**1.0 EMBEDDED SYSTEMS**

**Embedded**: something that is attached to another thing

**System:** is a set of interrelated part/component which is designed, developed to perform common tasks or to do some specific works for which it has been created

**Embedded System**: is an electronic/electro-mechanical system designed to perform a specific function and is a combination of both hardware and firmware (software). E.g. Electronic Toys, Mobile Handsets, Washing Machines, Air Conditioners, Automotive Control Units, Set Top Box, DVD Player etc

1. **Computer**: A computer is a combination of hardware and software resources that integrate together and provides various functionalities to the user.

2. **Embedded Device**: An embedded device is a part of an integrated system that is formed as a combination of computer hardware and software for a specific function and which can operate without human interaction.

**Comparison between General Purpose Computing System And Embedded systems**

|  |  |  |
| --- | --- | --- |
| Criteria | General Purpose Computing System | Embedded System |
| Content | It is a combination of hardware, Software and a general-purpose operating system for executing a variety of application | It is a combination of special purpose hardware and embedded operating system for executing a specific set of application |
| Operating System | It has general purpose Operating system(GPOS) | It may or may not have operating system |
| Change | Application are alterable by the user | The firmware of embedded system is pre- programmed and it is not-alterable by the end-user |
| Key Factor | Performance is key factor | Application specific is the key factor |
| Power Consumption | More | Less generally they are battery operated. |
| Response time | Not Time-Critical | It is time-critical in some application |
| Examples | Desktop, Laptop | Air Conditioner, Refrigerator |

**2.0 HISTORY OF EMBEDDED SYSTEMS**

First Recognized Modern Embedded System: Apollo Guidance Computer (AGC) developed by Charles Stark Draper at the MIT Instrumentation Laboratory.

It has two modules

1. Command module (CM)
2. Lunar Excursion Module (LEM)

RAM size 256, 1K, 2K words

ROM size 4K, 10K, 36K words

Clock frequency is 1.024MHz

5000, 3-input RTL NOR gates are used

User interface is DSKY (display/Keyboard)

First Mass Produced Embedded System: Autonetics D-17 Guidance computer for Minuteman-I missile

**3.0 CLASSIFICATION OF EMBEDDED SYSTEMS**

A. Generation B. Complexity and Performance C. Deterministic Behaviour D. Triggering

E. Functional Requirement

**3.1 Classification Of Embedded System Based On Generation**

1. **First Generation:** Built around 8 bit microprocessor and 4 bit microcontroller Simple in hardware circuit and firmware developed. Example: Digital telephone keypads.
2. **Second Generation:** Built around 16-bit microprocessor and 8 or 16 bit microcontroller. They are more complex and powerful than 1G microprocessor and microcontroller. Examples: data acquisition systems.
3. **Third Generation:** Built around 32-bit microprocessor and 16-bit microcontroller digital signal processing DSP, application specific integrated circuit ASIC .Examples: Robotics, Media.
4. Fourth Generation: Built around 64-bit microprocessor and 32-bit microcontroller System on chip (SoC), Multicore processors evolved Highly complex and powerful. Example: Smart phone

**3.2 Classification Of Embedded System Based On Complexity And Performance**

1. Small-Scale (Eg : Electronic toy) : Built around low performance and low cost 8-bit or 16-bit micro-controller/microprocessor. They can be powered by a battery. May or may not contain OS for its functioning. Simple in application need Performance not time-critical
2. Medium-Scale (Eg: Industrial machines): Built around medium performance and low cost 16-bit or 32-bit micro-controller/microprocessor. They usually contain embedded OS for functioning. Slightly complex in hardware and firmware requirement Java, C, C++ are the programming languages are used to develop medium scale embedded systems.
3. Large-Scale (Eg: Mission Critical Applications): Highly complex hardware and firmware. Built around 32 or 64 bit reduce instructions set computer RISC processors/controllers or multicore processors and programmable logic devices Response is time-critical. Contains real time purpose operation system RTOS for task scheduling, prioritization and management

**3.3 Classification Of Embedded System Based On Deterministic Behaviour**

This classification is applicable for Real Time Systems. The task execution behaviour for embedded systems may be deterministic or non-deterministic. Based on execution behaviour Real Time embedded systems are divided into Hard and Soft . These are classified in to two types

1. Soft Real time Systems: Missing a deadline may not be critical and can be tolerated to a certain degree
2. Hard Real time systems: Missing a program/task execution time deadline can have catastrophic consequences (financial, human loss of life, etc.)

**3.4 Classification Of Embedded System Based On Triggering**

Embedded Systems which are reactive in nature can be based on triggering. Reactive systems can be: Event triggered and Time triggered . These are classified into two types

1. Event Triggered: Activities within the system (e.g., task run-times) are dynamic and depend upon occurrence of different events.
2. Time triggered: Activities within the system follow a statically computed schedule (i.e., they are allocated time slots during which they can take place) and thus by nature are predictable.

**3.5 Classification Of Embedded System Based On Functional Requirement**

1. Stand Alone Embedded Systems; Stand alone embedded systems do not require a host system like a computer, it works by itself. It takes the input from the input ports either analog or digital and processes, calculates and converts the data and gives the resulting data through the connected device-Which either controls, drives or displays the connected devices. Examples for the stand alone embedded systems are microwave ovens, calculators and temperature measurement systems.
2. Real Time Embedded Systems: A real time embedded system is defined as, a system which gives a required o/p in a particular time. These types of embedded systems follow the time deadlines for completion of a task. Real time embedded systems are classified into two types such as soft and hard real time systems. Eg: Traffic Control System
3. Networked Embedded Systems: These types of embedded systems are related to a network to access the resources. The connected network can be LAN, WAN or the internet. The connection can be any wired or wireless. This type of embedded system is the fastest growing area in embedded system applications. The embedded web server is a type of system wherein all embedded devices are connected to a web server and accessed and controlled by a web browser. Example for the LAN networked embedded system is a home security system wherein all sensors are connected and run on the protocol TCP/IP. Eg:ATM
4. Mobile Embedded Systems: Mobile embedded systems are used in portable embedded devices like cell phones, mobiles etc. The basic limitation of these devices is the other resources and limitation of memory.

**4.0 MAJOR APPLICATION AREAS OF EMBEDDED SYSTEMS:**

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.

**5.0 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

**5.1 Data Collection/Storage/Representation**

Embedded Systems 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 digital (discrete) or analog (continuous). ES with analog data capturing techniques collect data directly in the form of analog signal whereas embedded system with digital data collection mechanism converts the analog signal to digital using analog to digital convertors. If the data is digital it can be directly captured by digital ES Eg: Digital Camera. Images are captured those may be stored within the memory of the camera. The captured image can also be presented to user through a graphic LCD unit.

**5.2 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

**5.3 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 etcComputational intensive systems, Employs Digital Signal Processors (DSPs)

**5.4 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.

**5.5 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.

**5.5 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 Eg: Mobile phone

**6.0 Characteristics Of Embedded Computing Applications**

1. Complex Algorithms
2. User Interface
3. Real Time
4. Multirate
5. Manufacturing Cost
6. Power
7. Complex algorithms: The operations performed by the microprocessor may be very sophisticated. For example, the microprocessor that controls an automobile engine must perform complicated filtering functions to optimize the performance of the car while minimizing pollution and fuel utilization.
8. User interface: Microprocessors are frequently used to control complex user interfaces that may include multiple menus and many options. The moving maps in Global Positioning System (GPS) navigation are good examples of sophisticated user interfaces.
9. To make things more difficult, embedded computing operations must often be performed to meet deadlines:
10. Real time: Many embedded computing systems have to perform in real time— if the data is not ready by a certain deadline, the system breaks. In some cases, failure to meet a deadline is unsafe and can even endanger lives. In other cases, missing a deadline does not create safety problems but does create unhappy customers— missed deadlines in printers, for example, can result in scrambled pages.
11. Multirate: Not only must operations be completed by deadlines, but many embedded computing systems have several real-time activities going on at the same time. They may simultaneously control some operations that run at slow rates and others that run at high rates. Multimedia applications are prime examples of multirate behaviour. The audio and video portions of a multimedia stream run at very different rates, but they must remain closely synchronized. Failure to meet a deadline on either the audio or video portions spoils the perception of the entire presentation.

**Costs of various sorts are also very important:**

1. Manufacturing cost: The total cost of building the system is very important in many cases. Manufacturing cost is determined by many factors, including the type of microprocessor used, the amount of memory required, and the types of I/O devices.
2. Power and energy: Power consumption directly affects the cost of the hardware, since a larger power supply may be necessary. Energy consumption affects battery life, which is important in many applications, as well as heat consumption, which can be important even in desktop applications.

7.0 **CHALLENGES IN EMBEDDED COMPUTING SYSTEM DESIGN**

1. How much hardware do we need?
2. How do we meet deadlines?
3. How do we minimize power consumption?
4. How do we design for upgradability?
5. Does it really work?
6. Complex testing
7. Limited observability and controllability
8. Restricted development environments
9. External constraints are one important source of difficulty in embedded system design. Let‘s consider some important problems that must be taken into account in embedded system design.

**7.1 How much hardware do we need?**

We have a great deal of control over the amount of computing power we apply to our problem. We cannot only select the type of microprocessor used, but also select the amount of memory, the peripheral devices, and more. Since we often must meet both performance deadlines and manufacturing cost constraints, the choice of hardware is important—too little hardware and the system fails to meet its deadlines, too much hardware and it becomes too expensive.

**7.2 How do we meet deadlines?**

The brute force way of meeting a deadline is to speed up the hardware so that the program runs faster. Of course, that makes the system more expensive. It is also entirely possible that increasing the CPU clock rate may not make enough difference to execution time, since the program‘s speed may be limited by the memory system.

**7.3 How do we minimize power consumption?**

In battery-powered applications, power consumption is extremely important. Even in non battery applications, excessive power consumption can increase heat dissipation. One way to make a digital system consume less power is to make it run more slowly, slowing down the system can obviously lead to missed deadlines. Careful design is required to slow down the noncritical parts of the machine for power consumption while still meeting necessary performance goals.

**7.4 How do we design for upgradability?**

The hardware platform may be used over several product generations or for several different versions of a product in the same generation, with few or no changes. However, we want to be able to add features by changing software.

**7.5 Does it really work?**

Reliability is always important when selling products—customers rightly expect that products they buy will work. Reliability is especially important in some applications. If we wait until we have a running system and try to eliminate the bugs, we will be too late—we won‘t find enough bugs, it will be too expensive to fix them, and it will take more time.

***Let‘s consider some ways in which the nature of embedded computing machines makes their design more difficult.***

* 1. **Complex testing**

Exercising an embedded system is generally more difficult than typing in some data. We may have to run a real machine in order to generate the proper data. The timing of data is often important, meaning that we cannot separate the testing of an embedded computer from the machine in which it is embedded.

**7.7Limited observability and controllability:**

Embedded computing systems usually do not come with keyboards and screens. This makes it more difficult to see what is going on and to affect the system‘s operation. We may be forced to watch the values of electrical signals on the microprocessor bus, for example, to know what is going on inside the system. Moreover, in real-time applications we may not be able to easily stop the system to see what is going on inside.

1. Restricted development environments: The development environments for embedded systems (the tools used to develop software and hardware) are often much more limited than those available for PCs and workstations.

8.0 **THE CORE OF THE EMBEDDED SYSTEMS**

The core of the embedded system falls into anyone of the following categories.

1. General Purpose and Domain Specific Processors

* Microprocessors
* Microcontrollers
* Digital Signal Processors

2. Programmable Logic Devices (PLDs)

3. Application Specific Integrated Circuits (ASICs)

4. Commercial off the shelf Components (COTS)

**8.1.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.

**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

**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

**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

**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 |

**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

**8.2 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

**3. 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). The ASSP is marketed to multiple customers just as a general-purpose product , but to a smaller number of customers since it is for a specific application. Some ASICs are proprietary products; the developers are not interested in revealing the internal details.

**4. 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. There is no need to design the module yourself and write the firmware. Everything will be readily supplied by the COTs manufacturer.

**9.0 COMMUNICATION INTERFACE**

Communication interface is essential for communicating with various subsystems of the embedded system and with the external world. The communication interface can be viewed in two different perspectives; namely;

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

2. Product level communication interface (External Communication Interface)

**1. Device/board level communication interface (Onboard Communication Interface):**

The communication channel which interconnects the various components within an embedded product is referred as Device/board level communication interface (Onboard Communication Interface)

Examples: Serial interfaces like I2C, SPI, UART, 1-Wire etc and Parallel bus interface I2C (Inter Integrated Circuit) Bus

SPI (Serial Peripheral Interface) Bus

UART (Universal Asynchronous Receiver Transmitter) 1-Wires Interface

Parallel 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.

Examples for wireless communication interface: Infrared (IR), Bluetooth (BT), Wireless LAN (Wi-Fi), Radio Frequency waves (RF), GPRS etc.

Examples for wired interfaces: RS-232C/RS-422/RS 485, USB, Ethernet (TCP-IP), IEEE 1394 port, Parallel port etc.