part : responsible for the physical transmission of data between devices
2. Protocol part : responsible for defining the rules of communication

✓The physical link works on the Wireless principle making use of RF waves for communication

✓Bluetooth enabled devices essentially contain a Bluetooth wireless radio for the transmission and reception of data

# External Communication Interface – Bluetooth

- ✓The rules governing the Bluetooth communication is implemented in the 'Bluetooth protocol stack'
- ✓The Bluetooth communication IC holds the stack
- ✓Each Bluetooth device will have a 48 bit unique identification number
- ✓Bluetooth communication follows packet based data transfer

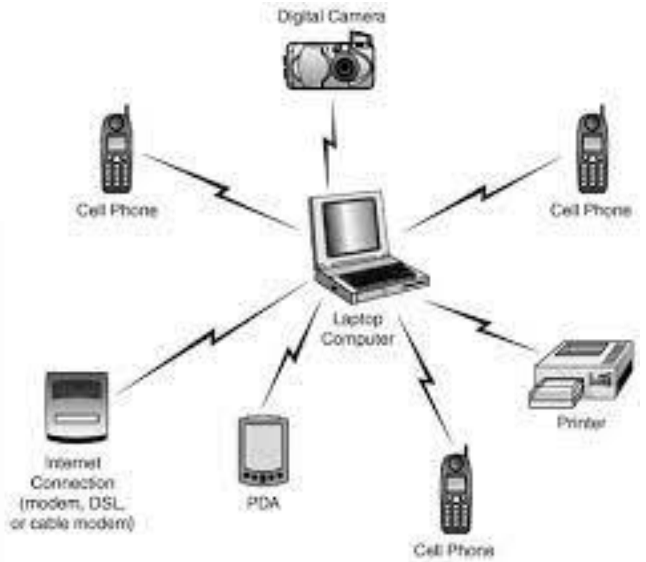
# External Communication Interface – Bluetooth

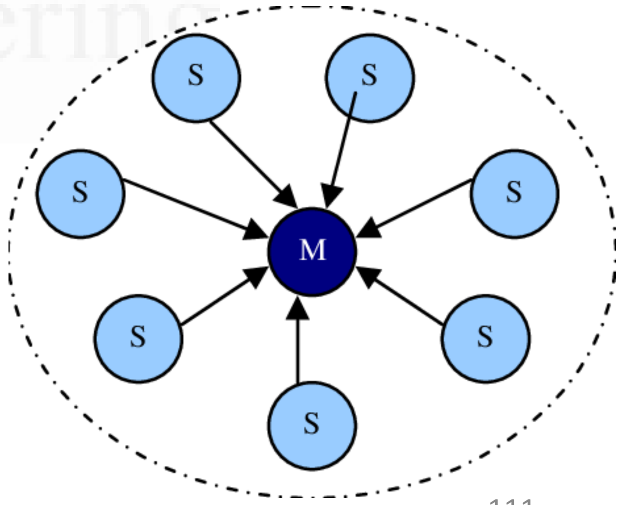
✓Bluetooth supports point-to-point (device to device) and point-to-multipoint (device to multiple device broadcasting) wireless communication.

✓The point-to-point communication follows the master-slave relationship.

✓ A Bluetooth device can function as either master or slave

✓A network formed with one Bluetooth device as master and more than one device as slaves is known as Piconet



# External Communication Interface – Wireless Fidelity (Wi-Fi)

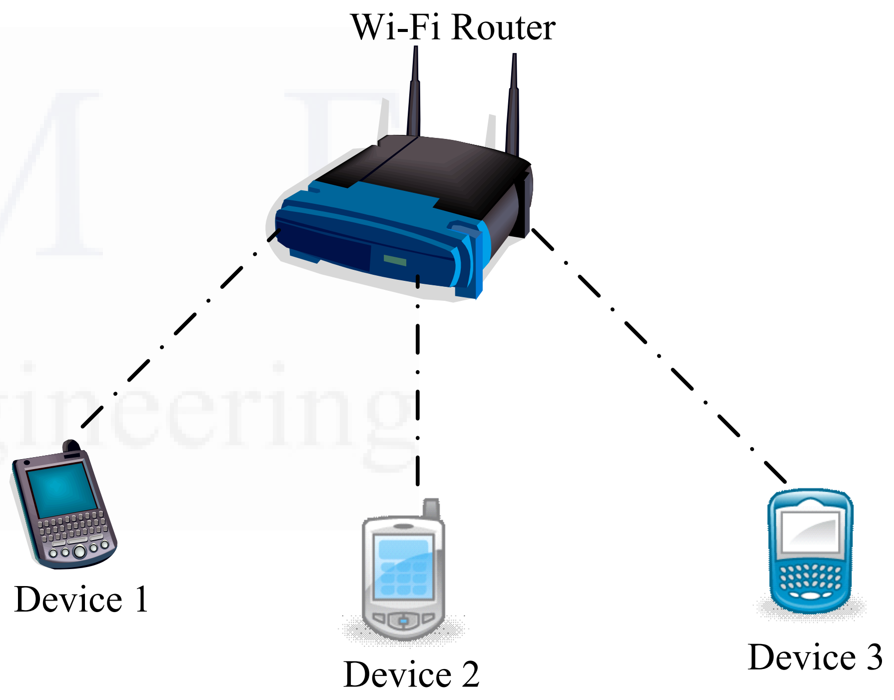
- ✓The popular wireless communication technique for networked communication of devices
- ✓Wi-Fi follows the IEEE 802.11 standard
- ✓Wi-Fi is intended for network communication and it supports Internet Protocol (IP) based communication
- ✓Wi-Fi based communications require an intermediate agent called Wi-Fi router/Wireless Access point to manage the communications

# External Communication Interface – Wi-Fi

✓The Wi-Fi router is responsible for restricting the access to a network, assigning IP address to devices on the network, routing data packets to the intended devices on the network

✓Wi-Fi enabled devices contain a wireless adaptor for transmitting and receiving data in the form of radio signals through an antenna

✓Wi-Fi operates at 2.4GHZ or 5GHZ of radio spectrum and they co-exist with other ISM band devices like Bluetooth



ATME College of Engineering

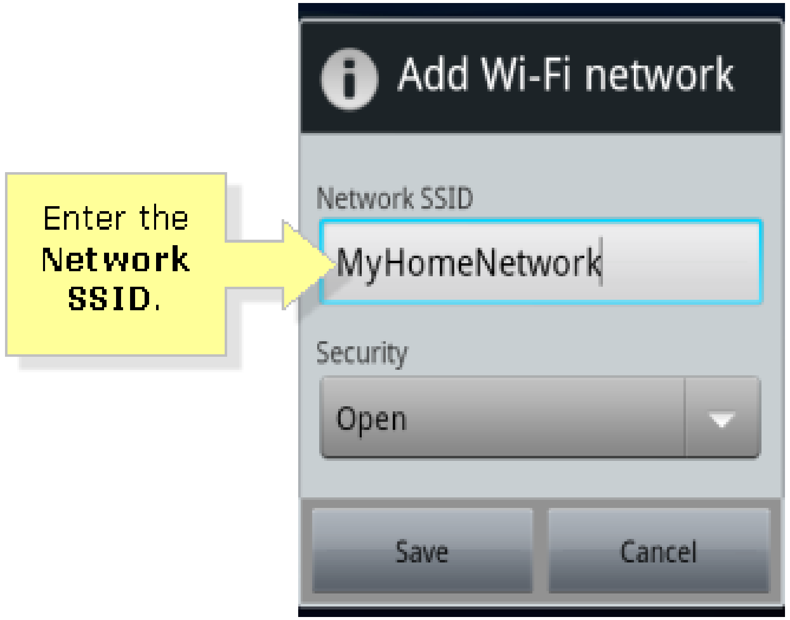
# External Communication Interface – Wi-Fi

✓A Wi-Fi network is identified with a Service Set Identifier (SSID).

✓A Wi-Fi device can connect to a network by selecting the SSID of the network and by providing the credentials if the network is security enabled

✓Wi-Fi networks implements different security mechanisms for authentication and data transfer

✓Wireless Equivalency Protocol (WEP), Wireless Protected Access (WPA) etc are some of the security mechanisms supported by Wi-Fi networks in data communication



## External Communication Interface – ZigBee

- ✓Low power, low cost, wireless network communication protocol based on the IEEE 802.15.4-2006 standard

- ✓ZigBee is targeted for low power, low data rate and secure applications for Wireless Personal Area Networking (WPAN)

- ✓The ZigBee specifications support a robust mesh network containing multiple nodes.

- ✓This networking strategy makes the network reliable by permitting messages to travel through a number of different paths to get from one node to another.

## External Communication Interface – ZigBee

- ✓ZigBee operates worldwide at the unlicensed bands of Radio spectrum, mainly at 2.400 to 2.484 GHz, 902 to 928 MHz and 868.0 to 868.6 MHz
- ✓ZigBee Supports an operating distance of up to 100 meters and a data rate of 20 to 250Kbps
- ✓ZigBee is primarily targeting application areas like Home & Industrial Automation, Energy Management, Home control/security, Medical/Patient tracking, Logistics & Asset tracking and sensor networks & active RFID
- ✓Automatic Meter Reading (AMR), smoke and detectors, wireless telemetry, heating control, Lighting controls, Environmental controls, etc are examples for applications which can make use of the ZigBee technology

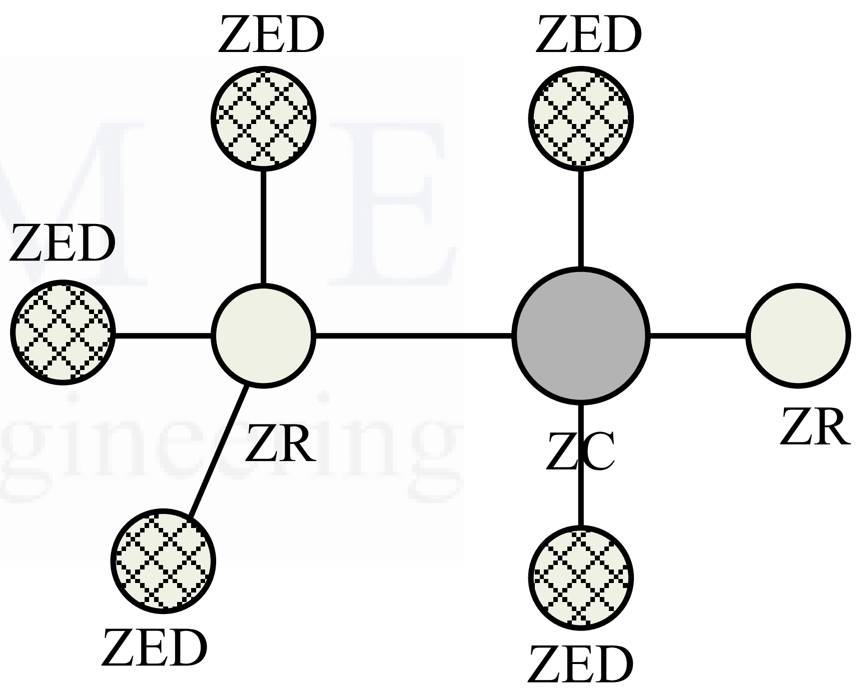
# External Communication Interface – ZigBee

In the ZigBee terminology, each ZigBee device falls under any one of the following ZigBee device category

**ZigBee Coordinator (ZC)/Network Coordinator:** The ZigBee coordinator acts as the root of the ZigBee network. The ZC is responsible for initiating the ZigBee network and it has the capability to store information about the network

**ZigBee Router (ZR)/Full function Device (FFD):** Responsible for passing information from device to another device or to another ZR

**ZigBee End Device (ZED)/Reduced Function Device (RFD):** End device containing ZigBee functionality for data communication. It can talk only with a ZR or ZC and doesn't have the capability to act as a mediator for transferring data from one device to another.



# SUMMARY

1. Embedded v/s General computing system
2. Classification of Embedded Systems
3. Major applications and purpose of ES
4. Elements of an Embedded System
5. Differences between RISC and CISC
6. Harvard and Von-Neumann
7. Big and Little Endian formats
8. Memory (ROM and RAM types)
9. Sensors, Actuators, Optocoupler
10. Communication Interfaces (I2C, SPI, IrDA, Bluetooth, Wi-Fi, Zigbee)

# JULY 2018

## Module-3

5 a. Differentiate between:
   (i) RISC and CISC architecture.
   (ii) Little Endian and Big Endian architecture. (08 Marks)

b. What are the features of the following:
   (i) I2C bus
   (ii) IrDA
   (iii) Optocoupler
   (iv) 1-wire interface (08 Marks)

OR

6 a. What are the different types of memories used in Embedded system design? Explain the role of each. (08 Marks)

b. Explain the following circuits in an embedded system :
   (i) Brown-out protection unit.
   (ii) Reset circuit. (08 Marks)

# JULY 2019

**Module-3**

| | | | |
|---|---|---|---|
| 5 | a. | Explain how embedded system are classified. | (08 marks) |
| | b. | With neat block diagram, explain the element of embedded system. | (08 Marks) |

**OR**

| | | | |
|---|---|---|---|
| 6 | a. | Differentiate between RISC and CISC. | (04 Marks) |
| | b. | Explain how program memory are classified. | (08 Marks) |
| | c. | Explain how brown-out protection circuits works. | (04 Marks) |

## JAN 2020

**Module-3**

5 a. Define embedded systems. Explain the 6 purpose of embedded systems with an example for each. (08 Marks)

b. Explain the classification of embedded systems based on generation. (04 Marks)

c. Mention the application of embedded system with an example for each. (04 Marks)

**OR**

6 a. Explain the different 'on board' communication interfaces in brief. (08 Marks)

b. Write a note on: (i) Reset circuit (ii) Watch dog timer. (08 Marks)

## SEP 2020

### Module-3

5 a. Define the term RAM. Mention different types of RAM and explain any one with neat circuit diagram. (06 Marks)

b. With a neat interfacing diagram explain the SPI bus. (06 Marks)

c. Bring out differences between FPGA and CPLD. (04 Marks)

OR

6 a. Mention all the cores around which an embedded system is built. Discuss any two in detail. (08 Marks)

b. Write a note on embedded firmware. (04 Marks)

c. Explain the importance of brown out protection circuit with a neat diagram. (04 Marks)