EMBEDDED SYSTEM: AN INTRODUCTION

1.1 DEFINITION OF AN EMBEDDED SYSTEM

- An embedded system is a combination of 3 things:
  a. Hardware
  b. Software
  c. Mechanical Components. And it is supposed to do one specific task only.

- **Example 1: Washing Machine**

  A washing machine from an embedded systems point of view has:
  a. Hardware: Buttons, Display & buzzer, electronic circuitry.
  b. Software: It has a chip on the circuit that holds the software which drives controls & monitors the various operations possible.
  c. Mechanical Components: the internals of a washing machine which actually wash the clothes control the input and output of water, the chassis itself.

- **Example 2: Air Conditioner**

  An Air Conditioner from an embedded systems point of view has:
  a. Hardware: Remote, Display & buzzer, Infrared Sensors, electronic circuitry.
  b. Software: It has a chip on the circuit that holds the software which drives controls & monitors the various operations possible. The software monitors the external temperature through the sensors and then releases the coolant or suppresses it.
  c. Mechanical Components: the internals of an air conditioner the motor, the chassis, the outlet, etc

- An embedded system is designed to do a specific job only. Example: a washing machine can only wash clothes, an air conditioner can control the temperature in the room in which it is placed.

- The hardware & mechanical components will consist all the physically visible things that are used for input, output, etc.

- An embedded system will always have a chip (either microprocessor or microcontroller) that has the code or software which drives the system.

1.2 HISTORY OF EMBEDDED SYSTEM

- The first recognised embedded system is the Apollo Guidance Computer(AGC) developed by MIT lab.
- AGC was designed on 4K words of ROM & 256 words of RAM.
- The clock frequency of first microchip used in AGC was 1.024 MHz.
- The computing unit of AGC consists of 11 instructions and 16 bit word logic.

- It used 5000 ICs.
- The UI of AGC is known DSKY(display/keyboard) which resembles a calculator type keypad with array of numerals.
- The first mass-produced embedded system was guidance computer for the Minuteman-I missile in 1961.
- In the year 1971 Intel introduced the world's first microprocessor chip called the 4004, was designed for use in business calculators. It was produced by the Japanese company Busicom.

1.3 EMBEDDED SYSTEM & GENERAL PURPOSE COMPUTER

The Embedded System and the General purpose computer are at two extremes. The embedded system is designed to perform a specific task whereas as per definition the general purpose computer is meant for general use. It can be used for playing games, watching movies, creating software, work on documents or spreadsheets etc.

Following are certain specific points of difference between embedded systems and general purpose computers:

Criteria	**General Purpose Computer**	**Embedded system**
Contents	It is combination of generic hardware and a general purpose OS for executing a variety of applications.	It is combination of special purpose hardware and embedded OS for executing specific set of applications
Operating System	It contains general purpose operating system	It may or may not contain operating system.
Alterations	Applications are alterable by the user.	Applications are non-alterable by the user.
Key factor	Performance" is key factor.	Application specific requirements are key factors.
Power Consumption	More	Less
Response Time	Not Critical	Critical for some applications

1.4 CLASSIFICATION OF EMBEDDED SYSTEM

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

- On generation
- On complexity & performance
- On deterministic behaviour
- On triggering

1.5.1 On generation

1. **First generation(1G):**
   - Built around 8bit microprocessor & microcontroller.
   - Simple in hardware circuit & firmware developed.
   - Examples: Digital telephone keypads.

2. **Second generation(2G):**
   - Built around 16-bit µp & 8-bit µc.
   - They are more complex & powerful than 1G µp & µc.
   - Examples: SCADA systems

3. **Third generation(3G):**
   - Built around 32-bit µp & 16-bit µc.
   - Concepts like Digital Signal Processors(DSPs), Application Specific Integrated Circuits(ASICs) evolved.
   - Examples: Robotics, Media, etc.

4. **Fourth generation**:
   - Built around 64-bit µp & 32-bit µc.
   - The concept of System on Chips (SoC), Multicore Processors evolved.
   - Highly complex & very powerful.
   - Examples: Smart Phones.

1.5.2 On complexity & performance

1. **Small-scale:**
   - Simple in application need
   - Performance not time-critical.
   - Built around low performance & low cost 8 or 16 bit µp/µc.
   - 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 operating system.
   - Examples: Industrial machines.

3. **Large-scale:**
   - Highly complex hardware & firmware.
   - Built around 32 or 64 bit RISC µp/µc or PLDs or Multicore Processors.
   - Response is time-critical. Examples: Mission critical applications.

1.5.3 On deterministic behaviour

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

1.5.4 On triggering

- Embedded systems which are “Reactive” in nature can be based on triggering.
- Reactive systems can be:
  - ✓ Event triggered
  - ✓ Time triggered

1.6 APPLICATION OF EMBEDDED SYSTEM

The application areas and the products in the embedded domain are countless.

1. Consumer Electronics: Camcorders, Cameras.
2. Household appliances: Washing machine, Refrigerator.
3. Automotive industry: Anti-lock breaking system(ABS), engine control.
4. Home automation & security systems: Air conditioners, sprinklers, fire alarms.
5. Telecom: Cellular phones, telephone switches.
6. Computer peripherals: Printers, scanners.
7. Computer networking systems: Network routers and switches.
8. Healthcare: EEG, ECG machines.
9. Banking & Retail: Automatic teller machines, point of sales.
10. Card Readers: Barcode, smart card readers.

1.7 PURPOSE OF EMBEDDED SYSTEM

1. DataCollection/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 embedded systems 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.

2. Data communication

- Embedded data communication systems are deployed in applications from complex satellite communication to simple home networking systems.
- The transmission of data is achieved either by a wire-line medium or by a wire-less medium.
- Data can either be transmitted by analog means or by digital means.
- Wireless modules-Bluetooth, Wi-Fi.
- Wire-line modules-USB, TCP/IP.
- Network hubs, routers, switches are examples of dedicated data transmission embedded systems.

3. Data signal processing

- Embedded systems with signal processing functionalities are employed in applications demanding signal processing like speech coding, audio video codec, transmission applications etc.
- A digital hearing aid is a typical example of an embedded system employing data processing.
- Digital hearing aid improves the hearing capacity of hearing impaired person

4. Monitoring

- All embedded products coming under the medical domain are with monitoring functions.
- Electro cardiogram machine is intended to do the monitoring of the heartbeat of a patient but it cannot impose control over the heartbeat.
- Other examples with monitoring function are digital CRO, digital multi-meters, and logic analyzers.

5. Control

- A system with control functionality contains both sensors and actuators.
- Sensors are connected to the input port for capturing the changes in environmental variable and the actuators connected to the output port are controlled according to the changes in the input variable.
- Air conditioner system used to control the room temperature to a specified limit is a typical example for CONTROL purpose.

6. Application specific user interface

- Buttons, switches, keypad, lights, bells, display units etc are application specific user interfaces.
- Mobile phone is an example of application specific user interface.
- In mobile phone the user interface is provided through the keypad, system speaker, vibration alert etc.

1.8 REVIEW QUESTIONS

1. Define Embedded System with the help of Microwave Owen as an example
2. Differentiate between general purpose computers & embedded systems
3. Give a classification of embedded systems
4. List some applications of embedded systems
5. Explain the various possible purposes of using and embedded system.

❖❖❖❖

ELEMENTS OF EMBEDDED SYSTEMS

3.1 ELEMENTS OF EMBEDDED SYSTEMS.

- As defined earlier, an embedded system is a combination of 3 things:
  d. Hardware
  e. Software
  f. Mechanical Components

And it is supposed to do one specific task only.

Diagrammatically an embedded system can be represented as follows:

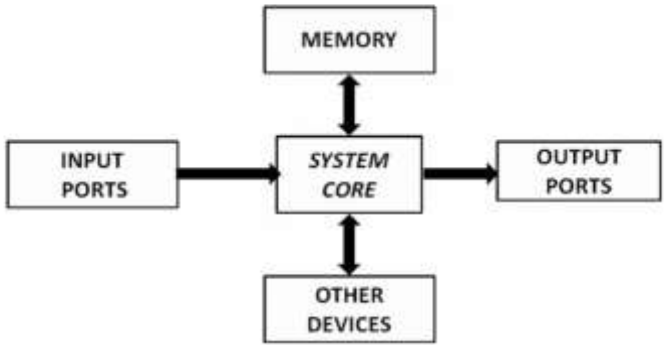


Figure 2.0 : Elements of an Embedded System

- Embedded systems are basically designed to regulate a physical variable (such Microwave Oven) or to manipulate the state of some devices by sending some signals to the actuators or devices connected to the output port system (such as temperature in Air Conditioner), in response to the input signal provided by the end users or sensors which are connected to the input ports.
- Hence the embedded systems can be viewed as a reactive system.
- Examples of common user interface input devices are keyboards, push button, switches, etc.
- The memory of the system is responsible for holding the code (control algorithm and other important configuration details).
- An embedded system without code (i.e. the control algorithm) implemented memory has all the peripherals but is not capable of making decisions depending on the situational as well as real world changes.
- Memory for implementing the code may be present on the processor or may be implemented as a separate chip interfacing the processor In a controller based embedded system, the controller may contain internal memory for storing code
- Such controllers are called Micro-controllers with on-chip ROM, eg. Atmel AT89C51.

2.3 CASE STUDIES (EXAMPLES)

Here are some case studies on some commonly used embedded systems that will help to better understand the concept.

2.3.1 Washing Machine

Let us see the important parts of the washing machine; this will also help us understand the working of the washing machine:

1) **Water inlet control valve**: Near the water inlet point of the washing there is water inlet control valve. When you load the clothes in washing machine, this valve gets opened automatically and it closes automatically depending on the total quantity of the water required. The water control valve is actually the solenoid valve.

2) **Water pump**: The water pump circulates water through the washing machine. It works in two directions, re-circulating the water during wash cycle and draining the water during the spin cycle.

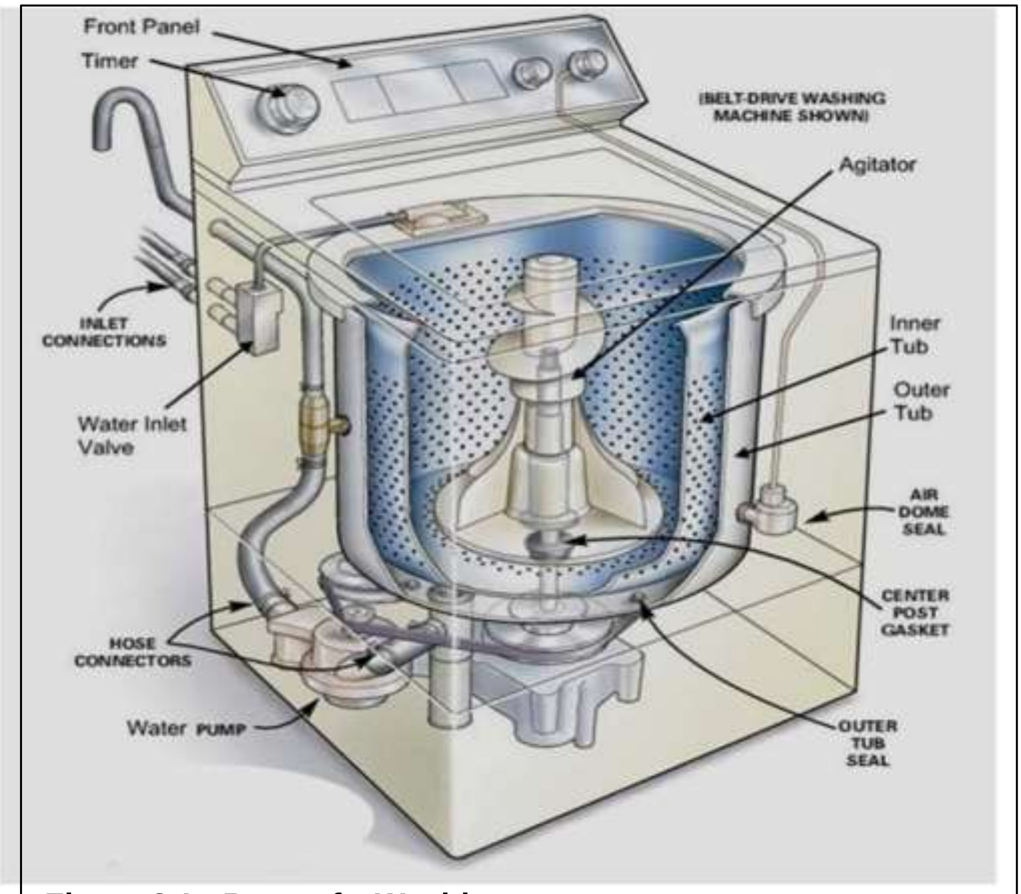


Figure 2.1 : Parts of a Washing

3) **Tub**: There are two types of tubs in the washing washing machine: inner and outer. The clothes are loaded in the inner tub, where the clothes are washed, rinsed and dried. The inner tub has small holes for draining the water. The external tub covers theinner tub and supports it during various cycles of clothes washing.

4) **Agitator or rotating disc**: The agitator is located inside the tub of the washing machine. It is the important part of the washing machine that actually performs the cleaning operation of the clothes. During the wash cycle the agitator rotates continuously and produces strong rotating currents within the water due to which the clothes also rotate inside the tub. The rotation of the clothes within water containing the detergent enables the removal of the dirt particles from the fabric of the clothes. Thus the agitator produces most important function of rubbing the clothes with each other as well as with water.

   In some washing machines, instead of the long agitator, there is a disc that contains blades on its upper side. The rotation of the disc and the blades produce strong currents within the water and the rubbing of clothes that helps in removing the dirt from clothes.

5) **Motor of the washing machine**: The motor is coupled to the agitator or the disc and produces it rotator motion. These are multispeed motors, whose speed can be changed as per the requirement. In the fully automatic washing machine the speed of the motor i.e. the agitator changes automatically as per the load on the washing machine.

6) **Timer**: The timer helps setting the wash time for the clothes manually. In the automatic mode the time is set automatically depending upon the number of clothes inside the washing machine.

7) **Printed circuit board (PCB)**: The PCB comprises of the various electronic components and circuits, which are programmed to perform in unique ways depending on the load conditions (the condition and the amount of clothes loaded in the washing machine). They are sort of artificial intelligence devices that sense the various external conditions and take the decisions accordingly. These are also called as fuzzy logic systems. Thus the PCB will calculate the total weight of the clothes, and find out the quantity of water and detergent required, and the total time required for washing the clothes. Then they will decide the time required for washing and rinsing. The entire processing is done on a kind of processor which may be a microprocessor or microcontroller.

8) **Drain pipe**: The drain pipe enables removing the dirty water from the washing that has been used for the washing purpose.

2.3.2 Microwave Owen

Let us see the important parts of the microwave oven; this will also help us understand the working of the washing machine:

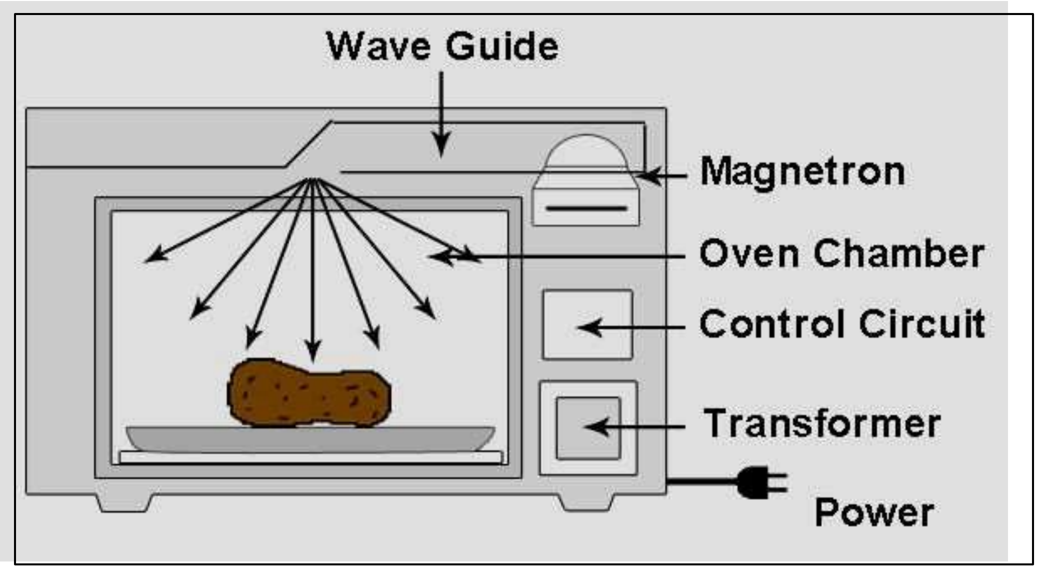


Figure 2.3 : Parts of a Microwave Owen

A microwave oven consists of:

1. A high voltage transformer, which passes energy to the magnetron
2. A cavity magnetron,
3. A Control circuit with a microcontroller,
4. A waveguide, and
5. A cooking chamber

1. A **Transformer** transfers electrical energy through a circuit by magnetic coupling without using motion between parts. These are used for supplying power to the magnetron.

2. A **Cavity magnetron** is a microwave antenna placed in a vacuum tube and oscillated in an electromagnetic field in order to produce high GHz microwaves. Magnetrons are used in microwave ovens and radar systems.

3. A **control circuit** with a microcontroller is integrated on a circuit board. The microcontroller controls the waveguide and the entire unit so the microwaves are emitted at a constant rate.

4. A **Waveguide** is any linear structure that guides electromagnetic waves for the purpose of transmitting power or signals.

   Generally constructed of a hollow metal pipe. Placing a waveguide into a vacuum causes radio waves to scatter.

5. A **Cooking Chamber** is a microwave safe container the prevents microwaves from escaping. The door has a microwave proof mesh

with holes that are just small enough that microwaves can't pass through but lightwaves can. The cooking chamber itself is a Faraday cage enclosure which prevents the microwaves from escaping into the environment. The oven door is usually a glass panel for easy viewing, but has a layer of conductive mesh to maintain the shielding.

2.3.3 Automotive Embedded System (AES)

- The Automotive industry is one of the major application domains of embedded systems.
- Automotive embedded systems are the one where electronics take control over the mechanical system. Ex. Simple viper control.
- The number of embedded controllers in a normal vehicle varies somewhere between 20 to 40 and can easily be between 75 to 100 for more sophisticated vehicles.
- One of the first and very popular use of embedded system in automotive industry was microprocessor based fuel injection.
- Some of the other uses of embedded controllers in a vehicle are listed below:
  a. Air Conditioner
  b. Engine Control
  c. Fan Control
  d. Headlamp Control
  e. Automatic break system control
  f. Wiper control
  g. Air bag control
  h. Power Windows
- AES are normally built around microcontrollers or DSPs or a hybrid of the two and are generally known as Electronic Control Units (ECUs).
- **Types Of Electronic Control Units(ECU)**

1. **High-speed Electronic Control Units (HECUs):**
   a. HECUs are deployed in critical control units requiring fast response.
   b. They Include fuel injection systems, antilock brake systems, engine control, electronic throttle, steering controls, transmission control and central control units.
2. **Low Speed Electronic Control Units (LECUs):-**
   a. They are deployed in applications where response time is not so critical.
   b. They are built around low cost microprocessors and microcontrollers and digital signal processors.
   c. Audio controller, passenger and driver door locks, door glass control etc.

- **Automotive Communication Buses**

  Embedded system used inside an automobile communicate with each other using serial buses. This reduces the wiring required.

  Following are the different types of serial Interfaces used in automotive embedded applications:

  a. **Controller Area Network (CAN):-**

  - CAN bus was originally proposed by Robert Bosch.
  - It supports medium speed and high speed data transfer
  - CAN is an event driven protocol interface with support for error handling in data transmission.

  b. **Local Interconnect Network (LIN):-**

  - LIN bus is single master multiple slave communication interface with support for data rates up to 20 Kbps and is used for sensor/actuator interfacing
  - LIN bus follows the master communication triggering to eliminate the bus arbitration problem
  - LIN bus applications are mirror controls , fan controls , seat positioning controls

  c. **Media-Oriented System Transport(MOST):-**

  - MOST is targeted for automotive audio/video equipment interfacing
  - A MOST bus is a multimedia fiber optics point–to-point network implemented in a star , ring or daisy chained topology over optical fiber cables.
  - MOST bus specifications define the physical as well as application layer , network layer and media access control.

2.4 REVIEW QUESTIONS

1. What is an embedded system? What are the working elements of an embedded system?
2. Explain the working of embedded system with respect to:
   B. Washing Machine
   C. MICROWAVE Owen
3. Conduct case studies for working of embedded systems for the following topics:
   A. Air Conditioner
   B. Automobile

CORE THE OF EMBEDDED SYSTEM

3.1 INTRODUCTION

The first two chapters attempted on explain what an embedded system is about and what the working parts are. This chapter attempts to go deeper and explain the core of embedded system along with other related topics.

3.2 CORE OF EMBEDDED SYSTEMS

Embedded systems are domain and application specific and are built around a central core. The core of the embedded system falls into any of the following categories:

1. General purpose and Domain Specific Processors
    1.1. Microprocessors
    1.2. Microcontrollers
    1.3. Digital Signal Processors
2. Application Specific Integrated Circuits. (ASIC)
3. Programmable logic devices(PLD's)
4. Commercial off-the-shelf components (COTs)

3.2.1 GENERAL PURPOSE AND DOMAIN SPECIFIC PROCESSOR.

- Almost 80% of the embedded systems are processor/ controller based.
- The processor may be microprocessor or a microcontroller or digital signal processor, depending on the domain and application.

3.2.1.1 MICROPROCESSORS

- A microprocessor is a silicon chip representing a central processing unit.
- A microprocessor is a dependent unit and it requires the combination of other hardware like memory, timer unit, and *interrupt* controller, etc. for proper functioning.
- Developers of microprocessors.
    - Intel – *Intel 4004* – November 1971(4-bit).
    - Intel – *Intel 4040.*
    - Intel – *Intel 8008* – April 1972.
    - Intel – *Intel 8080* – April 1974(8-bit).
    - Motorola – *Motorola 6800.*
    - Intel – *Intel 8085* – 1976.
    - Zilog - *Z80* – July 1976.

- Architectures used for processor design are Harvard or Von-Neumann.

Harvard architecture	**Von-Neumann architecture**
▣ It has separate buses for instruction as well as data fetching.	▣ It shares single common bus for instruction and data fetching.
▣ Easier to pipeline, so high performance can be achieve.	▣ Low performance as compared to Harvard architecture.
▣ Comparatively high cost.	▣ It is cheaper.
▣ Since data memory and program memory are stored physically in different locations, no chances exist for accidental corruption of program memory.	▣ Accidental corruption of program memory may occur if data memory and program memory are stored physically in the same chip,

- RISC and CISC are the two common Instruction Set Architectures (ISA) available for processor design.

RISC	CISC
▣ Reduced Instruction Set Computing ▣ It contains lesser number of instructions. ▣ Instruction pipelining and increased execution speed. ▣ Orthogonal instruction set(allows each instruction to operate on any register and use any addressing mode. ▣ Operations are performed on registers only, only memory operations are load and store. ▣ A larger number of registers are available. ▣ Programmer needs to write more code to execute a task since instructions are simpler ones. ▣ It is single, fixed length instruction. ▣ Less silicon usage and pin count. ▣ With Harvard Architecture.	▣ Complex Instruction Set Computing ▣ It contains greater number of instructions. ▣ Instruction pipelining feature does not exist. ▣ Non-orthogonalset(all instructions are not allowed to operate on any register and use any addressing mode. ▣ Operations are performed either on registers or memory depending on instruction. ▣ The number of general purpose registers are very limited. ▣ 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 instruction in RISC. ▣ It is variable length instruction. ▣ More silicon usage since more additional decoder logic is required to implement the complex instruction decoding. ▣ Can be Harvard or Von-Neumann Architecture.

3.2.1.2 MICROCONTROLLERS.

- 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.

- Texas Instrument's TMS 1000 Is considered as the world's first microcontroller.
- Some embedded system application require only 8 bit controllers whereas some requiring superior performance and computational needs demand 16/32 bit controllers.
- The instruction set of a microcontroller can be RISC or CISC.
- Microcontrollers are designed for either general purpose application requirement or domain specific application requirement.

3.2.1.3 Digital Signal Processors

- DSP are powerful special purpose 8/16/32 bit microprocessor designed to meet the computational demands and power constraints of today's embedded audio, video and communication applications.
- DSP are 2 to 3 times faster than general purpose microprocessors in signal processing applications. This is because of the architectural difference between DSP and general purpose microprocessors.
- DSPs implement algorithms in hardware which speeds up the execution whereas general purpose processor implement the algorithm in software and the speed of execution depends primarily on the clock for the processors.
- DSP includes following key units:

  i. **Program memory:** It is a memory for storing the program required by DSP to process the data.

  ii. **Data memory:** It is a working memory for storing temporary variables and data/signal to be processed.

  iii. **Computational engine:** It performs the signal processing in accordance with the stored program memory computational engine incorporated many specialized arithmetic units and each of them operates simultaneously to increase the execution speed. It also includes multiple hardware shifters for shifting operands and saves execution time.

  iv. **I/O unit:** It acts as an interface between the outside world and DSP. It is responsible for capturing signals to be processed and delivering the processed signals.

  - Examples: Audio video signal processing, telecommunication and multimedia applications.
  - SOP(Sum of Products) calculation, convolution, FFT(Fast Fourier Transform), DFT(Discrete Fourier Transform), etc are some of the operation performed by DSP.

3.2.2 Application Specific Integrated Circuits. (ASIC)

- ASICs is a microchip design to perform a specific and unique applications.
- Because of using single chip for integrates several functions there by reduces the system development cost.
- Most of the ASICs are proprietary (which having some trade name) products, it is referred as Application Specific Standard Products(ASSP).
- As a single chip ASIC consumes a very small area in the total system. Thereby helps in the design of smaller system with high capabilities or functionalities.
- The developers of such chips may not be interested in revealing the internal detail of it .

3.2.3 Programmable logic devices(PLD's)

- A PLD is an electronic component. It used to build digital circuits which are reconfigurable.
- A logic gate has a fixed function but a PLD does not have a defined function at the time of manufacture.
- PLDs offer customers a wide range of logic capacity, features, speed, voltage characteristics.
- PLDs can be reconfigured to perform any number of functions at any time.
- A variety of tools are available for the designers of PLDs which are inexpensive and help to develop, simulate and test the designs.
- PLDs having following two major types.
  1) **CPLD(Complex Programmable Logic Device):**
     CPLDs offer much smaller amount of logic up to 1000 gates.
  2) **FPGAs(Field Programmable Gate Arrays):**
     It offers highest amount of performance as well as highest logic density, the most features.
- **Advantages of PLDs :-**
  1) PLDs offer customer much more flexibility during the design cycle.
  2) PLDs do not require long lead times for prototypes or

production parts because PLDs are already on a distributors shelf and ready for shipment.

3) PLDs can be reprogrammed even after a piece of equipment is shipped to a customer

3.2.4 Commercial off-the-shelf components(COTs)

1) A Commercial off the Shelf product is one which is used 'as- is'.
2) The COTS components itself may be develop around a general purpose or domain specific processor or an ASICs or a PLDs.

3) The major advantage of using COTS is that they are readily available in the market, are chip and a developer can cut down his/her development time to a great extent

4) The major drawback of using COTS components in embedded design is that the manufacturer of the COTS component may withdraw the product or discontinue the production of the COTS at any time if rapid change in technology occurs.

5) **Advantages of COTS:**
    1) Ready to use
    2) Easy to integrate
    3) Reduces development time

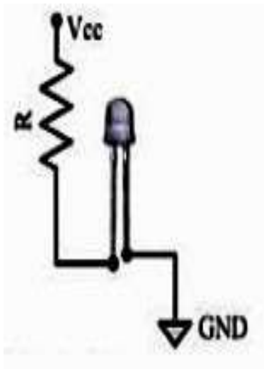
6) **Disadvantages of COTS:**
    1) No operational or manufacturing standard (all proprietary)

    2) Vendor or manufacturer may discontinue production of a particular COTS product

3.3 SENSORS & ACTUATORS

- **Sensor**
  - A Sensor is used for taking Input
  - It is a transducer that converts energy from one form to another for any measurement or control purpose
  - Ex. A Temperature sensor

- **Actuator**
  - Actuator is used for output.
  - It is a transducer that may be either mechanical or electrical which converts signals to corresponding physical actions.
  - Ex. LED (Light Emitting Diode)
  - LED is a p-n junction diode and contains a **CATHODE** and **ANODE**
  - For functioning the anode is connected to +ve end of power supply and cathode is connected to –ve end of power supply.
  - The maximum current flowing through the LED is limited by connecting a RESISTOR in series between the power supply and LED as shown in the figure below



- There are two ways to interface an LED to a microprocessor/microcontroller:

1. **The Anode of LED is connected to the port pin and cathode to Ground :** In this approach the port pin sources the current to the LED when it is at logic high(ie. 1).

2. **The Cathode of LED is connected to the port pin and Anode to Vcc :** In this approach the port pin sources the current to the LED when it is at logic high (ie. 1).
   Here the port pin sinks the current and the LED is turned ON when the port pin is at Logic low (ie. 0)

3.4 COMMUNICATIONINTERFACES

For any embedded system, the communication interfaces can broadly classified into:

1. Onboard Communication Interfaces

- These are used for internal communication of the embedded system i.e: communication between different components present on the system.

- Common examples of onboard interfaces are:
  - Inter Integrated Circuit (I2C)
  - Serial Peripheral Interface (SPI)
  - Universal Asynchronous Receiver Transmitter (UART)
  - 1-Wire Interface
  - Parallel Interface

- Example :**Inter Integrated Circuit (I2C)**
  - It is synchronous
  - Bi-directional, half duplex , two wire serial interface bus
  - Developed by Phillips semiconductors in 1980

  - It comprises of two buses :
    1. Serial clock –SCL
    2. Serial Data – SDA
  - SCL generates synchronization clock pulses
  - SDA transmits data serially across devices
  - I2C is a shared bus system to which many devices can be connected
  - Devices connected by I2C can act as either master or slave
  - The master device is responsible for controlling communication by initiating/ terminating data transfer.
  - Devices acting as slave wait for commands from the master and respond to those commands.

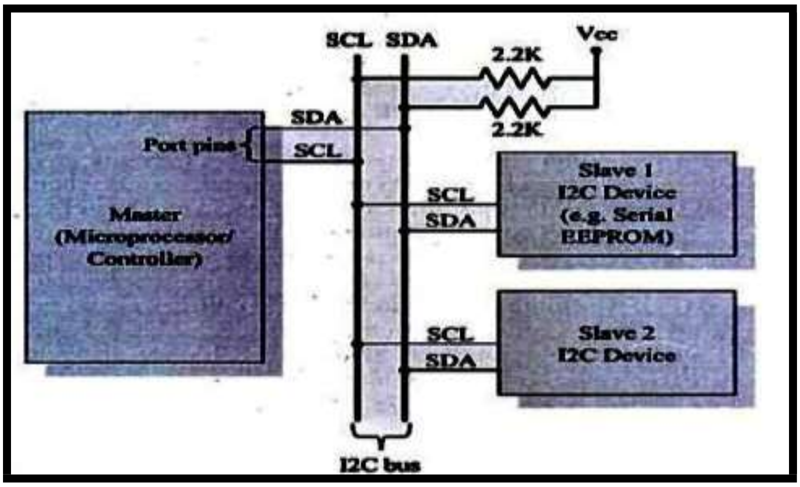


Figure: I2C Bus Interfacing

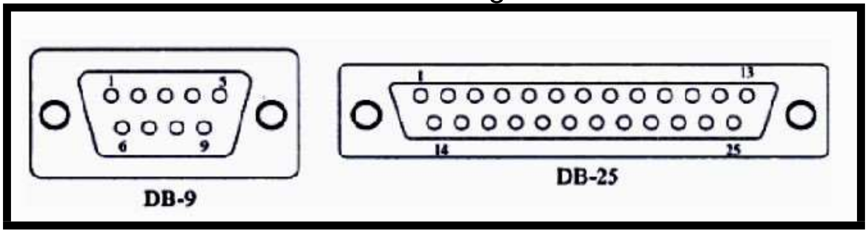
2. **External or Peripheral Communication Interfaces**

- These are used for external communication of the embedded system i.e: communication of different components present on the system with external or peripheral components/devices.

- Common examples of external interfaces are:
  - RS-232 C & RS-485
  - Universal Serial Bus (USB)
  - IEEE 1394 (Firewire)
  - Infrared (IrDA)
  - Bluetooth
  - Wi-Fi
  - Zig Bee
  - General Packet Radio Service (GPRS)
  - **Example: RS-232 C & RS-485**

  - It is wired, asynchronous, serial, full duplex communication
  - RS 232 interface was developed by EIA (Electronic Industries Associates) In early 1960s
  - RS 232 is the extension to UART for external communications
  - RS-232 logic levels use:
  - +3 to +25 volts to signify a "Space" (Logic 0) and
  - -3 to -25 volts to signify a "Mark" (logic 1).
  - RS 232 supports two different types of connectors :

  - **DB 9 and DB 25** as shown in figure below



  - RS 232 interface is a point to point communication interface and the devices involved are called as Data Terminating Equipment (DTE) And Data Communications Terminating Equipment (DCE)
  - Embedded devices contain UART for serial transmission and generate signal levels as per TTL/CMOS logic.
  - A level translator IC (like Max 232) is used for converting the signal lines from UART to RS 232 signal lines for communication.
  - The vice versa is performed on the receiving side.
  - Converter chips contain converters for both transmitters and receivers
  - RS 232 is used only for point to point connections

- It is susceptible to noise and hence is limited to short distances only
- RS 422 is another serial interface from EIA.
- It supports multipoint connections with 1 transmitter and 10 receivers.
- It supports data rates up to 100Kbps and distance up to 400 ft
- RS 485 is enhanced version of RS 422 and supports up to 32 transmitters and 32 receivers

3.5 REVIEW QUESTIONS

1. What do you mean by core of the embedded system? What is its significance? What are the possible options that can be used as a core?
2. Distinguish between Microprocessor & Microcontroller
3. Explain the different types of processors according to their system bus architecture
4. Explain the different types of processors according to Instruction set Architecture
5. Explain the different types of processors according to Endianness
6. Write short note on :
   i. DSP
   ii. PLD
   iii. ASIC
   iv. COTS
7. Explain Communication Interfaces with respect to embedded system
8. Explain the following with example:
   1. Onboard communication interface
   2. Peripheral communication interface
9. Find out information and write case studies on the following communication interfaces:
   i. Infrared
   ii. WiFi
   iii. Zigbee
   iv. UART

❖❖❖❖

CHARACTERISTICS & QUALITY ATTRIBUTES OF EMBEDDED SYSTEMS

4.1 CHARACTERISTICS OF EMBEDDED SYSTEM

Following are some of the characteristics of an embedded system that make it different from a general purpose computer:

1. **Application and Domain specific**
   - An embedded system is designed for a specific purpose only. It will not do any other task.
   - Ex. A washing machine can only wash, it cannot cook
   - Certain embedded systems are specific to a domain: ex. A hearing aid is an application that belongs to the domain of signal processing.

2. **Reactive and Real time**
   - Certain Embedded systems are designed to react to the events that occur in the nearby environment. These events also occur real-time.
   - Ex. An air conditioner adjusts its mechanical parts as soon as it gets a signal from its sensors to increase or decrease the temperature when the user operates it using a remote control.
   - An embedded system uses Sensors to take inputs and has actuators to bring out the required functionality.

3. **Operation in harsh environment**
   - Certain embedded systems are designed to operate in harsh environments like very high temperature of the deserts or very low temperature of the mountains or extreme rains.
   - These embedded systems have to be capable of sustaining the environmental conditions it is designed to operate in.

4. **Distributed**
   - Certain embedded systems are part of a larger system and thus form components of a distributed system.
   - These components are independent of each other but have to work together for the larger system to function properly.
   - Ex. A car has many embedded systems controlled to its dash board. Each one is an independent embedded system yet the entire car can be said to function properly only if all the systems work together.

5. **Small size and weight**
   - An embedded system that is compact in size and has light weight will be desirable or more popular than one that is bulky and heavy.
   - Ex. Currently available cell phones. The cell phones that have the maximum features are popular but also their size and weight is an important characteristic.
   - For convenience users prefer mobile phones than phablets. (phone + tablet pc)

6. Power concerns

- It is desirable that the power utilization and heat dissipation of any embedded system be low.
- If more heat is dissipated then additional units like heat sinks or cooling fans need to be added to the circuit.
- If more power is required then a battery of higher power or more batteries need to be accommodated in the embedded system.

4.2 QUALITY ATTRIBUTES OF EMBEDDED SYSTEM

These are the attributes that together form the deciding factor about the quality of an embedded system.
There are two types of quality attributes are:-

1. **Operational Quality Attributes.**
   - These are attributes related to operation or functioning of an embedded system. The way an embedded system operates affects its overall quality.
2. **Non-Operational Quality Attributes.**
   - These are attributes **not** related to operation or functioning of an embedded system. The way an embedded system operates affects its overall quality.
   - These are the attributes that are associated with the embedded system before it can be put in operation.

4.3.1 Operational Attributes

a) Response

- Response is a measure of quickness of the system.
- It gives you an idea about how fast your system is tracking the input variables.
- Most of the embedded system demand fast response which should be real-time.

b) Throughput

- Throughput deals with the efficiency of system.
- It can be defined as rate of production or process of a defined process over a stated period of time.
- In case of card reader like the ones used in buses, throughput means how much transaction the reader can perform in a minute or hour or day.

c) Reliability

- Reliability is a measure of how much percentage you rely upon the proper functioning of the system .
- Mean Time between failures and Mean Time To Repair are terms used in defining system reliability

.

- Mean Time between failures can be defined as the average time the system is functioning before a failure occurs.
- Mean time to repair can be defined as the average time the system has spent in repairs.

d) Maintainability

- Maintainability deals with support and maintenance to the end user or a client in case of technical issues and product failures or on the basis of a routine system checkup
- It can be classified into two types :-

1. Scheduled or Periodic Maintenance

- This is the maintenance that is required regularly after a periodic time interval.
- Example :
  - Periodic Cleaning of Air Conditioners
  - Refilling of printer cartridges.

2. Maintenance to unexpected failure

- This involves the maintenance due to a sudden breakdown in the functioning of the system.
- Example:
1. Air conditioner not powering on
2. Printer not taking paper in spite of a full paper stack

e) Security

- stones of information security.Confidentiality, Integrity and Availability are three corner
- Confidentiality deals with protection data from unauthorized disclosure.
- Integrity gives protection from unauthorized modification.
- Availability gives protection from unauthorized user
- Certain Embedded systems have to make sure they conform to the security measures. Ex. An Electronic Safety Deposit Locker can be used only with a pin number like a password.

f) Safety

- Safety deals with the possible damage that can happen to the operating person and environment due to the breakdown of an embedded system or due to the emission of hazardous materials from the embedded products.
- A safety analysis is a must in product engineering to evaluate the anticipated damage and determine the best course of action to bring down the consequence of damages to an acceptable level.

4.3.2 Non Operational Attributes

a) Testability and Debug-ability

- It deals with how easily one can test his/her design, application and by which mean he/she can test it.
- In hardware testing the peripherals and total hardware function in designed manner
- Firmware testing is functioning in expected way
- Debug-ability is means of debugging the product as such for figuring out the probable sources that create unexpected behavior in the total system

b) Evolvability

- For embedded system, the qualitative attribute “Evolvability” refer to ease with which the embedded product can be modified to take advantage of new firmware or hardware technology.

c) Portability

- Portability is measured of “system Independence”.
- An embedded product can be called portable if it is capable of performing its operation as it is intended to do in various environments irrespective of different processor and or controller and embedded operating systems.

d) Time to prototype and market

- Time to Market is the time elapsed between the conceptualization of a product and time at which the product is ready for selling or use
- Product prototyping help in reducing time to market.
- Prototyping is an informal kind of rapid product development in which important feature of the under consider are develop.
- In order to shorten the time to prototype, make use of all possible option like use of reuse, off the self component etc**.**

e) Per unit and total cost

- Cost is an important factor which needs to be carefully monitored. Proper market study and cost benefit analysis should be carried out before taking decision on the per unit cost of the embedded product.
- When the product is introduced in the market, for the initial period the sales and revenue will be low
- There won’t be much competition when the product sales and revenue increase.

EMBEDDED HARDWARE FROM SOFTWARE PROGRAMMERS PERSPECTIVE

7.1 COMPONENTS ON AN EMBEDDED SYSTEM

- Before the programmer can start to code anything, he has to invest some time in understand the functioning of the embedded system.

- He is expected to understand the following things:
  a. Functioning or purpose of the embedded system
  b. Individual components involved
  c. The way data flows through the components of an embedded system.

- Consider an example of an embedded system intended to be used as a printer-sharing device. This device is attached to a printer and allows access to two computers through serial interface and one printer through a parallel interface.

- The diagram below describes the way the devices are connected to each other. Data to be printed is accepted from either serial port, held in RAM until the printer is ready for more data, and delivered to the printer via the parallel port. The software that makes all of this happen is stored in ROM.

- The working or execution of the code is brought about by the processor. The processor knows two types of components: memory and peripherals.

- Memories are for data and code storage and retrieval. Ex. RAM & ROM

- Peripherals are specialized hardware devices that either coordinate interaction with the outside world (I/O) or perform a specific hardware function. Ex. Serial Port

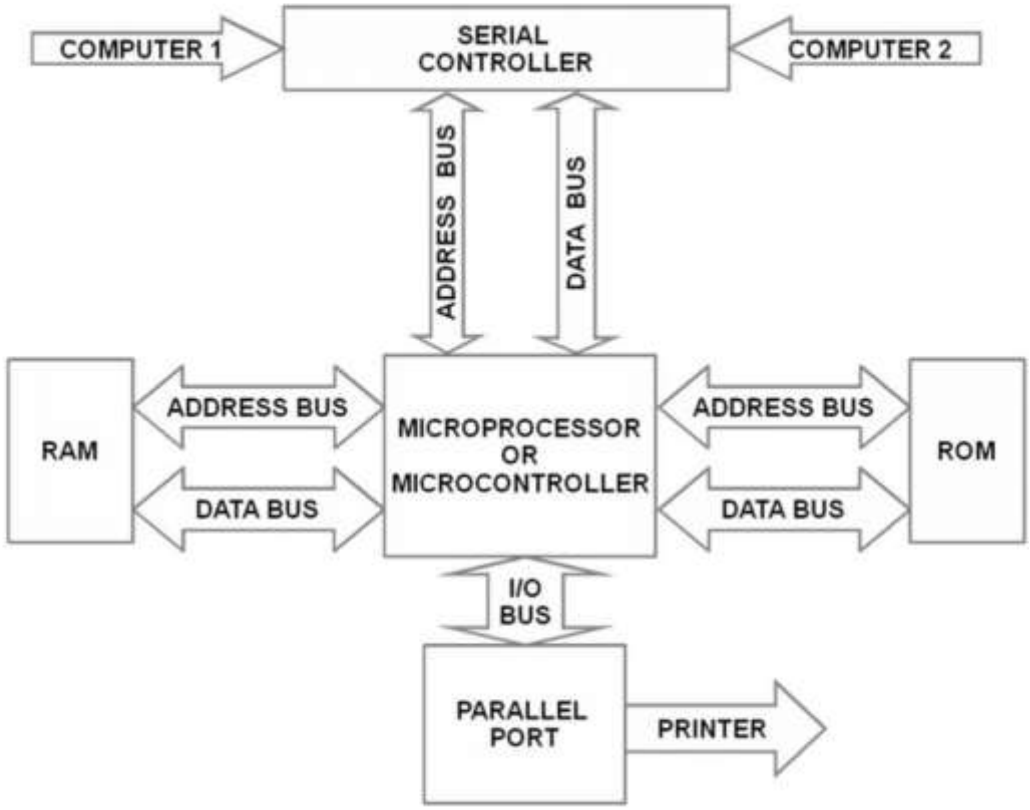


Figure: Components involved in an printer sharing device

- Certain processors like intel communicate with these memories and peripherals with two distinct address spaces.

- The first address space is called the memory space and is intended mainly for memory devices; the second is reserved exclusively for peripherals and is called the I/O space.

- When peripherals are located in I/O space they are called I/O Mapped peripheral else when peripherals are located in memory space they are called Memory Mapped peripherals or memory mapped I/O.

- If given a choice, Memory mapped peripherals are better because it has advantages for both the hardware and software developers. It is attractive to the hardware developer because he might be able to eliminate the I/O space, and some of its associated wires, altogether. It is attractive to the software developer who is able to use pointers, data structures, and unions to interact with the peripherals more easily and efficiently