Module-1

Introduction to Embedded System

Topics:

1. Embedded Vs General computing system,
2. Classification of Embedded systems
3. Major applications and purpose of ES.
4. Elements of an Embedded System (Block diagram and explanation),
5. Differences between RISC and CISC, Harvard and Princeton, Big and Little Endian formats,
6. Memory (ROM and RAM types), Sensors, Actuators, Optocoupler,
7. Communication Interfaces (I2C, SPI, IrDA, Bluetooth, Wi-Fi, Zigbee only)

1.1 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)

Examples:

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

Embedded Systems are:

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

1.2 Embedded Systems Vs General Computing Systems

General Purpose System	Embedded System
A system which is a combination of generic hardware and 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
Contain a General Purpose Operating System (GPOS)	May or may not contain an operating system for functioning
Applications are alterable (programmable) by user (It is possible for the end user to re-install the Operating System, and add or remove user applications)	The firmware of the embedded system is pre-programmed and it is non-alterable by end-user (There may be exceptions for systems supporting OS kernel image flashing through special hardware settings)
Performance is the key deciding factor on the selection of the system. Always ‘Faster is Better’	Application specific requirements (like performance, power requirements, memory usage etc) are the key deciding factors
Less/not at all tailored towards reduced operating power requirements, options for different levels of power management.	Highly tailored to take advantage of the power saving modes supported by hardware and Operating System
Response requirements are not time critical	For certain category of embedded systems like mission critical systems, the response time requirement is highly critical
Need not be deterministic in execution behavior	Execution behavior is deterministic for certain type of embedded systems like ‘Hard Real Time’ systems

1.3 History of Embedded Systems:

First Recognized Modern Embedded System: Apollo Guidance Computer (AGC)

First Mass Produced Embedded System: Autonetics D-17 Guidance computer

1.4 Classification of Embedded Systems:

The classification of embedded system is based on following criteria's:

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

1.4.1. Embedded Systems Classification based on Generation

1. First Generation:

- The early embedded systems built around 8bit microprocessors like 8085 and Z80 and 4bit microcontrollers.
- Simple in hardware circuit & firmware developed.
- Examples: Digital telephone keypads.

2. Second Generation:

- Embedded Systems built around 16bit microprocessors and 8 or 16bit microcontrollers, following the first generation embedded systems.
- They are more complex & powerful than first generation microprocessors and controller.
- Example: SCADA systems

3. 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)
- Examples: Robotics, Media, etc.

4. Fourth Generation:

- Embedded Systems built around System on Chips (SoCs), Re-configurable processors and multicore processors
- Highly complex and very powerful.
- Examples: Smart Phones

1.4.2. Embedded Systems Classification based on Complexity & Performance Requirements

1. Small Scale:

- Simple in application need.
- Performance not time-critical.
- Built around low performance & low cost 8 or 16 bit µp/µc.
- May or may not contain operating system for functioning.
- Example: an electronic toy.

2. Medium Scale:

- Slightly complex in hardware & firmware requirement.
- Built around medium performance & low cost 16 or 32 bit µp/µc.
- Usually contain embedded operating system (either general purpose of RTOS) for functioning.
- Examples: Industrial machines.

3. Large Scale/Complex:

- Highly complex hardware & firmware.
- Built around 32 or 64 bit RISC µp/µc or PLDs or Multicore-Processors.
- Contain a High performance RTOS for task scheduling, prioritization and management.
- Response is time-critical.
- Examples: Mission critical applications which demands high performance
  Decoding/encoding of media
  Cryptographic function implementation

1.4.3. Embedded Systems Classification based on deterministic behavior

- This classification is applicable for "Real Time" systems.
- The task execution behavior for an embedded system may be deterministic or non-deterministic.
- Based on execution behavior Real Time embedded systems are divided into Hard and Soft.

1.4.4. Embedded Systems Classification based on triggering

- Embedded systems which are "Reactive" in nature can be based on triggering.
- Reactive systems can be: Event triggered or Time triggered.

1.5 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
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.

1.6 Purpose of Embedded Systems

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

1. Data Collection/Storage/Representation
2. Data Communication
3. Data (Signal) Processing
4. Monitoring
5. Control
6. Application Specific User Interface

1.6.1 Data Collection/Storage/Representation

- Embedded system designed for the purpose of data collection performs acquisition of data from the external world.
- Data collection is usually done for storage, analysis, manipulation and transmission.
- Data can be analog or digital.
- Embedded systems with analog data capturing techniques collect data directly in the form of analog signal whereas an embedded system with digital data collection mechanism converts the analog signal to the digital signal using analog to digital converters.
- If the data is digital it can be directly captured by digital embedded system.

- A digital camera is a typical example of an embedded System with data collection/storage/representation of data.
- Images are captured and the captured image may be stored within the memory of the camera.
- The captured image can also be presented to the user through a graphic LCD unit.

1.6.2 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/IEEE1394 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.

1.6.3 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).

1.6.4 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 heartbeat 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.

1.6.5 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.

- 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.

1.6.6 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.

1.7 Elements of an Embedded System

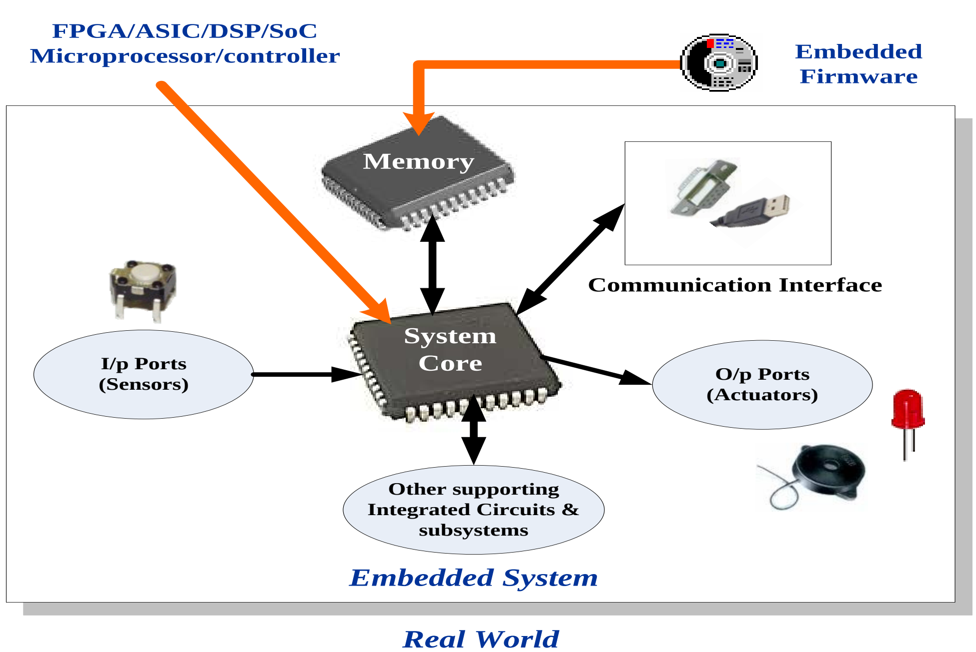


Figure 3.1: Elements of an Embedded System

A typical embedded system contains:

- **System Core:**
  - It is a Single chip controller, which acts as master brain of the system.
  - It can be Microprocessor (eg. Intel 8085, 8086) or Microcontroller (eg .Atmel AT89C51) or Field Programmable Gate Array [FPGA] (eg. Xilinx, Spartan) or Digital Signal Processor (DSP) or Application Specific Integrated Circuits (ASIC)/ Application Specific Standard Product (ASSP)

- **Embedded Firmware**:

- Designed to regulate a state of device through control signals.
- It acts as intelligence to the system.
- In a controller based embedded system, the controller may contain internal memory for storing the controller algorithm (eg. EEPROM or FLASH memory).
- Embedded hardware or software systems are basically designed to regulate a physical variable or to manipulate the state of some devices by sending some control signal to the actuator or devices connected to the output ports of the system, in response to the input signals provided by the end users or Sensor which are connected to the input ports.
- Hence an embedded system can be viewed as a reactive system.
- The control is achieved by processing the information coming from the sensors and user interfaces and controlling some actual tests that regulate the physical variable.

- **I/O devices:**

- Keyboard push buttons switches extra example for common user interface input devices.
- LEDs, liquid crystal display, piezoelectric buzzer ..etc are example for output device
- It is not necessary that all embedded system should incorporate these user interfaces it's only depends on the type of application for which the embedded system is designed.
- For example if the embedded system is designed for any handheld application such as mobile handset application then the system should contain user interfaces like keyboard performing input operation and display unit for providing you the status of various activities in progress.

- **Memory:**

- The memory of the system is responsible for holding the control algorithm and other important configuration details.
- For most of embedded systems the memory for storing the algorithm or configuration data is a fixed type which is a kind of read only memory ROM and it is not available for the end user for modification which means the memory is protected from unwanted user interaction by implementing some kind of memory protection mechanism.
- The most common types of memories used in embedded system for control algorithms storage are OTP, PROM, UVEPROM, EEPROM and FLASH
- Sometimes the system required for temporary memory for performing arithmetic operation or control algorithm execution and this type of memory is known as working memory.
- Random Access Memory (RAM) is used in most of the system as the working memory
- Various types of RAM like S-RAM, D-RAM and NVRAM are used for this purpose.

- **Communication Interface:**

- Communication interface is essential for communicating with various subsystems of the embedded system and with the external world.
- Device/board level communication interface (Onboard Communication Interface)
  Eg. I2C, SPI, UART, 1-Wire etc and Parallel bus

- Product level communication interface (External Communication Interface
  It can be either ***wired media***: Eg. S-232C/RS-422/RS 485, USB, Ethernet (TCP-IP), IEEE 1394 port, Parallel port, CF-II Slot, SDIO, PCMCIA.
  It can be ***wireless media***: Eg. Infrared (IR), Bluetooth (BT), Wireless LAN (Wi-Fi), Radio Frequency waves (RF), GPRS.

1.8 The Core of the Embedded Systems

The core of the embedded system falls into any one of the following categories:

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

1.8.1 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).
- 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.

1.8.2. 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 dependant 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.

1.8.3. 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).
- 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.

1.8.4 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.
- Digital Signal Processors are 2 to 3 times faster than the general purpose microprocessors in signal processing applications.
- DSPs 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.
- 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.

1.8.5. 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

1.8.6 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
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

1.8.7. Harvard V/s Von-Neumann Processor/Controller Architecture

- ✓ The terms Harvard and Von-Neumann refers to the processor architecture design.
- ✓ Microprocessors/controllers based on the **Von-Neumann** architecture shares a single common bus for fetching both instructions and data. Program instructions and data are stored in a common main memory
- ✓ Microprocessors/controllers based on the **Harvard** architecture will have separate data bus and instruction bus. This allows the data transfer and program fetching to occur simultaneously on both buses
- ✓ With Harvard architecture, the data memory can be read and written while the program memory is being accessed. These separated data memory and code memory buses allow one instruction to execute while the next instruction is fetched ("Pre-fetching")

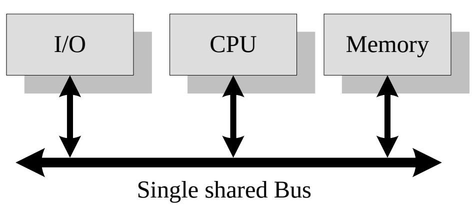
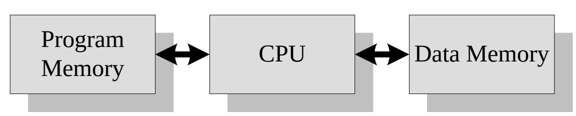


Fig: 3.2 Von-Neumann Architecture



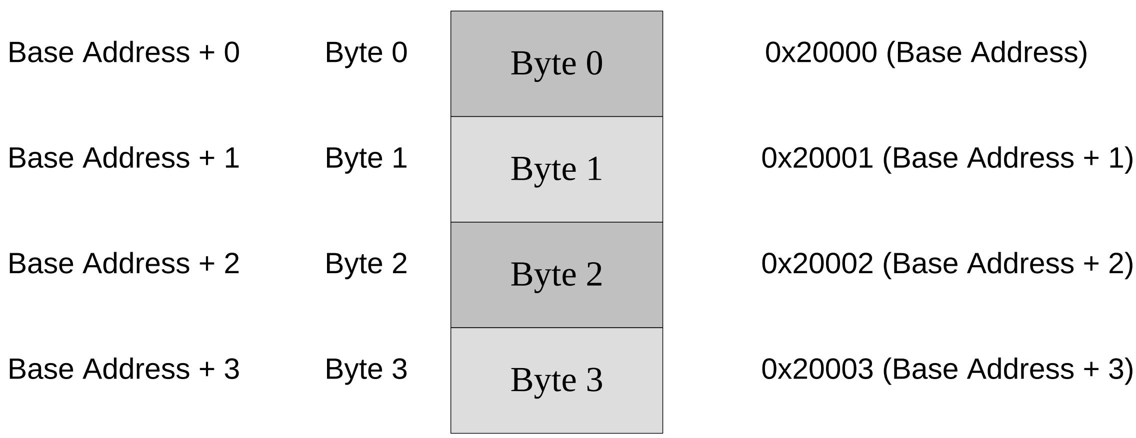
Harvard Architecture

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
No memory alignment problems	Allows self-modifying codes†
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

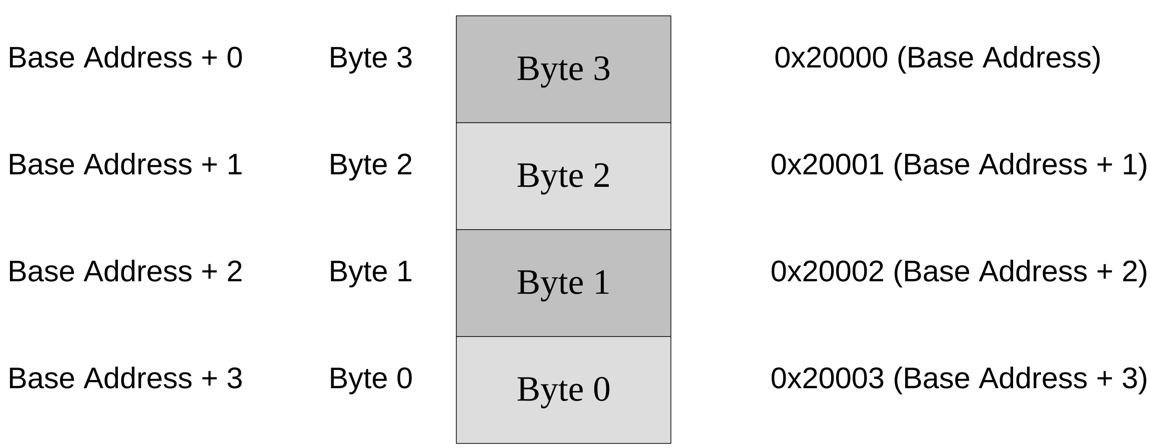
1.8.8 Big-endian V/s Little-endian processors

Endianness specifies the order in which the data is stored in the memory by processor operations in a multi byte system (Processors whose word size is greater than one byte). Suppose the word length is two byte then data can be stored in memory in two different ways.

- Higher order of data byte at the higher memory and lower order of data byte at location just below the higher memory.
- Lower order of data byte at the higher memory and higher order of data byte at location just below the higher memory.
- ***Little-endian*** means the lower-order byte of the data is stored in memory at the lowest address, and the higher-order byte at the highest address. (The little end comes first)
- ***Big-endian*** means the higher-order byte of the data is stored in memory at the lowest address, and the lower-order byte at the highest address. (The big end comes first.)



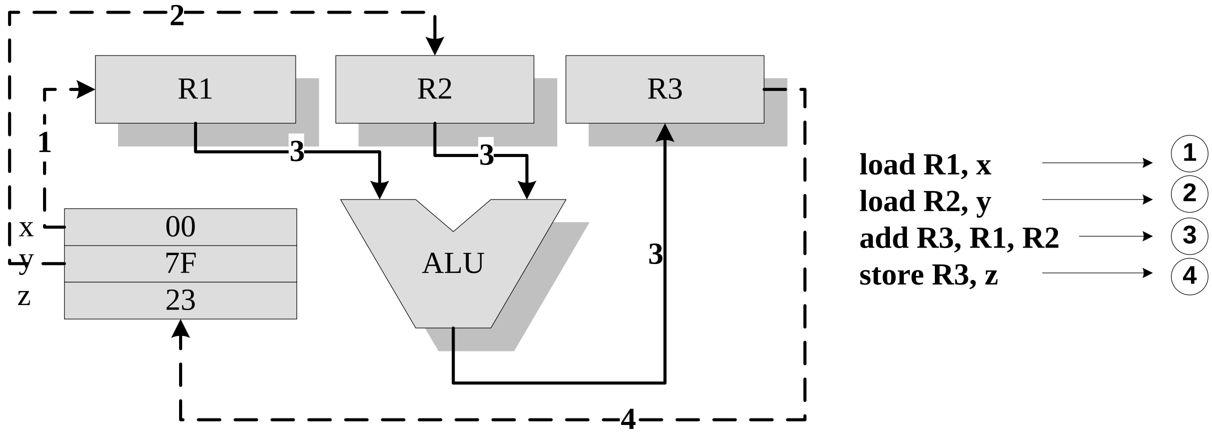
Little-endian Operation



Big-endian Operation

1.8.9 Load Store Operation & Instruction Pipelining

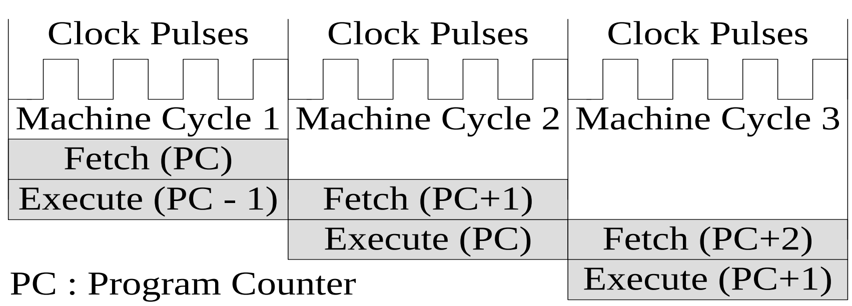
The RISC processor instruction set is orthogonal and it operates on registers. The memory access related operations are performed by the special instructions *load* and *store*. If the operand is specified as memory location, the content of it is loaded to a register using the *load* instruction. The instruction *store* stores data from a specified register to a specified memory location.



Load Store Operation

Instruction Pipelining

- The conventional instruction execution by the processor follows the fetch-decode-execute sequence.
- The ‘fetch’ part fetches the instruction from program memory or code memory and the decode part decodes the instruction to generate the necessary control signals.
- The execute stage reads the operands, perform ALU operations and stores the result. In conventional program execution, the fetch and decode operations are performed in sequence.
- During the decode operation the memory address bus is available and if it possible to effectively utilize it for an instruction fetch, the processing speed can be increased.
- In its simplest form instruction pipelining refers to the overlapped execution of instructions.



The Single stage pipelining concept

1.9 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 - Fixed and Programmable. The circuits in a fixed logic device are permanent, they perform one function or set of functions - once manufactured, they cannot be changed.
- 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.
- Designers can use inexpensive software tools to quickly develop, simulate, and test their logic designs in PLD based design. The design can be quickly programmed into a device, and immediately tested in a live circuit.
- PLDs are based on re-writable memory technology and the device is reprogrammed to change the design.

1.9.1 CPLDs and FPGA

- ✓ Field Programmable Gate Arrays (FPGAs) and Complex Programmable Logic Devices (CPLDs) are the two major types of programmable logic devices.
- ✓ 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.
- ✓ 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.

1.10 Application Specific Integrated Circuit (ASIC)

- A microchip designed to perform a specific or unique application. It is used as replacement to conventional general purpose logic chips.
- ASIC integrates several functions into a single chip and thereby reduces the system development cost.
- Most of the ASICs are proprietary products. As a single chip, ASIC consumes very small area in the total system and thereby helps in the design of smaller systems with high capabilities/functionalities.
- ASICs can be pre-fabricated for a special application or it can be custom fabricated by using the components from a re-usable '*building block*' library of components for a particular customer application.
- Fabrication of ASICs requires a non-refundable initial investment (Non Recurring Engineering (NRE) charges) for the process technology and configuration expenses.
- If the Non-Recurring Engineering Charges (NRE) is born by a third party and the Application Specific Integrated Circuit (ASIC) is made openly available in the market, the ASIC is referred as Application Specific Standard Product (ASSP).

1.11 Commercial off the Shelf Component (COTS)

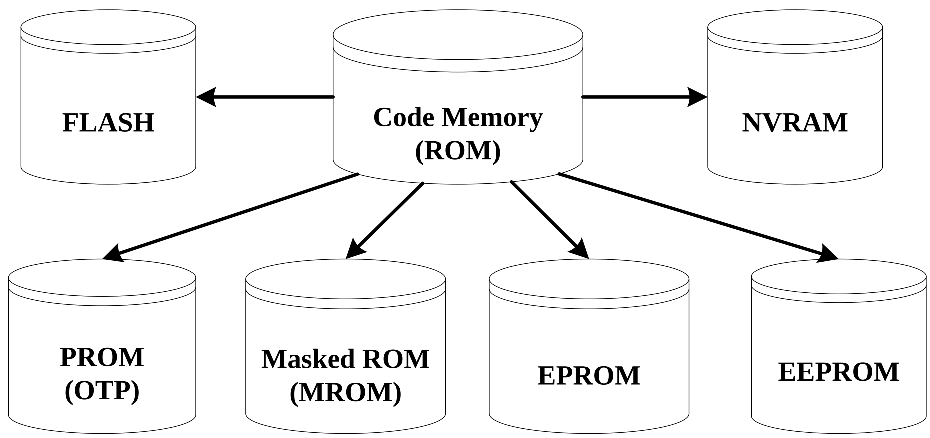
- A Commercial off-the-shelf (COTS) product is one which is used '*as-is*'.
- COTS products are designed in such a way to provide easy integration and interoperability with existing system components.
- Typical examples for the COTS hardware unit are Remote Controlled Toy Car control unit including the 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.
- A COTS component in turn contains a General Purpose Processor (GPP) or Application Specific Instruction Set Processor (ASIP) or Application Specific Integrated Chip (ASIC)/Application Specific Standard Product (ASSP) or 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 extent.

1.12 Memory

- Memory is an important part of an embedded system. The memory used in embedded system can be either Program Storage Memory (ROM) or Data memory (RAM).
- Certain Embedded processors/controllers contain built in program memory and data memory and this memory is known as on-chip memory.

1.12.1 Program Storage Memory

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



1. 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.
- Different mechanisms are used for the masking process of the ROM, like
  - Creation of an enhancement or depletion mode transistor through channel implant.
  - By creating the memory cell either using a standard transistor or a high threshold transistor. In the high threshold mode, the supply voltage required to turn ON the transistor is above the normal ROM IC operating voltage. This ensures that the transistor is always off and the memory cell stores always logic 0.
- The limitation with MROM based firmware storage is the inability to modify the device firmware against firmware upgrades. Since the MROM is permanent in bit storage, it is not possible to alter the bit information.

2. 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.
- Fuses which are not blown/burned represents a logic “1” whereas 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.

3. 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.

4. 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).

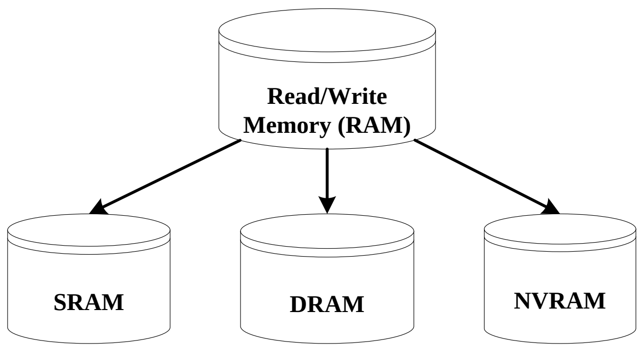
5. FLASH Memory

- 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.

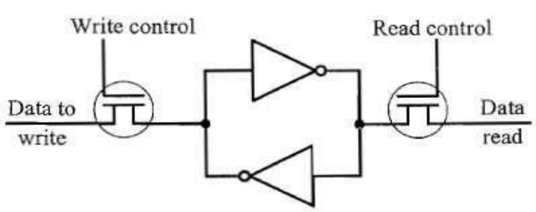
1.12.2 Read-Write Memory/Random Access Memory (RAM)

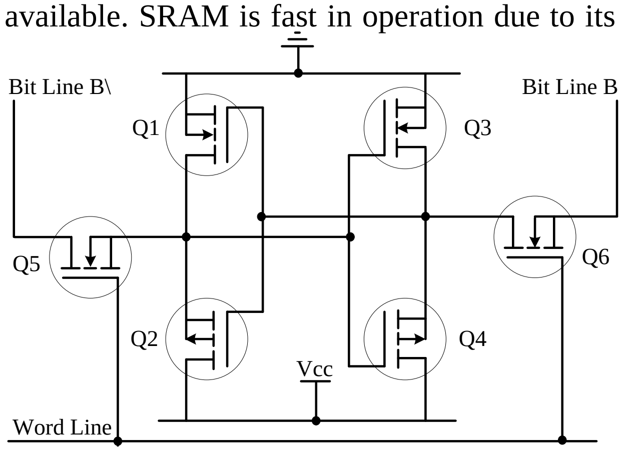
- RAM is the data memory or working memory of the controller/processor.
- RAM is volatile, meaning when the power is turned off, all the contents are destroyed.
- RAM is a direct access memory, meaning we can access the desired memory location directly without the need for traversing through the entire memory locations to reach the desired memory position (i.e. Random Access of memory location).



1. **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.
- Static RAM is the fastest form of RAM available. SRAM is fast in operation due to its resistive networking and switching capabilities.



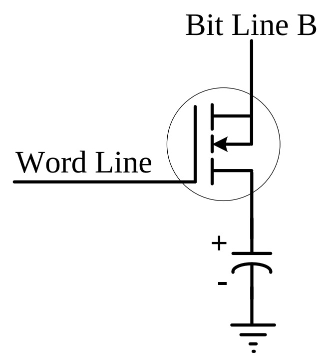


SRAM cell implementation

- Accesses to the memory cell is controlled by the word line, which controls the access transistors (MOSFET Q5 and Q6) .
- The access transistors control the connections to bit line B &B\.
- In order to write a value to the memory cell, apply the desired value to the bit control lines (For writing 1, make B=1 and B\=0; For writing 0 , , make B=o and B\=1) and assert word line.
- For reading the content of the memory cell, assert both B and B\ bit lines to 1 and set word line to 1.
- The major limitation of SRAM is low capacity and high cost.

2. 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.
- Special circuits called DRAM controllers are used for the refreshing operation. The refresh operation is done periodically in milliseconds interval.



DRAM cell implementation

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.

3. 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 nonvolatile 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.

1.13 Sensors & Actuators

Sensor:

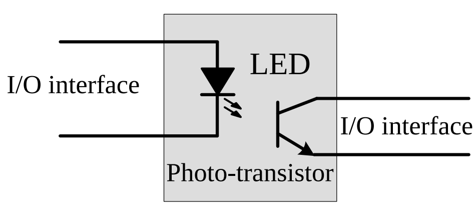
- A transducer device which converts energy from one form to another for any measurement or control purpose. Sensors acts as input device
- Eg. Hall Effect Sensor which measures the distance between the cushion and magnet in the Smart Running shoes from adidas.

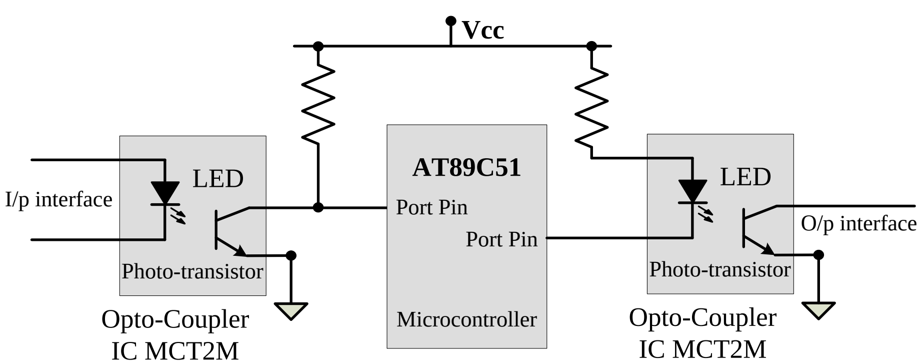
Actuator:

- A form of transducer device (mechanical or electrical) which converts signals to corresponding physical action (motion). Actuator acts as an output device
- Eg. Micro motor actuator which adjusts the position of the cushioning element in the Smart Running shoes from adidas.

1.14 I/O Devices – Opto-coupler

- ✓ 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).
- ✓ 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.





Opto-coupler in input and output circuit

1.15 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)

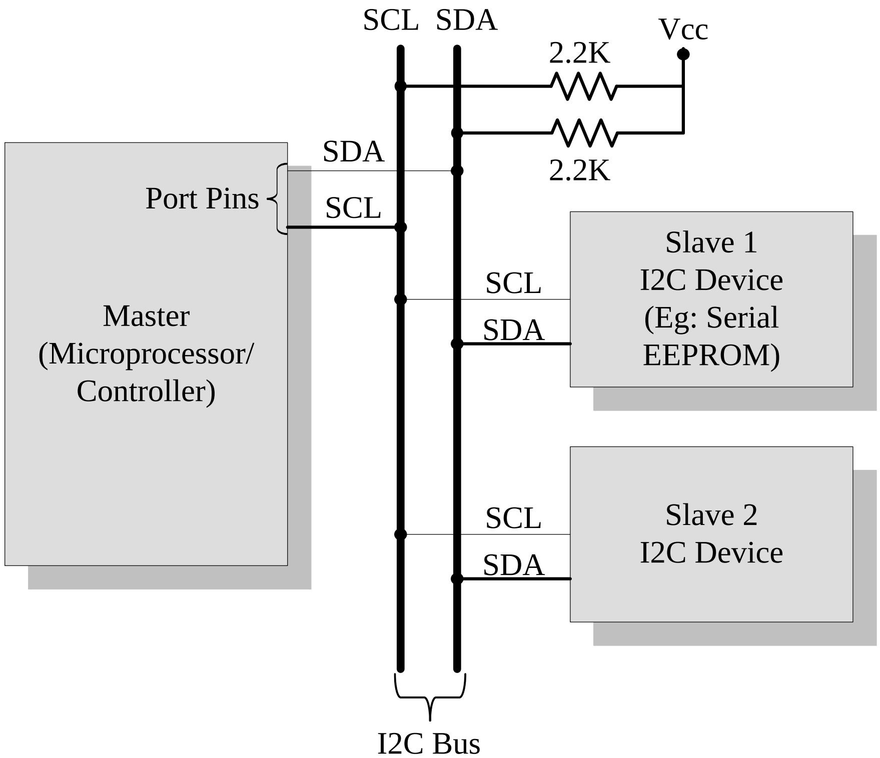
- Embedded product is a combination of different types of components (chips/devices) arranged on a Printed Circuit Board (PCB).
- The communication channel which interconnects the various components within an embedded product is referred as Device/board level communication interface (Onboard Communication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

2. Product level communication interface (External Communication Interface)

- 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.

1.15.1 Onboard Communication Interface -Inter Integrated Circuit Bus (I2C)

- Inter Integrated Circuit Bus (I2C - Pronounced 'I square C') is a synchronous bi-directional half duplex (one-directional communication at a given point of time) two wire serial interface bus
- The concept of I2C bus was developed by 'Philips Semiconductors' in the early 1980's. 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.
- The I2C bus is comprised of two bus lines, namely; Serial Clock – SCL and Serial Data – SDA. SCL line is responsible for generating synchronization clock pulses and SDA is responsible for transmitting the serial data across devices.
- I2C bus is a shared bus system to which many number of I2C devices can be connected. Devices connected to the I2C bus can act as either 'Master' device or 'Slave' device.
- 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.



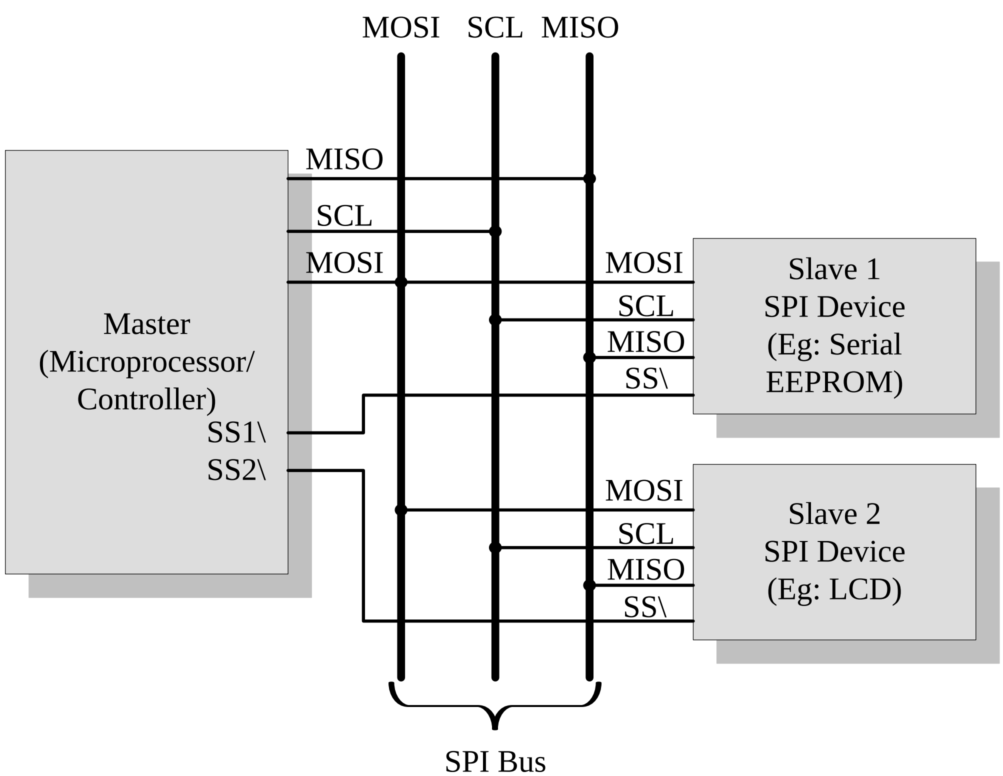
The sequence of operation for communicating with an I2C slave device is:

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 (7 bit or 10 bit wide) 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.
4. Master sends the Read or Write bit (Bit value = 1 Read Operation; Bit value = 0 Write Operation) according to the requirement.
5. Master waits for the acknowledgement bit from the slave device whose address is sent on the bus along with the Read/Write operation command. 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, if the requested operation is 'Write to device'. If the requested operation is 'Read from device', the slave device sends data to the master over the SDA line.
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).

1.15.2 Onboard 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.
- SPI requires four signal lines for communication. They are:

1. **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).
2. **Master In Slave Out (MISO):** Signal line carrying the data from slave to master device. It is also known as Slave Output (SO/SDO).
3. **Serial Clock (SCLK):** Signal line carrying the clock signals.
4. **Slave Select (SS):** Signal line for slave device select. It is an active low signal.



- 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

- 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, baud rate selection for communication, clock signal control etc. The status register holds the status of various conditions for transmission and reception.

- 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. The size of the shift register is device dependent. Normally it is a multiple of 8. During transmission from the master to slave, the data in the master's shift register is shifted out to the MOSI pin and it enters the shift register of the slave device through the MOSI pin of the slave device. At the same time the shifted out data bit from the slave device's shift register enters the shift register of the master device through MISO pin.

1.15.3 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.
- IR supports data rates ranging from 9600bits/second to 16Mbps. Depending on the speed of data transmission IR is classified into Serial IR (SIR), Medium IR (MIR), Fast IR (FIR), Very Fast IR (VFIR) and Ultra Fast IR (UFIR).
- SIR supports transmission rates ranging from 9600bps to 115.2kbps. MIR supports data rates of 0.576Mbps and 1.152Mbps. FIR supports data rates up to 4Mbps. VFIR is designed to support high data rates up to 16Mbps. The UFIR specs are under development and it is targeting a data rate up to 100Mbps.
- 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.

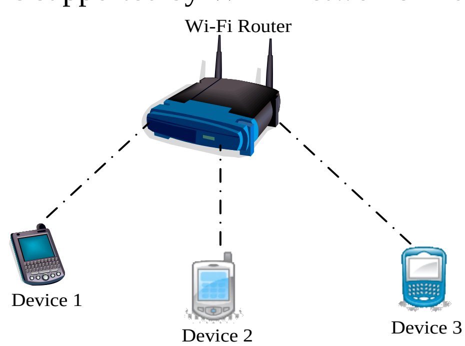
1.15.4 External Communication Interface – Bluetooth

- 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.
- Bluetooth communication has two essential parts; a physical link part and a protocol part. The physical link is responsible for the physical transmission of data between devices supporting Bluetooth communication and protocol part is 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.
- 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.
- 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.

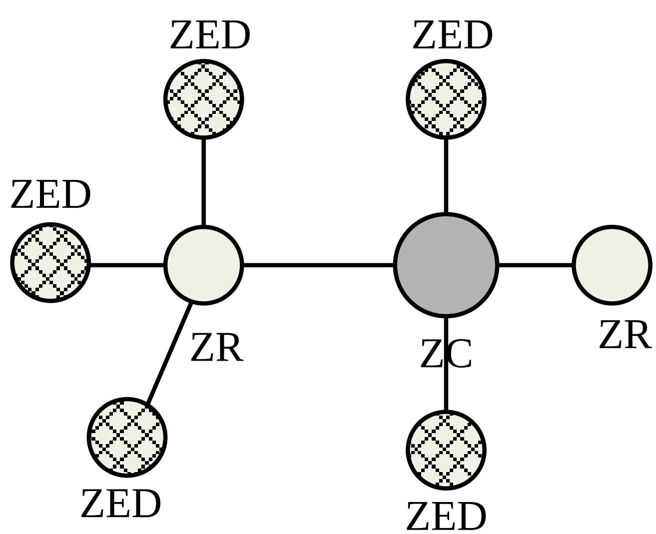
1.15.5.External Communication Interface – Wi-Fi

- 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.
- 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.
- 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.



1.15.6.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.
- 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, HVAC control, heating control, Lighting controls, Environmental controls, etc are examples for applications which can make use of the ZigBee technology.
- 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.



Recommended Questions

1. What is an embedded system? Mention its applications.

2. Explain the purposes of embedded systems with an example for each.

3. Explain how embedded systems are classified.

4. Mention the application of embedded system with an example for each.

5. With a neat diagram explain the elements of Embedded System.

6. Mention all the cores around which an embedded system is built. Explain any two in detail.

7. Differentiate between

    i) General Computing Systems and Embedded Systems
    ii) Microprocessor and microcontroller
    iii) RISC and CISC
    iv) Harvard and Von-Numen architecture
    v) Big endian and Little endian architecture.
    vi) SRAM and DRAM

8. Compare PLD, ASIC and COTS.

9. What are the different types of memories used in embedded system design? Explain the role of each

10. Explain SRAM design and features with a diagram.

11. Write a short note on Optocoupler

12. Explain the sequence of operation for communicating with I2C slave device.

13. Explain Serial Peripheral Interface (SPI) for onboard communication.

14. Write the features of the following:

    a. IrDA

    b. Wi-Fi

    c. Bluetooth

    d. Zig-Bee