Embedded System and Internet of Things (ESIOT)

Introduction to Embedded Systems *(Unit-1):*

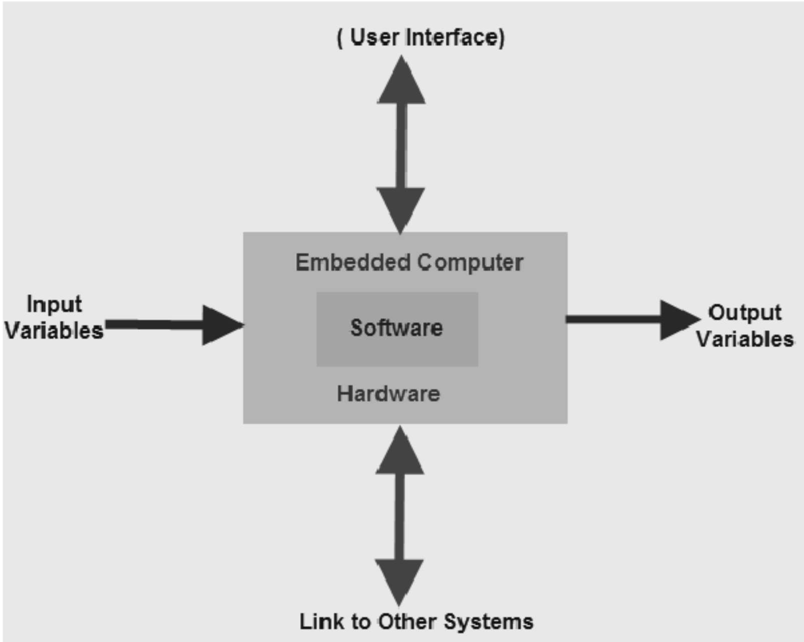
B.E.-CSE 3rd Sem.

Department of Interdisciplinary Courses in Engineering (DICE)

Embedded system

Embedded systems are a combination of hardware and software where software is usually known as firmware that is embedded into the hardware. One of its most important characteristics of these systems is, it gives the o/p within the time limits. Embedded systems support to make the work more perfect and convenient. So, we frequently use embedded systems in simple and complex devices too.

The applications of embedded systems mainly involve in our real life for several devices like microwave, calculators, TV remote control, home security and neighborhood traffic control systems



Embedded Processor?

- An embedded processor is a microprocessor that is used in an embedded system. An embedded system is a computer system that is designed to perform a specific task within a larger system. It is a self-contained system that is embedded within a larger device or system and is used to control the operation of the device.
- Embedded processors are typically found in devices that require real-time processing capabilities, such as industrial control systems, automotive systems, and consumer electronics.
- Embedded processors are designed to be small, low-power, and efficient, as they are typically used in devices where space and power are at a premium.
- They are also designed to be reliable and to operate for long periods of time without failure.
- Some examples of devices that use embedded processors include smartphones, TVs, washing machines, and aircraft systems.

Types of Embedded Processors

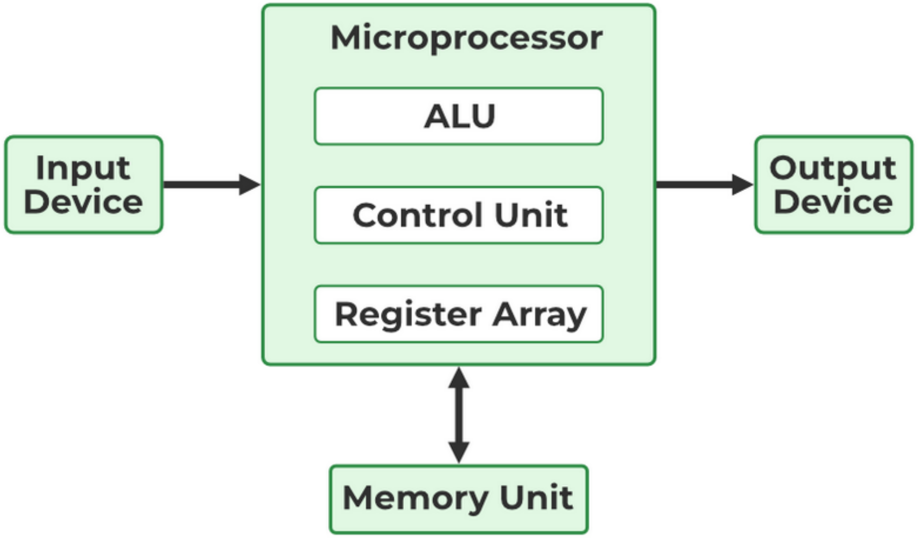
- General Purpose Processors (GPPs)
- Microprocessors
- Microcontrollers
- Digital Signal Processor (DSP)
- Single-Purpose Embedded Processor
- System-on-Chip (SoC)

Microprocessor

- A microprocessor is a processor that is contained on a microchip, or integrated circuit (IC). It is a central processing unit (CPU) that executes the instructions of a computer program. Some features of microprocessors include:
- **Instruction set:** Microprocessors have a specific instruction set that defines the operations that they can perform.
- **Clock speed:** The clock speed of a microprocessor determines how fast it can execute instructions. Microprocessors typically have high clock speeds, which allows them to perform tasks quickly.
- **Data bus:** The data bus is a communication pathway that is used to transfer data between the microprocessor and other components in a system.
- **Address bus:** The address bus is a communication pathway that is used to transfer the address of a memory location between the microprocessor and other components in a system.
- **Cache:** Many microprocessors have one or more levels of cache, which is a small amount of high-speed memory that is used to store frequently accessed data. This helps to improve the performance of the microprocessor.

Microprocessor

- **Power consumption:** Microprocessors can have relatively high power consumption, which can be a concern in devices where power is limited.
- **Size:** Microprocessors are designed to be small, as they are typically used in devices where space is at a premium.
- **Cost:** Microprocessors can vary in cost depending on their capabilities and features.



Microcontroller

A microcontroller is a small, low-power computer that is contained in a single integrated circuit (IC). It is a type of embedded processor that is used in a wide range of devices, including consumer electronics, industrial control systems, and automotive systems. Some features of microcontrollers include:

On-chip peripherals: Many microcontrollers have a variety of on-chip peripherals, such as timers, serial ports, and analog-to-digital converters, which allow them to interface with external devices.

Memory: Microcontrollers have both program memory, which stores the instructions that are executed by the processor, and data memory, which is used to store variables and other data.

Input/output (I/O) pins: Microcontrollers have a set of I/O pins that can be used to interface with external devices, such as sensors or actuators.

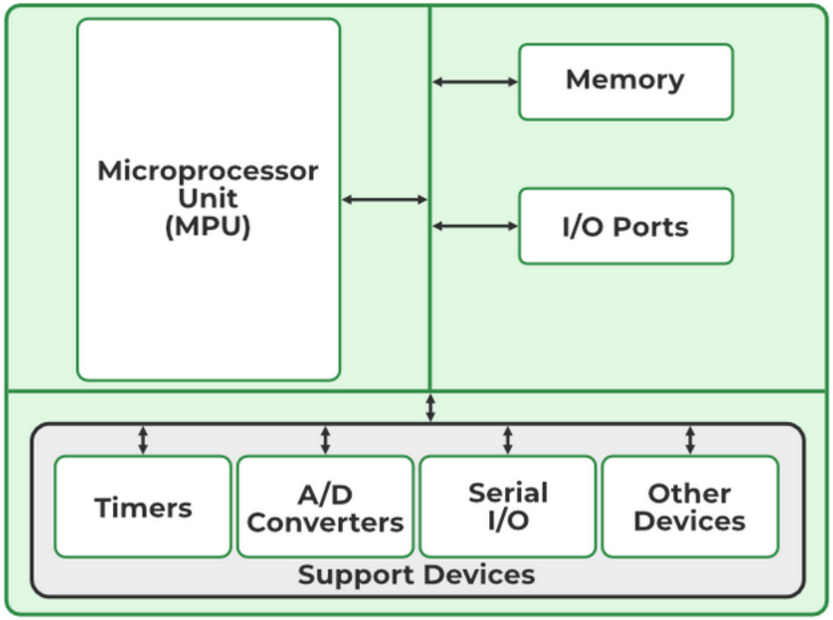
Low power consumption: Microcontrollers are designed to be low-power, which makes them suitable for use in battery-powered devices.

Microcontroller

Cost: Microcontrollers are typically less expensive than general-purpose processors, as they are designed for specific tasks and do not have as many capabilities.

Size: Microcontrollers are small, which makes them suitable for use in compact devices.

Flexibility: Microcontrollers are highly flexible and can be programmed to perform a wide range of tasks.



Microprocessor vs Microcontroller

CHITKARA UNIVERSITY

Features	Microprocessor	Microcontroller
Definition	A microprocessor is a central processing unit (CPU) that performs the majority of the processing in a computer or other device.	A microcontroller is a small computer that is integrated into a single chip and is designed to perform a specific task or set of tasks.
Clock speed	A microprocessor typically has a higher clock speed and more processing power than a microcontroller.	A microcontroller typically has a lower clock speed and more processing power than a microprocessor.
Memory requirement	A microprocessor typically requires external memory and other components to function.	A microcontroller has memory and other peripherals integrated into the same chip.
Programming language	A microprocessor is usually programmed using a high-level programming language.	A microcontroller is often programmed using a low-level language or assembly code.
Usage	• A microprocessor is generally used for tasks that require more processing power, such as running an operating system or performing complex calculations. • A microprocessor is typically used in devices that require frequent updates or upgrades, such as desktop computers and laptops. • A microprocessor is used in general-purpose computers and devices.	• A microcontroller is typically used for tasks that require more control over hardware, such as controlling a motor or reading sensors. • A microcontroller is used in devices that are designed to perform a specific task and are not often updated or upgraded, such as appliances and industrial control systems. • A microcontroller is used in specialized devices and systems that require more control over the hardware.
Examples	Examples of microprocessors include the Intel Core series of processors used in desktop computers and laptops and the Qualcomm Snapdragon processors used in smartphones.	Examples of microcontrollers include the Arduino Uno, which is often used in DIY electronics projects, and the PIC microcontrollers used in a variety of applications, including industrial control systems and consumer devices.

Characteristics of Embedded System

Generally, an embedded system executes a particular operation and does the similar work continually. For instance: A pager is constantly functioning as a pager.

- Embedded system must posses the design metric in terms of execution features like size, power, cost and also performance.
- It must perform fast enough and consume less power to increase battery life.
- Several embedded systems should constantly react to changes in the system and also calculate particular results in real time without any delay. For instance, a car cruise controller; it continuously displays and responds to speed & brake sensors. It must calculate acceleration/de-accelerations frequently in a limited time; a delayed computation can consequence in letdown to control the car.

Characteristics of Embedded System

- It must be based on a microcontroller or microprocessor based.
- It must require a memory, as its software generally inserts in ROM. It does not require any secondary memories in the PC.
- It must need connected peripherals to attach input & output devices.
- An Embedded system is inbuilt with hardware and software where the hardware is used for security and performance and Software is used for more flexibility and features.

Applications of Embedded systems

Consumer electronics: Embedded systems are used in many types of consumer electronics, such as smartphones, tablets, laptops, and other portable devices.

Industrial control systems: Embedded systems are used to control and monitoring industrial equipment, such as robots, conveyor belts, and other manufacturing systems.

Automotive systems: Embedded systems are used in automotive systems, such as engine control, traction control, and advanced driver assistance systems (ADAS).

Medical devices: Embedded systems are used in medical devices, such as monitoring devices, imaging devices, and drug delivery systems.

Applications of Embedded systems

Military systems: Embedded systems are used in military systems, such as missiles, aircraft, and other defense systems.

Communication systems: Embedded systems are used in communication systems, such as routers, switches, and other networking equipment.

Home automation systems: Embedded systems are used in home automation systems, such as smart thermostats, security systems, and lighting control systems.

Other applications: Embedded systems are used in many other applications, such as point-of-sale systems, cash registers, vending machines, and many others.

Introduction to Von Neumann Architecture

Von Neumann Architecture is a digital computer architecture whose design is based on the concept of stored program computers where program data and instruction data are stored in the same memory. This architecture was designed by the famous mathematician and physicist **John Von Neumann** in 1945.

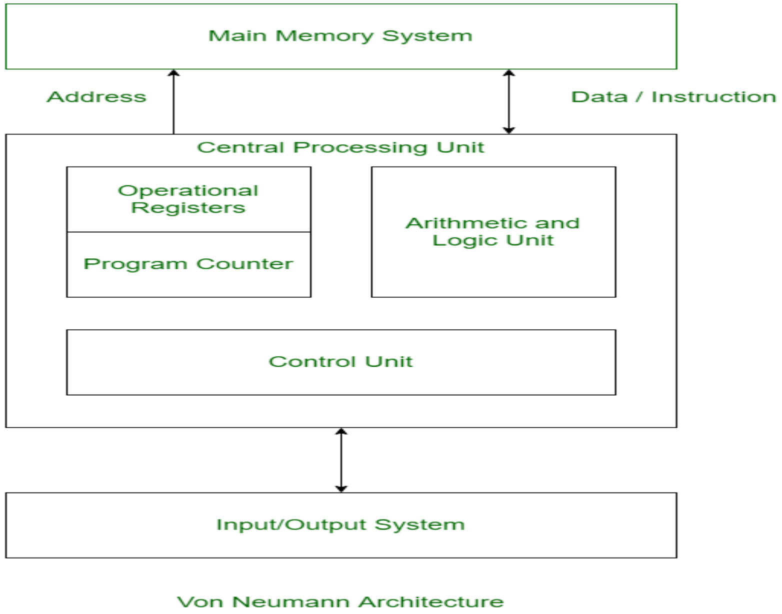


Fig. 1. Block diagram of Von Neumann Architecture

Introduction to Harvard Architecture

Harvard Architecture is the digital computer architecture whose design is based on the concept where there are separate storage and separate buses (signal path) for instruction and data. It was basically developed to overcome the bottleneck of Von Neumann Architecture.

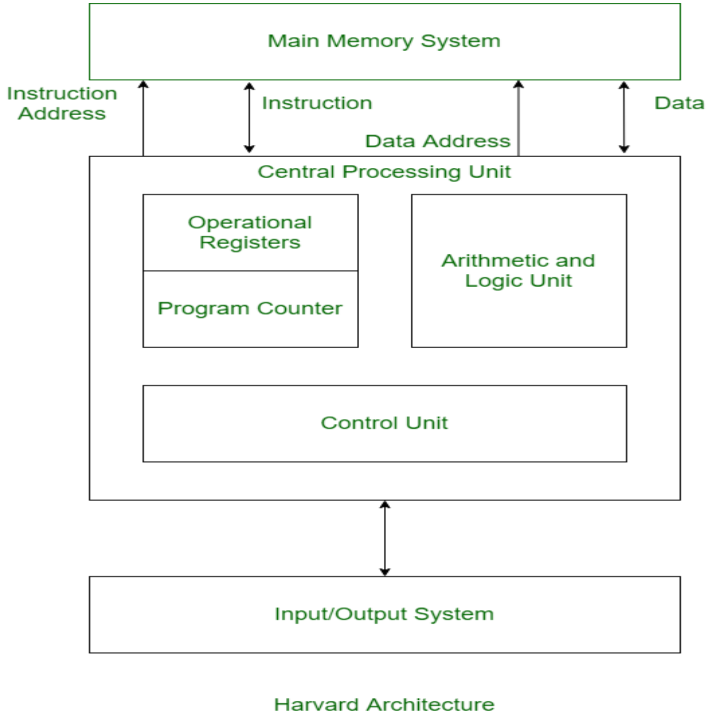


Fig. 2. Block diagram of Harvard Architecture

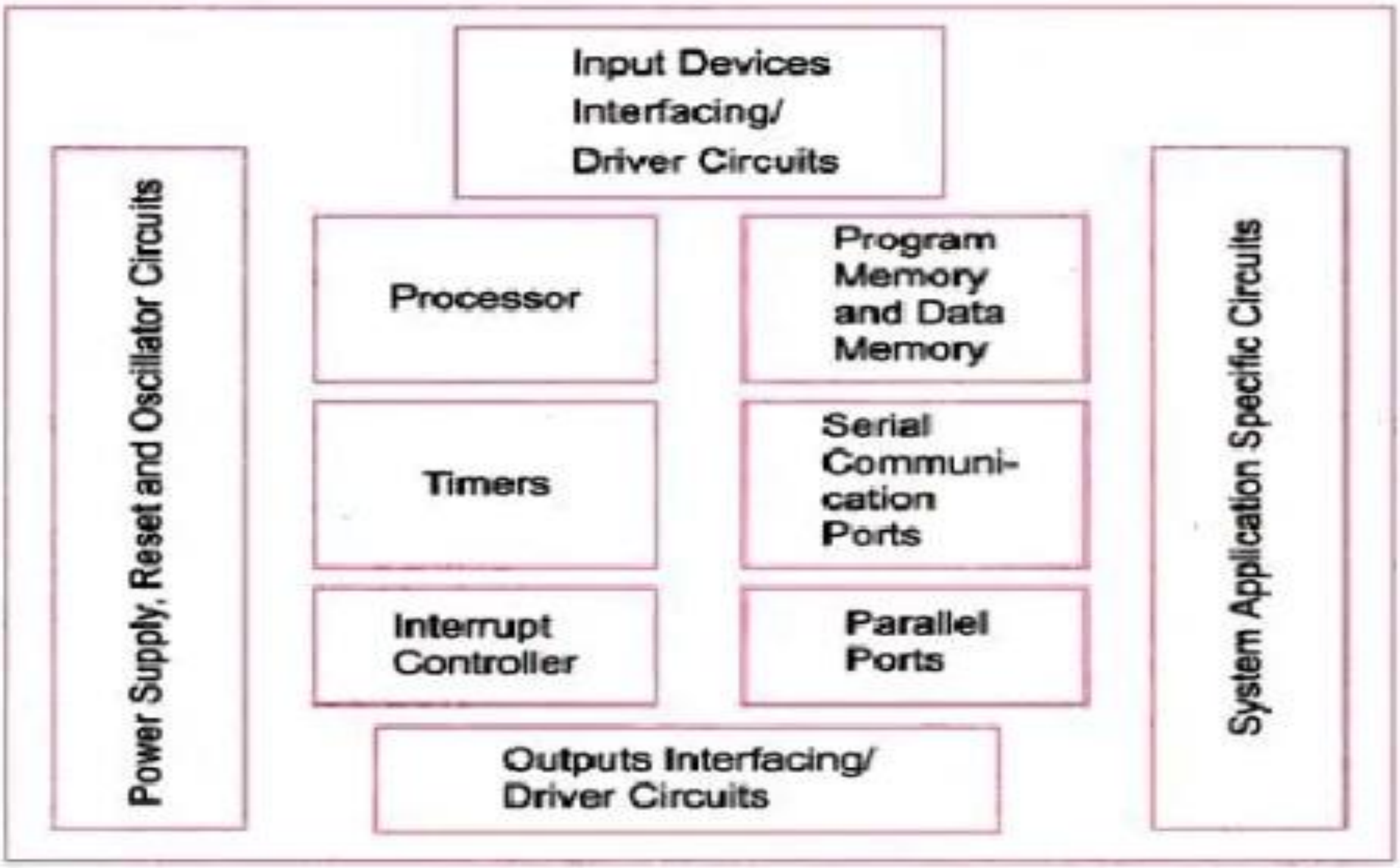
Difference Between Harvard an Von Neumann Architecture

Von Neumann Architecture	Harvard Architecture
It is ancient computer architecture based on stored program computer concept.	It is modern computer architecture based on Harvard Mark I relay based model.
Same physical memory address is used for instructions and data.	Separate physical memory address is used for instructions and data.
There is common bus for data and instruction transfer.	Separate buses are used for transferring data and instruction.
Two clock cycles are required to execute single instruction.	An instruction is executed in a single cycle.
It is cheaper in cost.	It is costly than Von Neumann Architecture.
CPU can not access instructions and read/write at the same time.	CPU can access instructions and read/write at the same time.
It is used in personal computers and small computers.	It is used in micro controllers and signal processing.

Embedded Systems Architecture

The architecture of an embedded system is an abstraction of the embedded device, meaning that it is a generalization of the system that typically doesn't show detailed implementation information such as software source code or hardware circuit design. At the architectural level, the hardware and software components in an embedded system are instead represented as some composition of interacting elements. Elements are representations of hardware and/or software whose implementation details have been abstracted out, leaving only behavioral and inter-relationship information. Architectural elements can be internally integrated within the embedded device, or exist externally to the embedded system and interact with internal elements. In short, an embedded architecture includes elements of the embedded system, elements interacting with an embedded system, the properties of each of the individual elements, and the interactive relationships between the elements.

Block Diagram of Embedded Systems Architecture

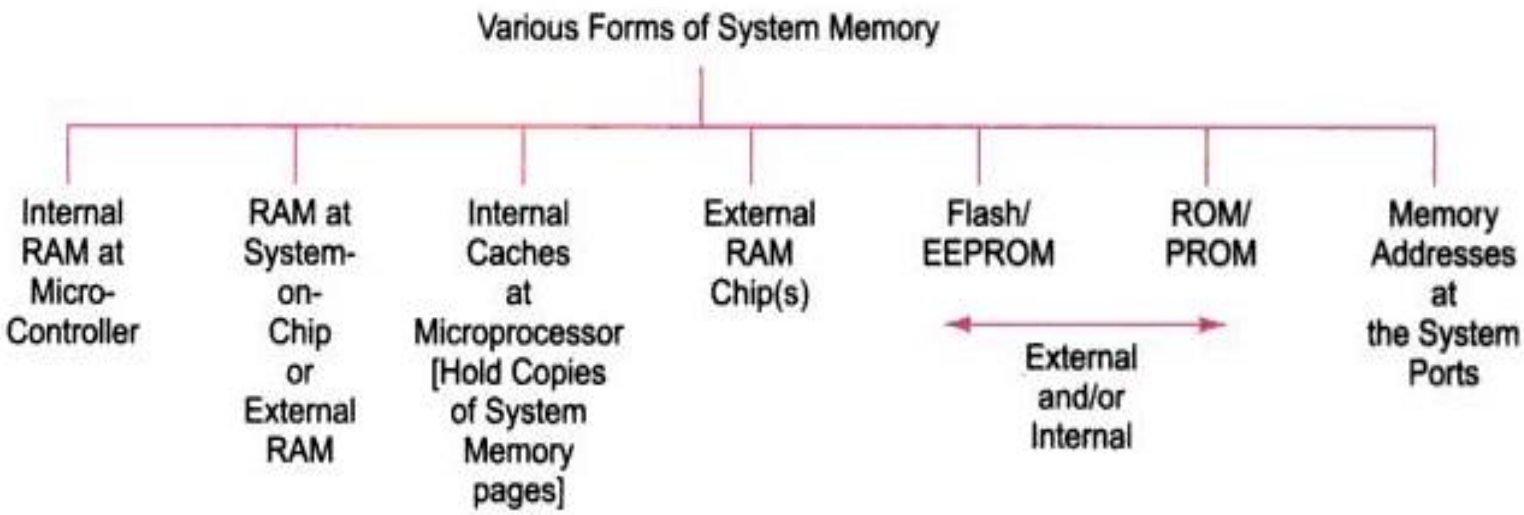


Embedded Hardware Units and Devices in a System

1. **Power Source:** Most systems have a power supply of their own. The Network Interface Card (NIC) and Graphic Accelerator are examples of embedded systems that do not have their own power supply and connect to PC power supply lines. The supply has a specific operation range or range of voltages. Various unis in an embedded system operate in one of the following four power ranges: 5.0V ± 0.25V, 3.3V ± 0.3V, 2.0V ± 0.2V and 1.5V ± 0.2V. There is generally an inverse relationship between propagation delay in the gates and operational voltage. Therefore, the 5V system processor and units are used in most high performance systems.
2. **Clock Oscillator Circuit and Clocking Units:** The clock controls the time for executing an instruction. After the power supply, the clock is the basic unit of a system. A processor needs a clock oscillator circuit. The clock controls the various clocking requirements of the CPU, of the system timers and the CPU machine cycles. The machine cycles are for fetching codes and date from memory and then decoding and executing them at the processor and for transferring the results to memory.

3. **System Timers and Real-time Clocks:** A timer circuit is suitably configured as the system clock, which ticks and generates system interrupts periodically; for example 60 times in 1s. The interrupt service routines then perform the required operation. A timer circuit is suitably configured as the real-time clock (RTC) that generates system interrupts periodically for the schedulers, real-time programs and for periodic saving of time and date in the system. The RTC or system timer is also used to obtain software-controlled delays and time-outs.
4. **Reset Circuit:** the reset circuit activates for a fixed period and then deactivates. The processor circuit keeps the reset pin active and then deactivates to let the program proceed from a default beginning address. The reset pin or the internal reset signal, if connected to the other units in the system, is activated again by the processor; it becomes an outgoing pin to enforce a reset state in other sister units of the system. On deactivation of the reset that succeeds the processor activation, a program executes from a start-up address.

5. **Watchdog Timer:** It is timing device that resets the system after a predefined timeout. It is activated within the first few clock cycles after power-up. It has number of applications. In many embedded systems reset by a watchdog timer is very essential because it helps in rescuing the system if a fault develops and the program gets stuck. On restart, the system can function normally.
6. **Memory:** In a system, there are various types of memory as shown below:



Embedded Software in a System

The software is like the brain of the embedded system.

1. **Coding of Software in Machine Codes:** During coding in this format, the programmer defines the addresses and the corresponding bytes or bits at each address. In configuring some specific physical device or subsystem, machine code-based coding is used. For example, in a transceiver, placing certain machine code and bits can configure it to transmit at specific megabytes per second or gigabytes per second, using specific bus and networking protocols.
2. **Software in High Level Language:** Since the coding in assembly language is very time consuming in most cases, software is developed in a high-level language, C or C++ or Java in most cases. C is usually the preferred language. The programmer needs to understand only the hardware organization when coding in high level language.

Application Specific Integrated Circuit (ASIC)

CHITKARA UNIVERSITY

What is an ASIC Chip (Application Specific Integrated Circuit)?

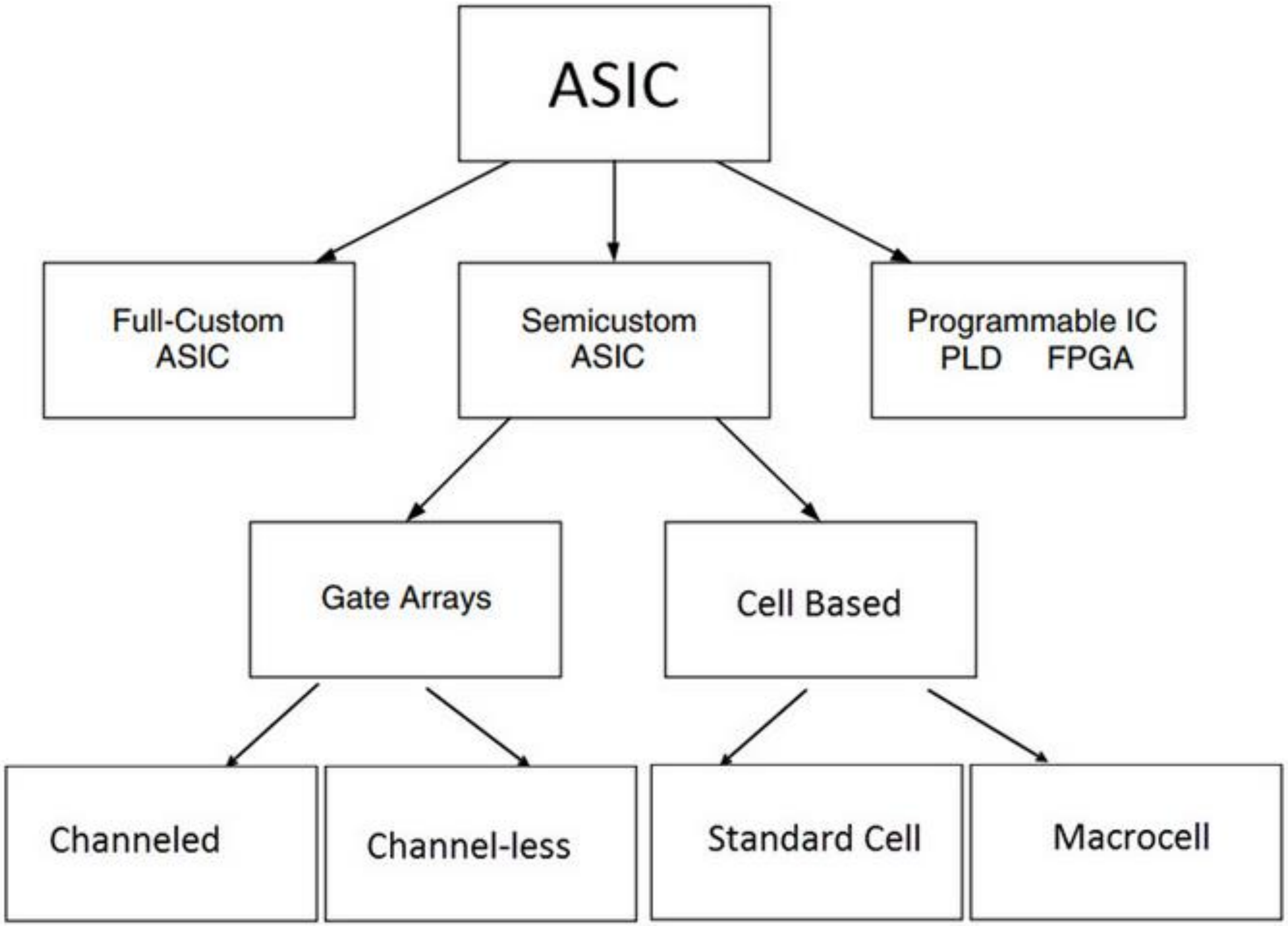
- ASICs are integrated circuits that are specifically designed and tailored for a particular application or use.

- Unlike general-purpose integrated circuits that can be used in a variety of devices, they are usually created from the ground up, based on the specific needs of the application they are intended for.

- Examples include chips found in toys or those used for the interfacing of memory and microprocessors.

Types of ASIC

•



Types of ASIC

1. Full Custom ASICs

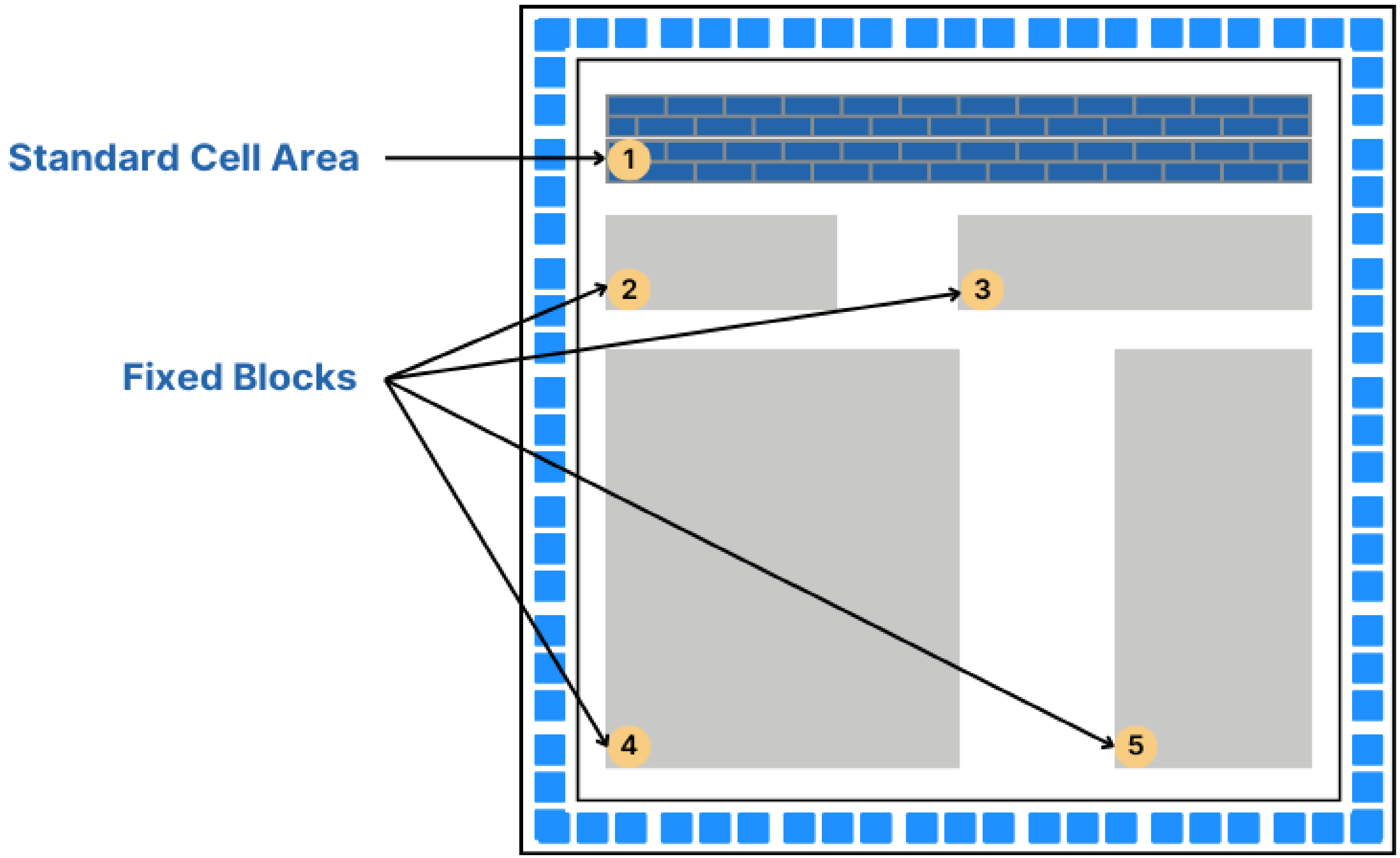
- These ASICs are designed from the ground up for a specific application. Every aspect of the chip, from the logic gates to the layout of the circuit, is custom-made to meet the precise requirements of the intended application.
- Full custom ASICs offer the highest performance and the lowest power consumption, but they are also the most expensive and time-consuming to design and manufacture.
- They are typically used in applications where the volume is high enough or the performance requirements are strict enough to justify the additional cost and effort.

2. Semi-Custom ASICs

Semi-custom ASICs, including Standard Cell ASICs and Gate Array ASICs, offer a balance between customization and cost.

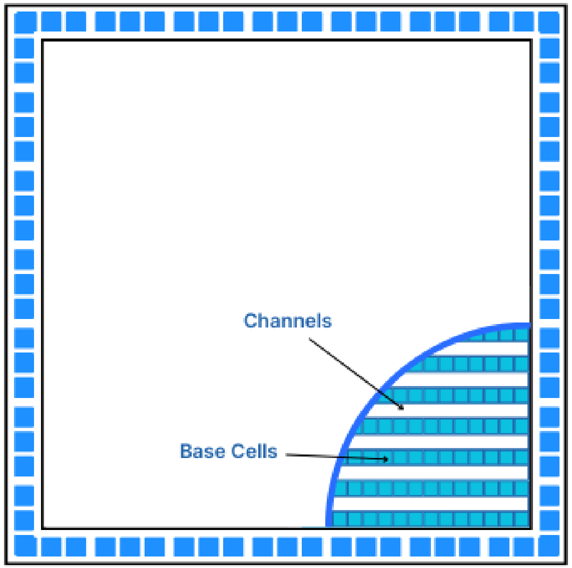
- **Standard Cell ASICs:** In standard cell-based ASIC design, a standard cell library contains pre-designed logic cells like AND gates, OR gates, multiplexers, and flip-flops.
- These cells are standardized and stored for use in ASIC chip designs.
- The ASIC chip typically includes a standard cell area or flexible block, made up of these cells arranged in rows, and may also incorporate mega cells such as microcontrollers or microprocessors, referred to as mega functions, system-level macros, or fixed blocks/functional standard blocks.

Standard Cell ASIC

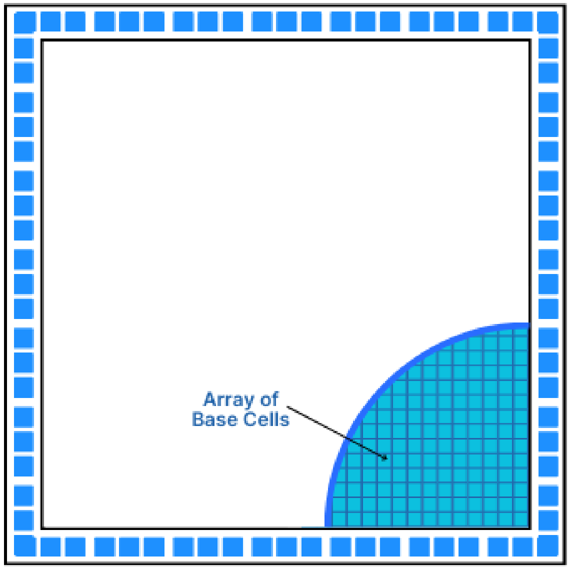


Gate Array ASIC

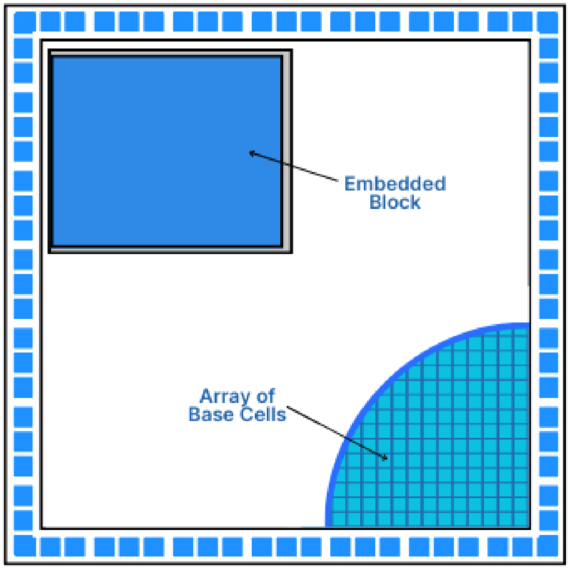
- **Gate Array ASICs:** Gate Array ASICs are a type of semi-custom ASIC with predefined transistors on the silicon wafer, where the designer cannot alter the transistor placement but can change the interconnections between them using the die's initial metal layers.
- The design utilizes a gate array library for configuration, typically resulting in **Channeled, Channel-less, or Structured Gate Arrays**, each varying in interconnection approach. This method, known as Masked Gate Array, relies on a base array pattern and base cells for circuit design.



Channeled Gate Arrays



Channel-less Gate Arrays



Structured Gate Arrays

Gate Array ASIC and Programmable ASIC

- **Channeled Gate Arrays:** Utilize predefined routing channels between logic cells for wire connections, suitable for standardized designs needing flexible interconnect pathways.
- **Channel-less Gate Arrays:** Lack predefined routing channels, allowing more compact designs by placing interconnects directly over cells, enhancing chip density.
- **Structured Gate Arrays:** Combine predefined logic blocks with customizable interconnect layers, offering a balance between design flexibility and rapid development.

3. Programmable ASICs

- **Programmable Logic Devices (PLD):** a type of digital integrated circuit that can be programmed to perform a wide range of logical operations. They are used in various applications to implement custom logic circuits without the need for custom semiconductor manufacturing.
- **FPGAs:** Reprogrammable and can be used for various applications. They can be configured by the user after manufacturing according to different needs.

Applications of ASIC

- The distinct characteristics of ASICs have transformed electronic manufacturing, leading to smaller die sizes and greater logic gate density per chip.
- Typically chosen for advanced applications, ASIC chips serve as IP cores in satellites, are crucial in ROM production, and are utilized in microcontrollers, as well as in a wide array of medical and research applications.
- Currently, one of the most notable uses of ASIC technology is in Bitcoin mining.

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, Parallel bus interface etc.

2. Product level communication interface (External Communication Interface):

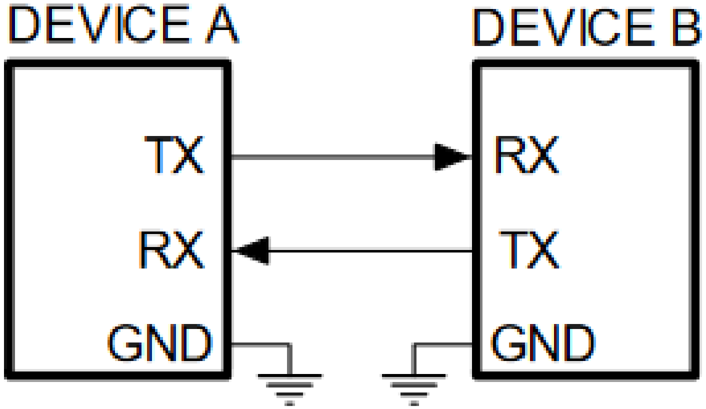
The Product level communication interface (External Communication Interface) is responsible for data transfer between the embedded system and other devices or modules. The external communication interface can be either wired media or wireless media and it can be a serial or parallel interface.

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), Parallel port etc.

Serial Communication-RS232

- A popular way to transfer commands and data between a personal computer and a microcontroller is the use of standard interface, like the one described by protocols RS232 (older) or USB (newer).
- The protocol RS232 defines the signals used in communication, and the hardware to transfer signals between devices.



Serial Communication-RS232

- An RS232 serial bus consists of just two communication wires - one for sending data and another for receiving. As such, serial devices should have two serial pins: the receiver, RX, and the transmitter, TX.

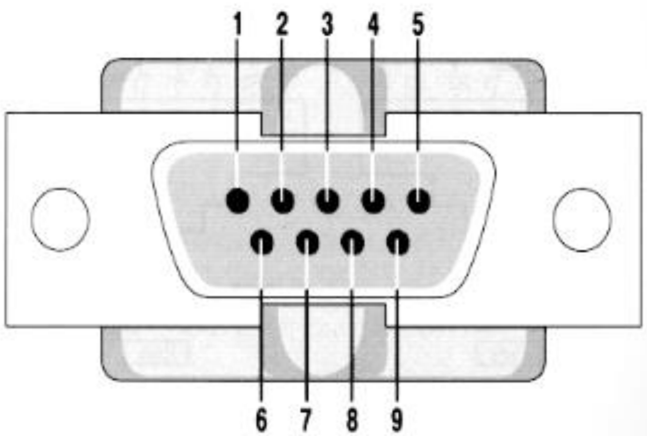
- RS232 data is sent serially, each bit is sent one after the next because there is only one data line in each direction. This mode of data transmission also requires that the receiver knows when the actual data bits are arriving so that it can synchronize itself to the incoming data.

Serial Communication-RS232

RS-232 Signals:

- Architecturally RS-232 is a bi-directional point to point link.
- Two independent channels are established for two-way (full-duplex) communications.
- RS-232 can also carry additional signals used for flow control (RTS, CTS) and modem control (DCD, DTR, DSR, RI).

(serial port - PC side)



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

How fast can RS-232 be?

• The maximum speed, according to the standard, is 20kbit/s. However, modern equipment can operate much faster than this. (i.e. Lynx can reach 115200 baud.)

• The length of the cable also plays a part in maximum speed. The longer the cable the slower the speed at which you can obtain accurate results.

• A large wire capacitance and inductance limits the maximum length of the cable and/or the maximum speed; Moreover higher is the capacitance of the cable higher is the interference between two adjacent signal wire.

Serial Communication-RS485

CHITKARA UNIVERSITY

- RS-485 is a standard interface for physical communication. It is a form of serial communication.
- Typical serial communication standards are RS232 and RS485. RS485 expands the physical function on the basis RS232 interface.
- RS-485 adopts balanced transmission and differential reception, so it has the ability to suppress common mode interference.
- In addition, the bus transceiver has high sensitivity and can detect voltages as low as 200mv, so the transmission signal can be recovered from kilometers away.
- The RS485 maximum transmission rate is 10Mbps.
- The RS485 interface is strong, and the anti-noise interference is good.

CHITKARA UNIVERSITY

Difference Between RS 485 and RS 232

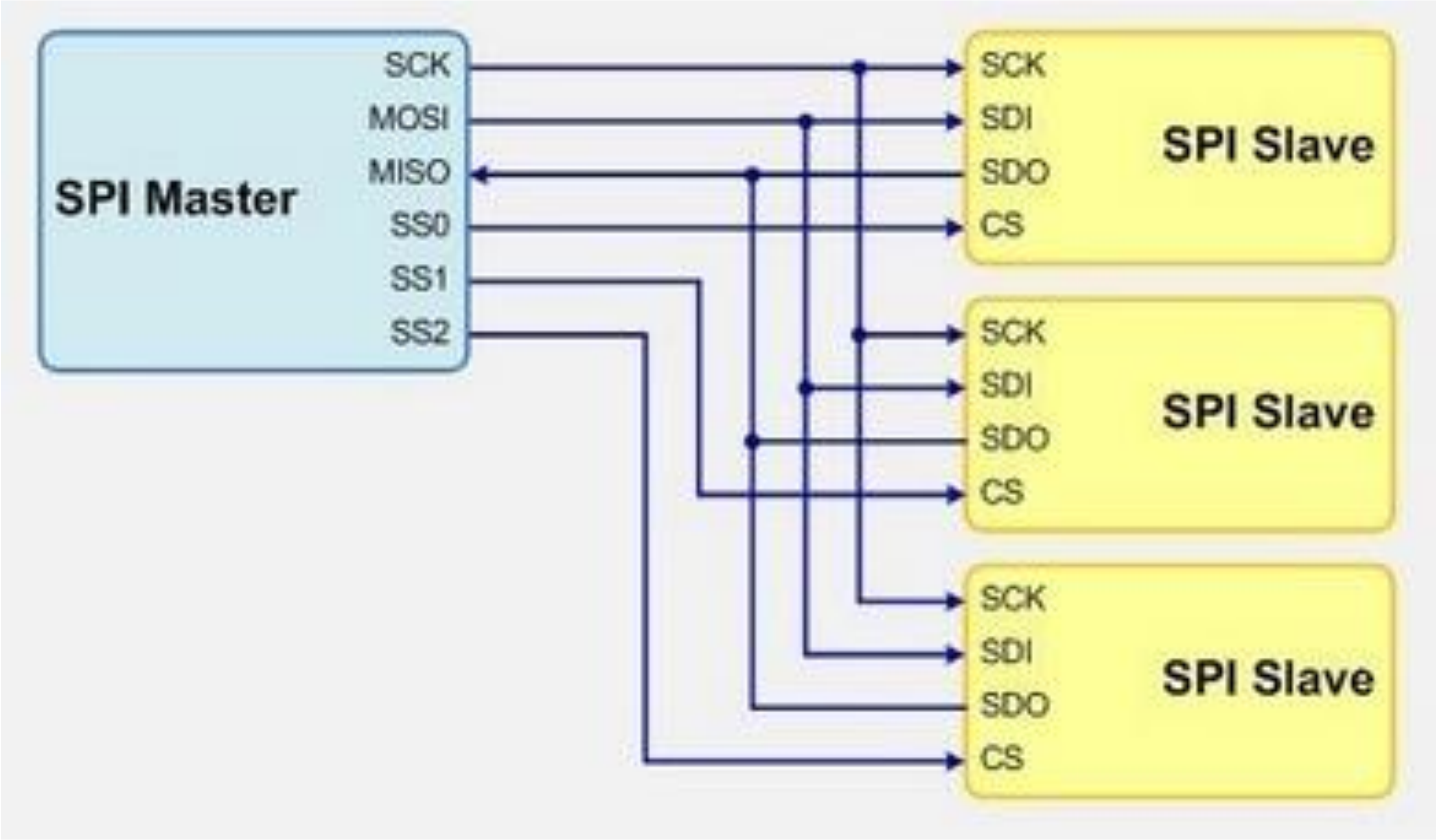
RS485	RS232
Adopt balanced transmission – differential transmission	Using unbalanced transmission – single-ended communication
The transmission distance can reach 3000 meters	The transmission distance does not exceed 20 meters
1 to 32 communication	1 to 1 communication
+2V~+6V means logic 1, -6V~-2V means logic 0	-15V-5V means logic 1, 5V~ 15V means logic 0
Two wires	Three wires
Communication rate 10MB/S	Communication rate 20KB/S

Serial Peripheral Interface (SPI)

CHITKARA UNIVERSITY

- 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 is used to send data between Microcontrollers and small peripherals such as shift registers, sensors, and SD cards.

Serial Peripheral Interface (SPI)

- 

CHITKARA UNIVERSITY

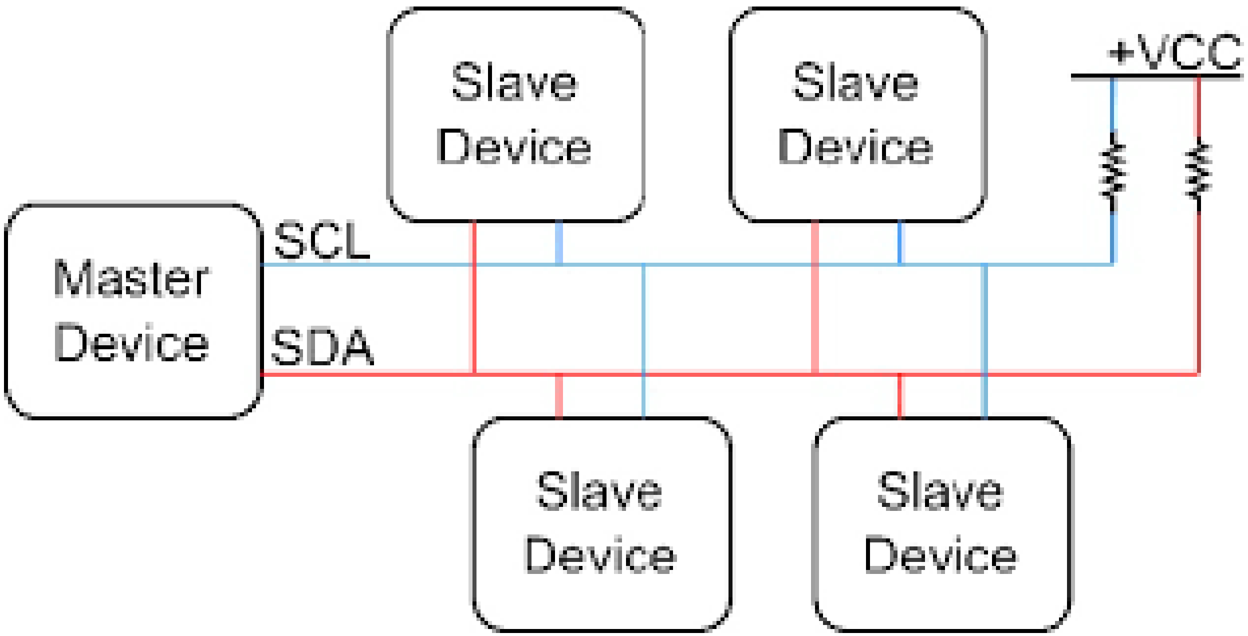
I2C:

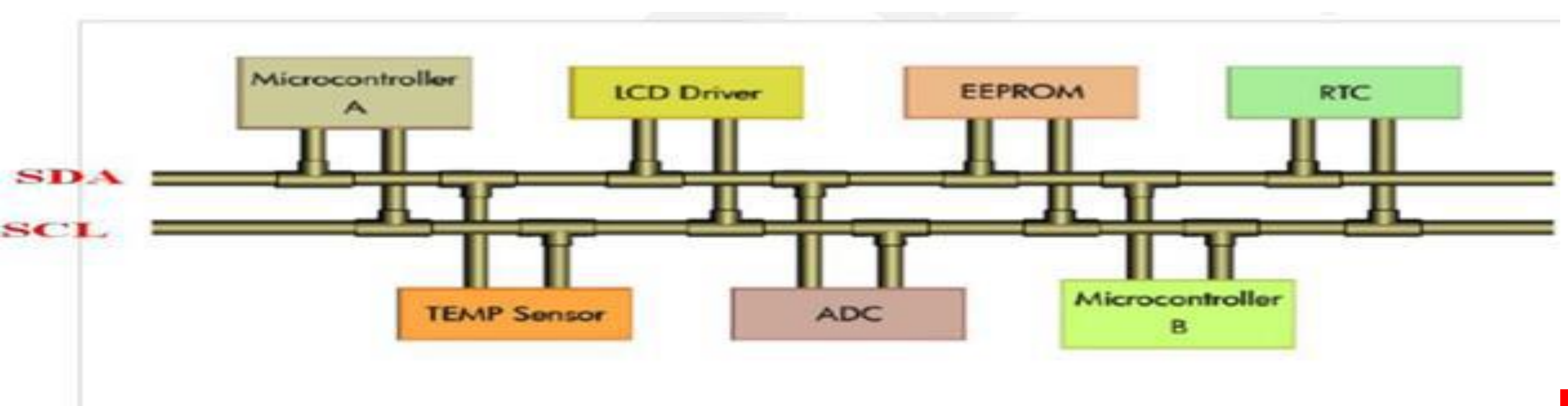
- Inter Integrated Circuit Bus (I2C - Pronounced as I square C) is a synchronous bi-directional half duplex (one-directional communication at a given point of time).
- The concept of I2C bus was developed by Philips Semiconductors in the early 1980’s.
- The original intention of I2C was to provide an 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.

I2C (Inter Integrated Circuit)

I2C Bus





I2C:

- 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.

CHITKARA UNIVERSITY

USB:

- A common interface that is used to allow communication between different peripheral devices like mouse, digital cameras, printers, keyboards, media devices, scanners, flash drives & external hard drives as well as a host controller like a smartphone or PC is known as USB protocol.

- There are different types of USB connectors available in the market where Type A and Type B are the most frequently used ones. At present, older connectors are replaced by Mini-USB, Micro-USB & USB-C cables.

Pin Configuration of USB:

- The typical Type-A USB connector is used in various applications.
- These USBs include 4 pins that are given below.
- This type of USB is observed mostly in connecting various devices to PC because it is the typical four-pin USB connector.
- This connector is taller and narrower including 4-pins arranged within a box.

Type-A USB Connector Pin Configuration

The pins of Type A USB are indicated with color wires to perform a particular function.

- **Pin1 (VBUS):** It is a red color wire, used for providing power supply.
- **Pin2 (D-):** It is a differential pair pin available in white color, used for connectivity of USB.
- **Pin3 (D+):** It is a differential pair pin available in green color, used for connectivity of USB.
- **Pin4 (GND):** It is a Ground pin, available in black color.

USB (Universal Serial Bus)

- In the above pins, both the D+ & D- pins indicate the transfer of data. When a ‘1’ is sent across the wires, then the D+ line will have positive flow, and if ‘0’ is sent then the reverse happens.

How Does The USB Protocol Work?

- The USB protocol simply works on the polling principle because, in polling, the processor continuously checks whether the input/output device is prepared for transmitting data or not. Thus, the I/O devices do not have to update the processor regarding their conditions because it is the main responsibility of the processor to check continuously. So this will make the USB low-cost & simple.

Type-A USB Connector Pin Configuration

USB (Universal Serial Bus)

USB Protocol Features

The **features of USB** include the following.

- The maximum speed of USB 2.0 is up to 480 Mbps.
- An individual USB length can reach up to 40 meters including a hub and up to five meters without a hub
- USB is a plug & play device.
- It can draw power from a computer or through its own supply.
- By using a single USB host controller, above 100 peripherals can be connected.
- The power used by a USB device is up to 5 V & delivers up to 500 mA.
- Once a computer changes into power-saving mode then some types of USBs convert automatically into sleep mode.
- A USB includes two wires; one wire is used for power & another is used for carrying the data.
- At 5V, the computer can provide power up to 500mA on the power wires.
- Low-power-based devices can draw their power from the USB directly.
- Two-way communication is possible by using a USB in between the computer & peripheral devices.

CHITKARA UNIVERSITY

USB Standards and Specifications

The **specifications of USB** will change based on USB standards that include the following.

USB supports three types of speed low speed -1.5 Mbps, Full speed -12 Mbps & High speed – 480 Mbps.

USB 2.0 Standard

- It is a high-speed USB with 480Mbps of maximum data transfer speed. This USB supports all connectors.
- The maximum length of the cable is 5 meters.
- Its max charging power is up to 15w.

USB 3.2 Standard

- USB 3.2 (Generation1) is a super speed USB with 5Gbps of maximum data transfer speed.
- It supports different connectors like USB 3 USB-A, USB 3 USB-B & USB-C.
- The maximum length of cable for this USB is 3 meters.
- Its max charging power is up to 15w.

USB Standards and Specifications

USB 3.2 (Generation2)

- USB 3.2 (Generation2) is also a super speed USB with 10Gbps of maximum data transfer speed.
- The maximum length of cable for this USB is 1meter.
- It also supports different connectors like USB 3 USB-A, USB 3 USB-B & USB-C.
- Its max charging power is up to 100w.

USB 3.2 Generation 2×2

- USB 3.2 Generation 2×2 is a super speed USB with 20Gbps of maximum data transfer speed.
- The maximum length of cable for this USB is 1meter.
- It also supports USB Connector.
- Its max charging power is up to 100w.

USB Standards and Specifications

Thunderbolt 3 Standard

- This USB is also called thunderbolt including up to 40Gbps of maximum data transfer speed.
- The maximum length of cable for this USB is 2 meters for active and 0.8meters for passive cables.
- It supports USB Connector.
- Its max charging power is up to 100w.

USB 4 Standard

- This USB is also known as Thunderbolt 4 with up to 40Gbps of maximum data transfer speed.
- The maximum length of cable for this USB is 2m for active & 0.8m for passive cables.
- It supports USB Connector.
- Its max charging power is up to 100w.

USB (Universal Serial Bus)

Advantages of USB

The **advantages of USB** include the following.

- Easy to use.
- For multiple devices, a single interface is used.
- Its size is compact.
- Its connector system is robust.
- These are not expensive.
- These are available in different sizes with different connectors.
- Auto configuration.
- Its expanding is easy.
- High speed.
- Reliable and low cost.
- Power consumption is low.
- Compatible and durable.

Disadvantages of USB

The **disadvantages of USB** include the following.

- Some manufacturers design low-quality USBs with less cost.
- Its capacity is limited.
- As compared to other systems, its data transfer is not fast.
- USB does not give the broadcasting feature, so individual messages are only communicated between the host & peripheral.

Bluetooth

- Bluetooth is used for short-range wireless voice and data communication.
- It is a Wireless Personal Area Network (WPAN) technology and is used for data communications over smaller distances.
- It operates within the unlicensed, business, scientific, and clinical (ISM) bands from 2.4 GHz to 2.485 GHz.
- Bluetooth stages up to 10 meters.
- Depending upon the version, it presents information up to at least 1 Mbps or 3 Mbps.
- The spreading method that it uses is FHSS (Frequency-hopping unfold spectrum).
- A Bluetooth network is called a piconet and a group of interconnected piconets is called a scatternet.

Bluetooth

- Bluetooth simply follows the principle of transmitting and receiving data using radio waves.
- It can be paired with the other device which has also Bluetooth but it should be within the estimated communication range to connect.
- When two devices start to share data, they form a network called piconet which can further accommodate more than five devices.

Key Features of Bluetooth

- The transmission capacity of Bluetooth is 720 kbps.
- Bluetooth is a wireless device.
- Bluetooth is a Low-cost and short-distance radio communications standard.
- Bluetooth is robust and flexible.
- The basic architecture unit of Bluetooth is a piconet.

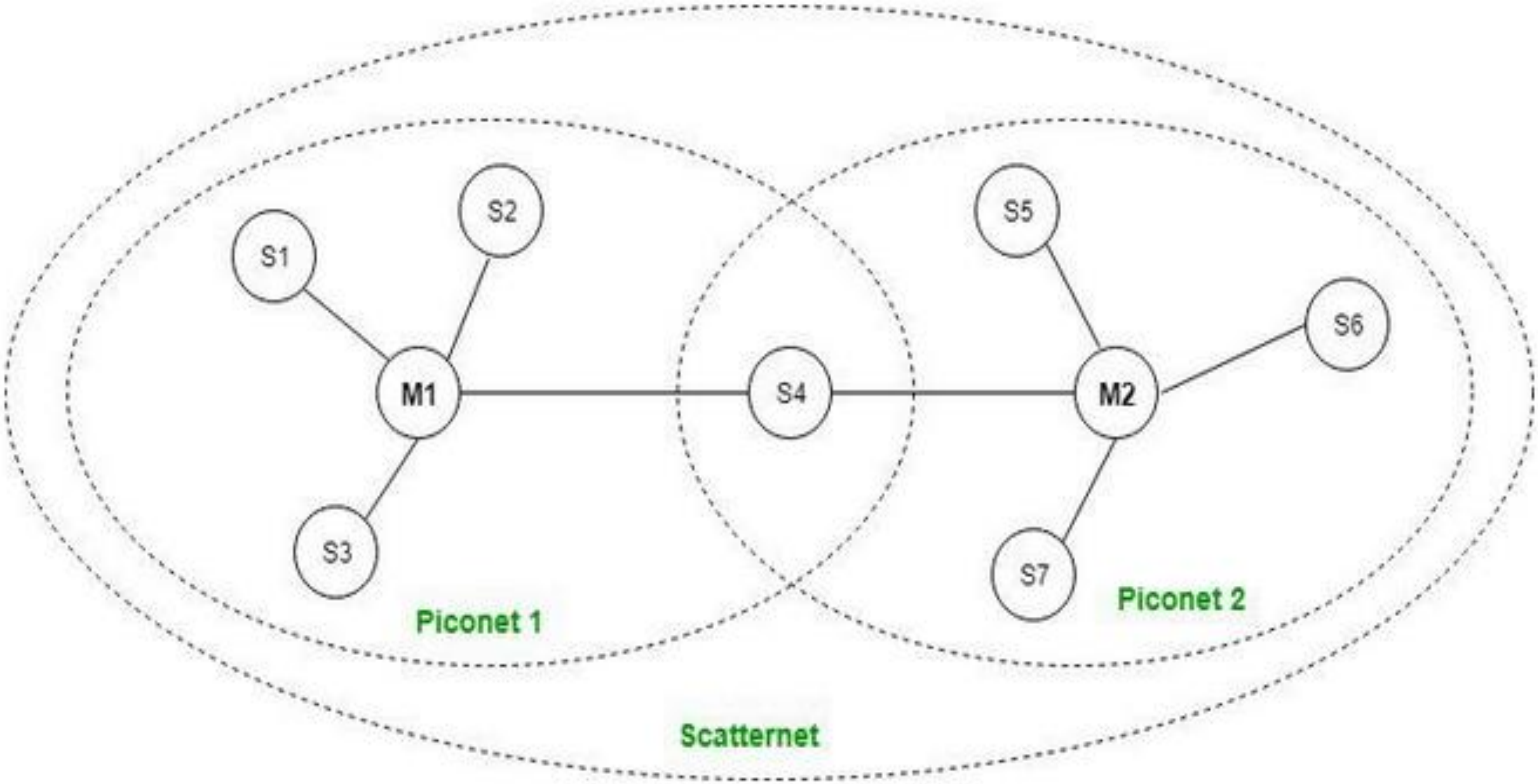
Architecture of Bluetooth

The architecture of Bluetooth defines two types of networks:

- **Piconet:** Piconet is a type of Bluetooth network that contains one primary node called the master node and seven active secondary nodes called slave nodes.
- Thus, we can say that there is a total of 8 active nodes which are present at a distance of 10 meters.
- The communication between the primary and secondary nodes can be one-to-one or one-to-many.
- Possible communication is only between the master and slave.
- Slave-slave communication is not possible.

CHITKARA UNIVERSITY

Architecture of Bluetooth



Bluetooth Architecture

Architecture of Bluetooth

Scatternet:

- It is formed by using various piconets.
- A slave that is present in one piconet can act as master or we can say primary in another piconet.
- This kind of node can receive a message from a master in one piconet and deliver the message to its slave in the other piconet where it is acting as a master.
- This type of node is referred to as a bridge node. A station cannot be mastered in two piconets.

Bluetooth

CHITKARA UNIVERSITY

Advantages of Bluetooth

- It is a low-cost and easy-to-use device.
- It can also penetrate through walls.
- It creates an Ad-hoc connection immediately without any wires.
- It is used for voice and data transfer.

Disadvantages of Bluetooth

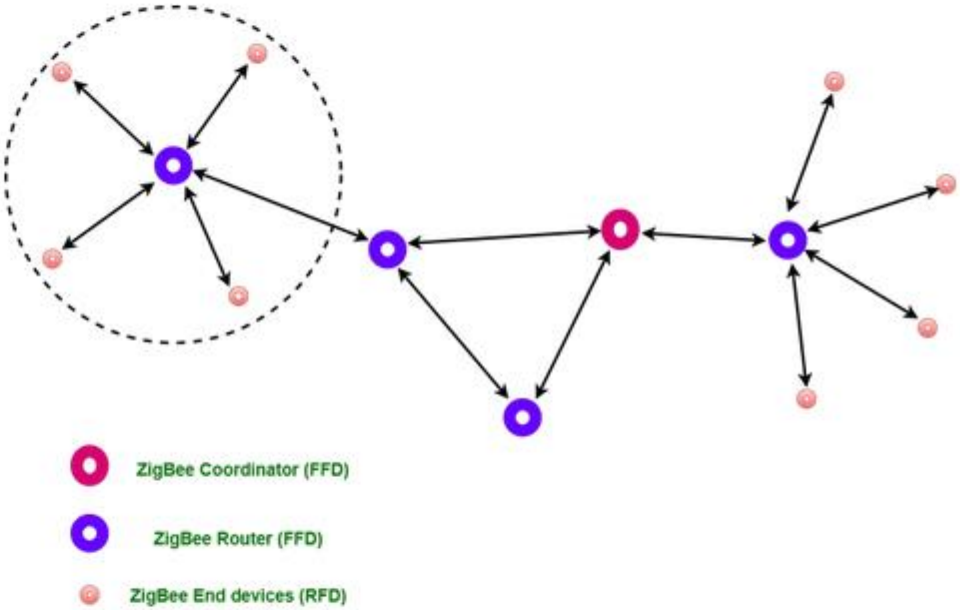
- It can be hacked and hence, less secure.
- It has a slow data transfer rate of 3 Mbps.
- Bluetooth communication does not support routing.

Zigbee

- ZigBee is a technological standard created for controlling and sensing the network.
- ZigBee is the Personal Area Network and it is based on IEEE 802.15.4 and is created by Zigbee Alliance.
- ZigBee is an open, global, packet-based protocol designed to provide an easy-to-use architecture for secure, reliable, low power wireless networks.
- ZigBee is a standard that addresses the need for very low-cost implementation of Low power devices with Low data rates for short-range wireless communications.
- IEEE 802.15.4 supports star and peer-to-peer topologies. The ZigBee specification supports star and two kinds of peer-to-peer topologies, mesh and cluster tree.
- ZigBee-compliant devices are sometimes specified as supporting point-to-point and point-to-multipoint topologies.

Types of ZigBee Devices:

- **Zigbee Coordinator Device:** It communicates with routers. This device is used for connecting the devices.
- **Zigbee Router:** It is used for passing the data between devices.
- **Zigbee End Device:** It is the device that is going to be controlled.



General Characteristics of Zigbee Standard:

- Low Power Consumption
- Low Data Rate (20- 250 kbps)
- Short-Range (75-100 meters)
- Network Join Time (~ 30 msec)
- Support Small and Large Networks (up to 65000 devices (Theory); 240 devices (Practically))
- Low Cost of Products and Cheap Implementation (Open Source Protocol)
- 3 frequency bands with 27 channels.

Operating Frequency Bands (Only one channel will be selected for use in a network):

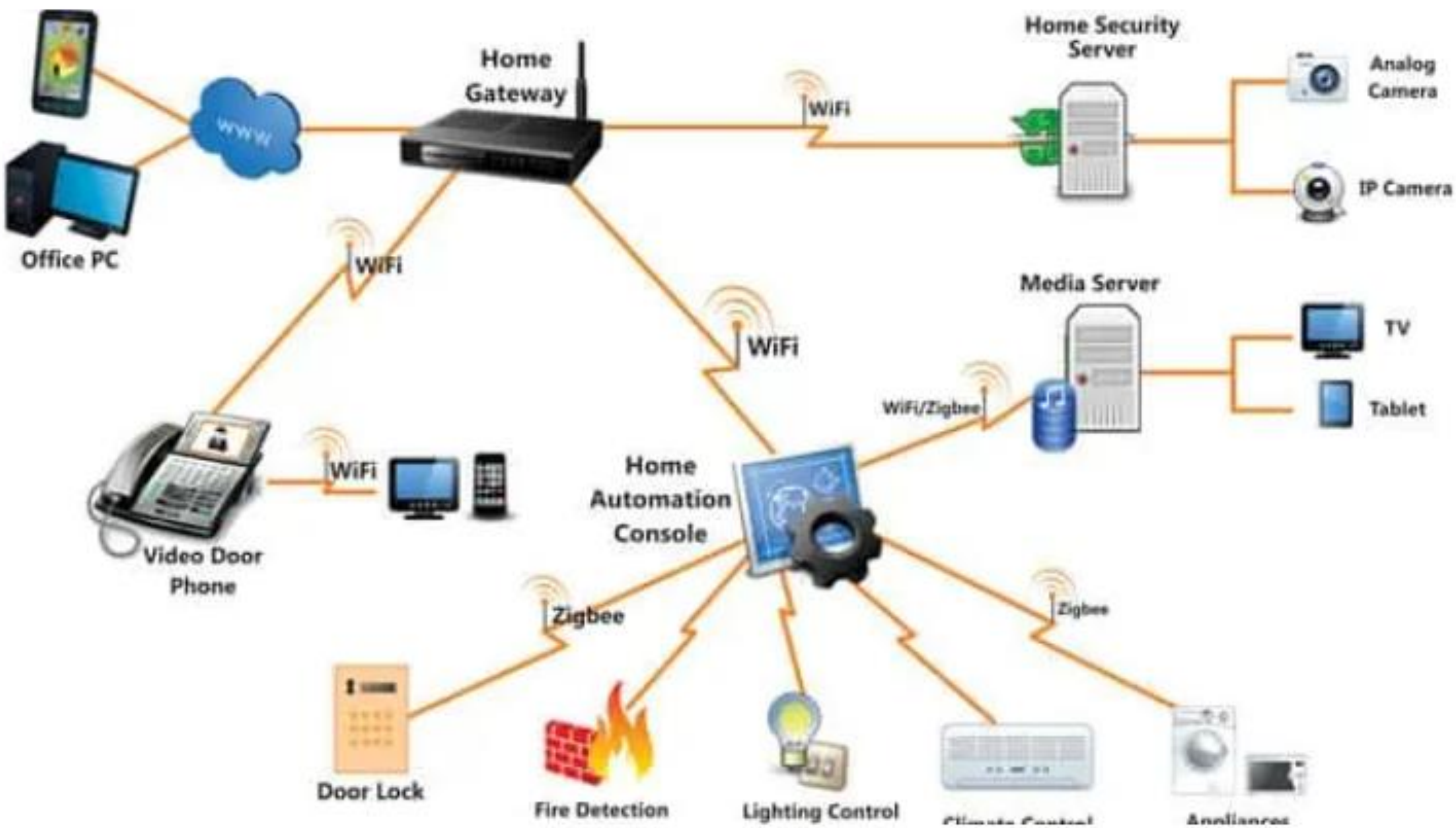
1. **Channel 0**: 868 MHz (Europe)
2. **Channel 1-10**: 915 MHz (the US and Australia)
3. **Channel 11-26**: 2.4 GHz (Across the World)

Zigbee Network Topologies:

- **Star Topology** (ZigBee Smart Energy): Consists of a coordinator and several end devices, end devices communicate only with the coordinator.
- **Mesh Topology** (Self Healing Process): Mesh topology consists of one coordinator, several routers, and end devices.
- **Tree Topology**: In this topology, the network consists of a central node which is a coordinator, several routers, and end devices. the function of the router is to extend the network coverage.

Zigbee

Zigbee Network:

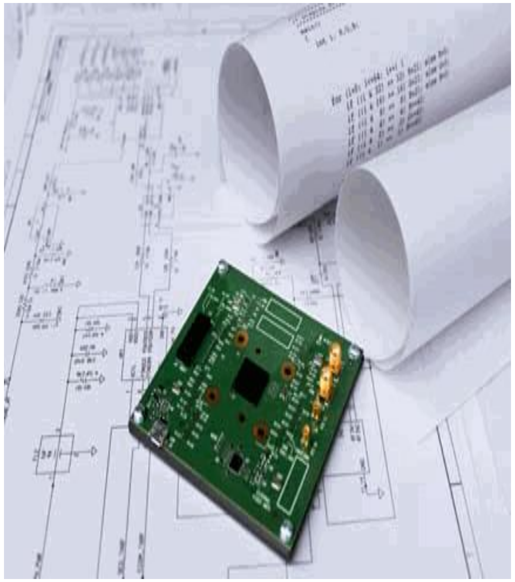


Zigbee Applications:

1. Home Automation
2. Medical Data Collection
3. Industrial Control Systems
4. meter reading system
5. light control system
6. Commercial
7. Government Markets Worldwide
8. Home Networking

Real Time Embedded Systems

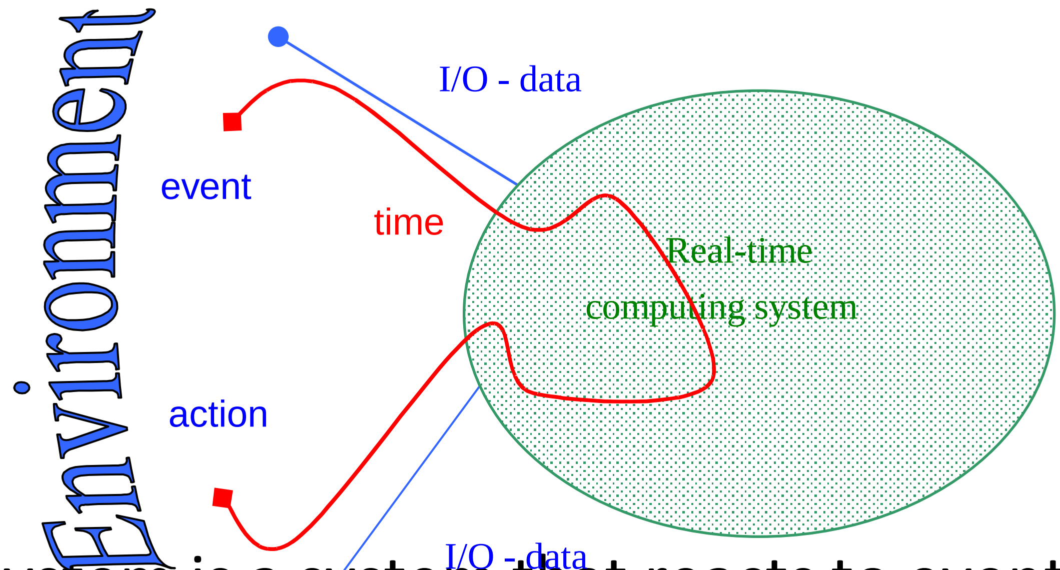
- One of the earliest decision points in embedded systems design is whether the system will require real-time computing capabilities
- From controlling elevators, stoplights, the valves of your washing machine or even how long your toaster toasts, embedded computing is everywhere.
- Real time systems are those systems that work within strict time constraints and provide a worst case time estimate for critical situations
- Think of a car's airbag deployment system. If the microcontroller doesn't detect a collision or electronically trigger the airbag within a fraction of a second, the result is tragic.
- Real time computing describes the ability to react to inputs and deliver the prescribed output within a constrained time frame. Devices that use real-time computing are deployed in applications where their correct functioning can make the difference between life and death.



Embedded Systems v/s Real Time Embedded Systems

Feature	Embedded System	Real-Time Embedded System
Definition	Specialized computing system performing dedicated functions within a larger system	Subset of embedded systems designed to perform tasks within strict timing constraints
Timing Constraints	May or may not have real-time constraints	Always have strict timing constraints
Application Sensitivity	Timing is less critical Focus on reliability and functionality	Timing is crucial Deadlines must be met to avoid unacceptable consequences
Design Complexity	Can range from simple to complex, focusing on functionality, efficiency and cost	Generally more complex due to the need to guarantee timely responses, predictability, and reliability

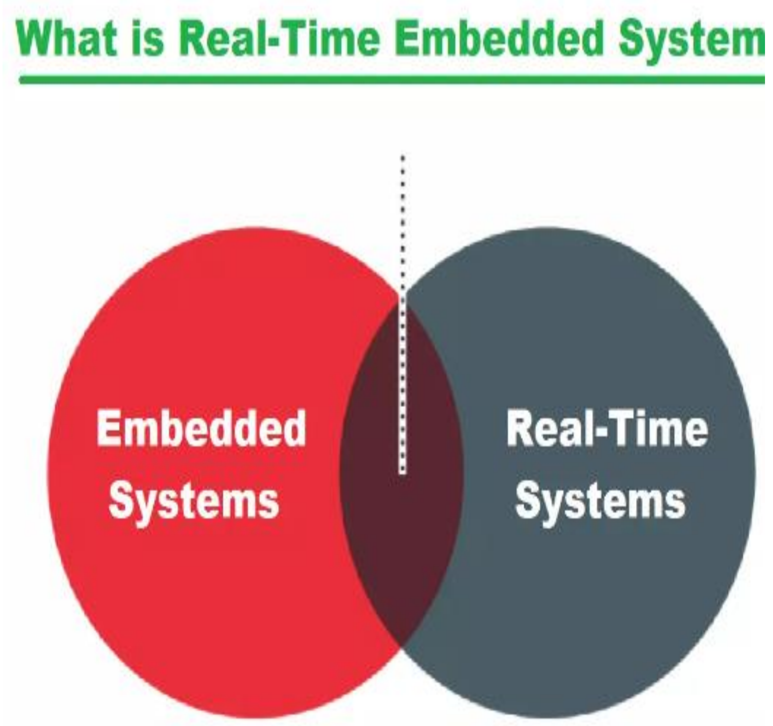
Real Time Systems



A real time system is a system that reacts to events in the environment by performing predefined actions <u>WITHIN SPECIFIED TIME INTEVALS</u>

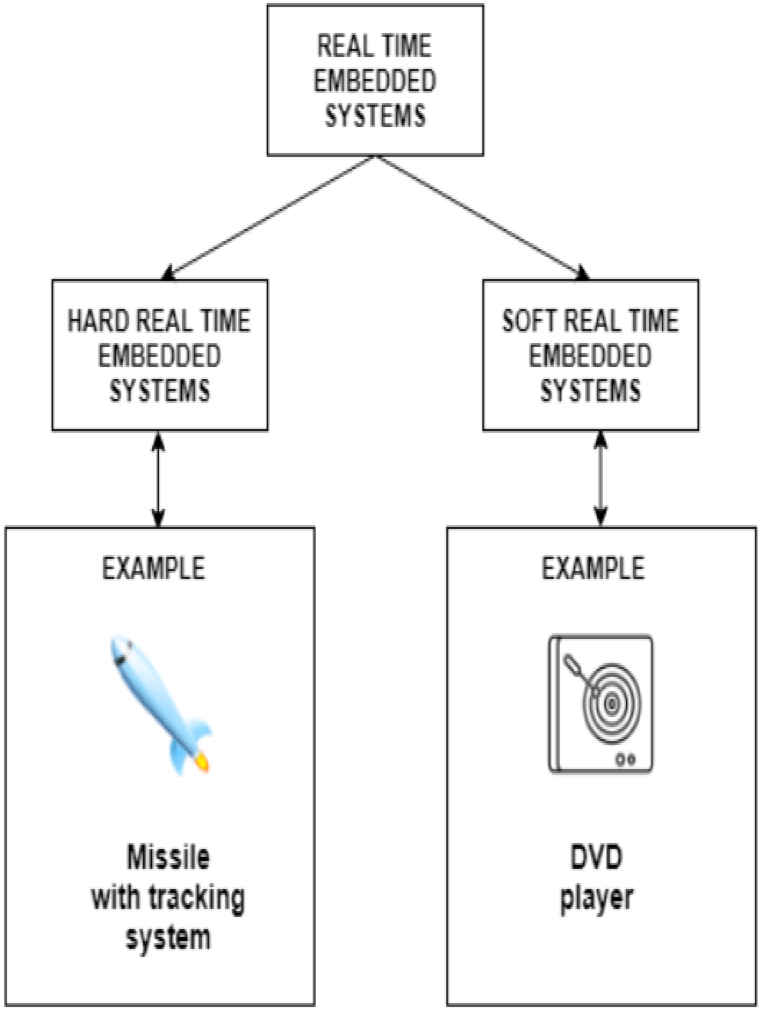
Real Time Embedded Systems

- The embedded systems which respond to real time situation with the help of its embedded software and hardware, within the specified time constraints are called real time embedded systems



Real Time Embedded Systems

- The Real time System is a system which is used for performing some specific tasks.
- The tasks assigned to real-time systems need to be completed in given time interval.
- A real-time embedded system combines the technologies of embedded systems and real-time computing. To achieve the most complete and accurate description



Real Time Operating System (RTOS)

- An RTOS is used for embedded systems applications that are time sensitive or time critical.
- In a time-critical system, the value of completing a task is linked to its timeliness and tasks that are completed past the deadline may have a negative value.
- RTOS includes a task scheduler component whose goal is to ensure that critical tasks meet their deadline, even when it means sacrificing other areas of performance.

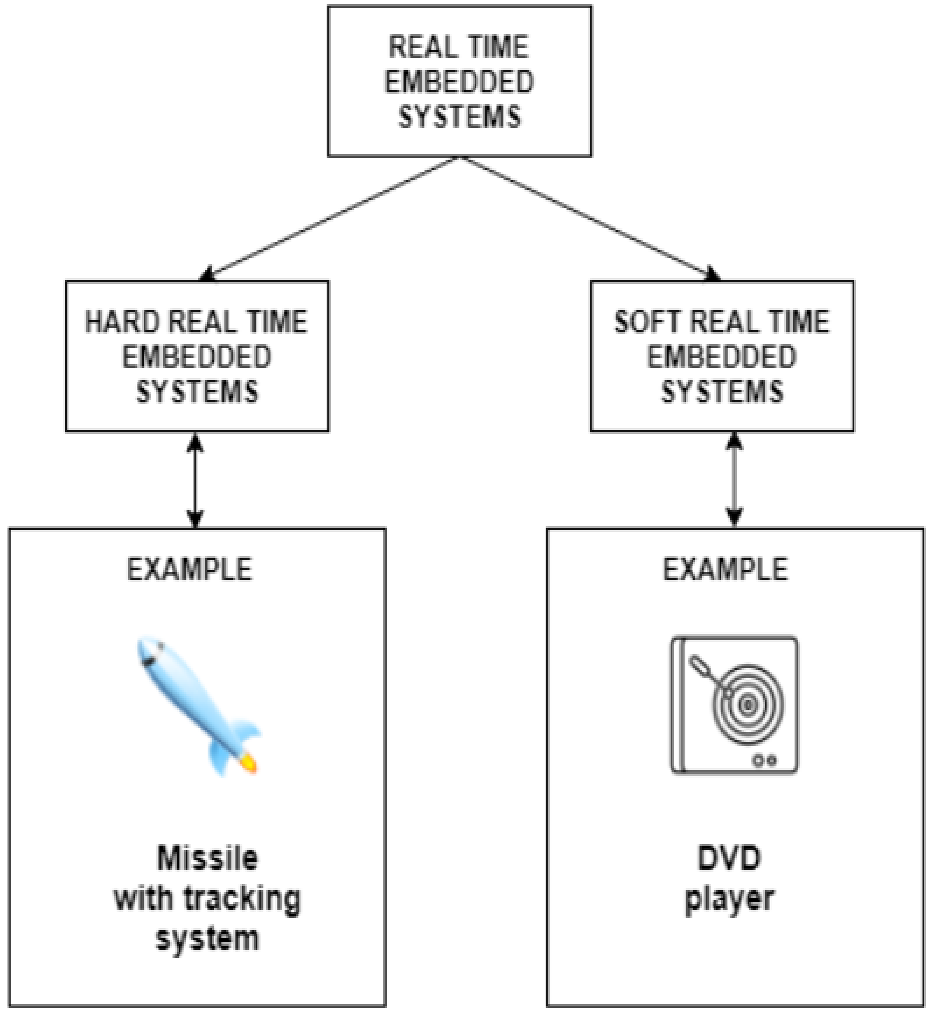
Characteristics of RTES

1. **Constant Response:** A real-time embedded system always responds in the same manner to a certain situation, it is not allowed to deviate from its normal designated output.
2. **Deadline:** A deadline is crucial to the working of an embedded system, a missed deadline can cost lives and finances
3. **Accuracy:** Accuracy is important characteristic for RTES. what would happen if the pacemaker can't maintain the heartbeat, patient would eventually die
4. **Quick Response:** The real-time embedded system must be swift enough to respond to the changing external environment with immediate effect

Examples of RTES

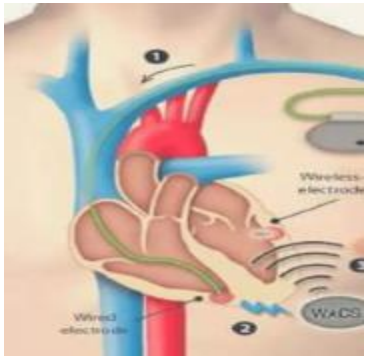
Domain	Application
Avionics	Navigation; displays
Multimedia	Games; simulators
Medicine	Robot surgery; remote surgery; medical imaging
Industrial systems	Robot assembly lines; automated inspection
Civilian	Elevator control Automotive system; Global positioning system (GPS)

Types of RTES



Hard Real Time Embedded System

- This type of system makes sure that all critical processes are completed within the given time frame.
- This means that all the delays in the system are strictly time bound.
- Also, there is little to no secondary memory and data is stored in short term memory or read only memory.
- Hard real time systems are used in various areas such as missiles, airplanes etc.



Pacemaker

Hard RTES Example

Example of Hard RTES

Hard Real Time Embedded System

Advantages:

- **Reliability and Predictability:** Ensures that critical tasks are completed within strict deadlines, which is essential for safety-critical applications.
- **Safety:** Suitable for applications where failure to meet deadlines can result in catastrophic consequences, ensuring safety and compliance with regulatory standards.
- **Deterministic Behavior:** Provides precise and deterministic system behavior, which is crucial for applications requiring high levels of precision and control.

Disadvantages:

- **Complexity and Cost:** Designing and implementing hard real-time systems is complex and costly due to stringent timing and reliability requirements.
- **Resource Utilization:** Often requires over-provisioning of resources to guarantee deadlines, leading to potentially inefficient resource utilization.
- **Inflexibility:** Lack of flexibility in handling dynamic workloads and changes in the system environment, as strict adherence to deadlines is mandatory.

Soft Real Time Embedded System

- This These are much less constrictive than hard real time systems but the basic premise is the same i.e critical processes need to be completed within the given time frame.
- However, this time frame can be a little flexible.
- Soft real time systems are used in various areas such as multimedia, scientific projects etc.

Soft RTES Example

Examples of Soft RTES

Soft Real Time Embedded System

Advantages:

- **Flexibility:** Can handle dynamic and variable workloads, making them suitable for applications where occasional deadline misses are acceptable.
- **Efficiency:** Higher resource utilization as the system can overcommit resources and manage priorities dynamically.
- **Cost-Effective:** Generally cheaper to develop and maintain compared to hard real-time systems due to more relaxed timing constraints.

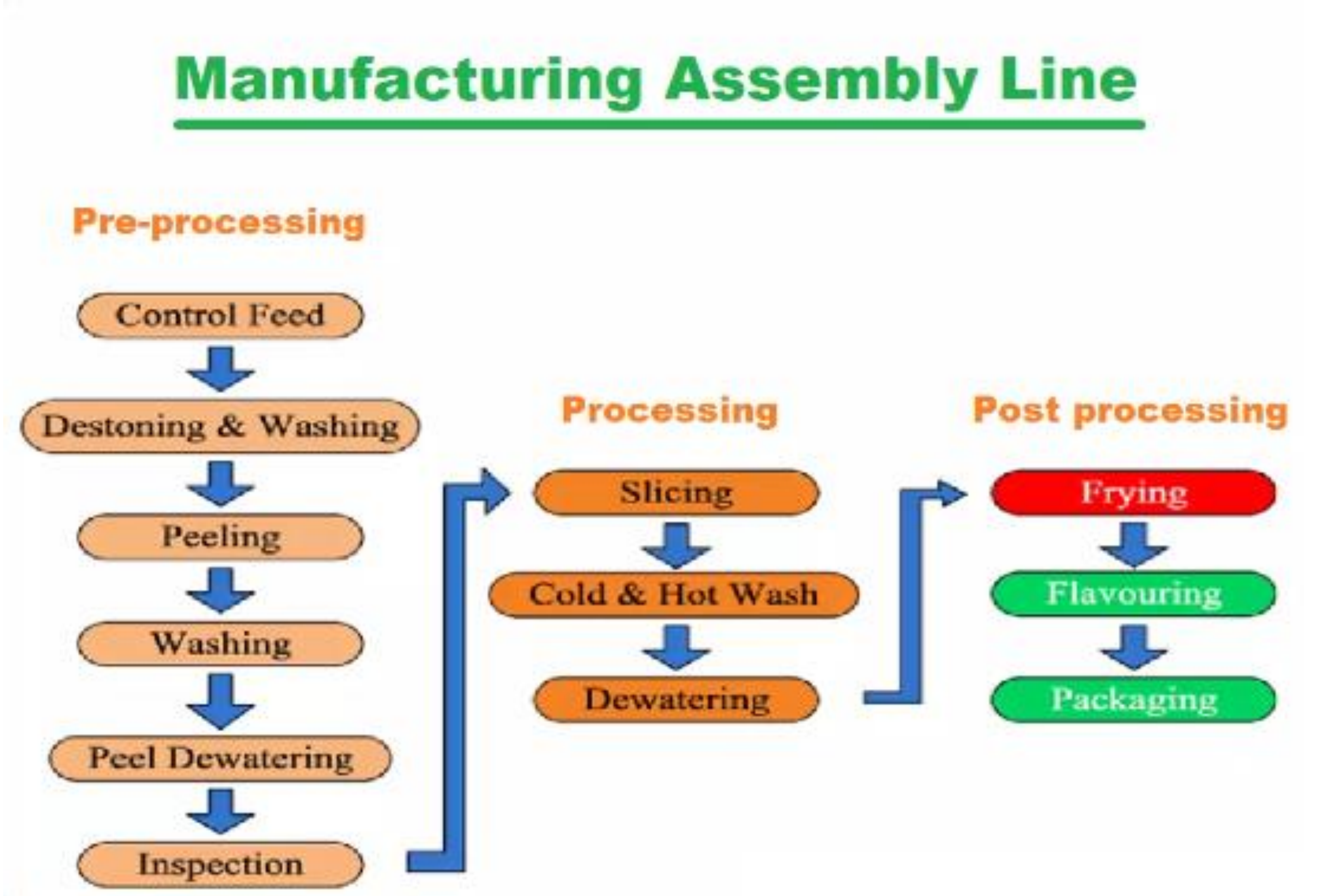
Disadvantages:

- **Unpredictability:** Performance can be less predictable, and occasional deadline misses can lead to a degradation of service quality.
- **Suitability:** Not suitable for safety-critical applications where missing deadlines can have severe consequences.
- **Quality Variability:** The quality of service may vary depending on the system load and priority management, leading to potential user dissatisfaction.

Firm Real Time Embedded System

- In a firm RTES, errors are occasionally permissible but there is an understanding that missed deadlines result in degraded performance of the device.
- A device using a firm RTES may occasionally miss a deadline, but the application can recover as long as failures are relatively infrequent.

Firm RTES Example



A fully automated assembly line doesn't crash when a task isn't performed in time, it rather ignores that missed part and continues to complete the rest.

Firm Real Time Embedded System

Advantages:

- **Balanced Approach:** Combines the reliability of hard real-time systems with the flexibility of soft real-time systems, offering a balanced solution for many applications.
- **Tolerance for Occasional Misses:** Can tolerate occasional deadline misses without catastrophic consequences, making them suitable for a wide range of applications.
- **Resource Optimization:** Can achieve a good balance between resource utilization and deadline adherence, optimizing overall system performance.

Disadvantages:

- **Complexity:** Balancing strict deadlines with occasional flexibility adds complexity to system design and implementation.
- **Resource Management:** Requires careful management of resources to ensure that the system remains reliable while allowing some flexibility.
- **Potential for Degradation:** Repeated or excessive deadline misses can still lead to significant performance degradation or system errors, requiring robust handling mechanisms.

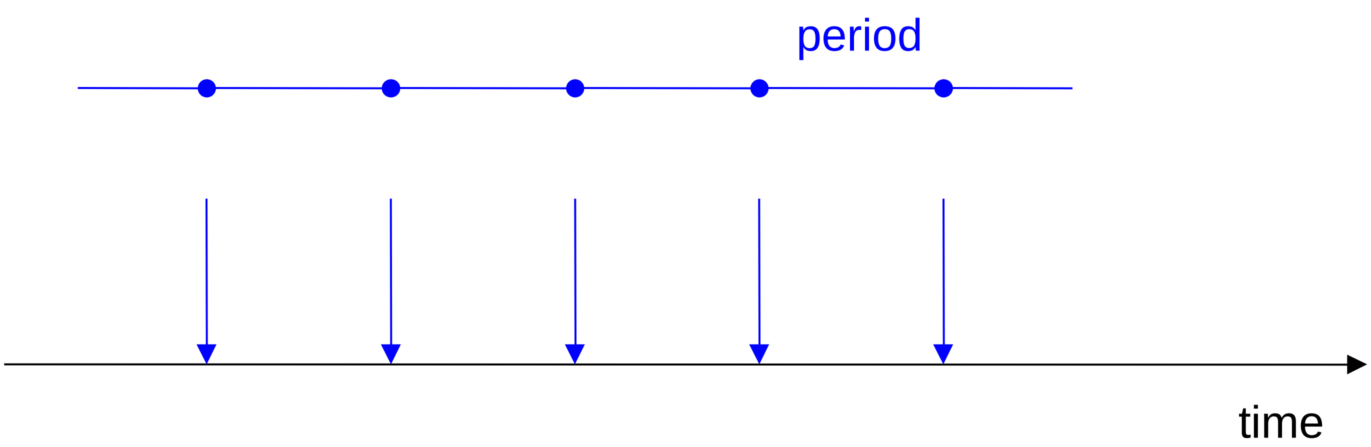
Hard v/s Soft v/s Firm

Feature	Hard Real-Time Embedded Systems	Soft Real-Time Embedded Systems	Firm Real-Time Embedded Systems
Definition	Systems where missing a deadline can lead to catastrophic failures.	Systems where missing a deadline can degrade performance but is not catastrophic.	Systems where missing a deadline occasionally is tolerable but repeated misses are problematic.
Examples	Airbag systems, pacemakers, industrial control systems	Multimedia systems, online transaction processing	Automated teller machines (ATM), navigation systems
Deadline Importance	Strict and non-negotiable	Flexible, negotiable within limits	Generally strict, but some deadlines can be missed occasionally
Consequences of Missing Deadlines	Catastrophic failures, possible loss of life or major system failures	Reduced quality of service, performance degradation	Degradation in quality, potential for system errors if repeated misses occur
System Predictability	Highly predictable and deterministic	Less predictable, can tolerate some variability	Predictable with some tolerance for infrequent deadline misses
Scheduling Approach	Priority-based or time-triggered	Best-effort or dynamic priority-based	Mixture of hard and soft real-time scheduling
System Design Complexity	High, due to need for strict timing guarantees and predictability	Moderate, balancing performance and deadline adherence	Moderate to high, depending on tolerance for deadline misses
Typical Use Cases	Critical systems where safety and reliability are paramount	Consumer electronics, telecommunication systems	Financial systems, some mission-critical applications with tolerance for occasional misses

- Rate Monotonic Scheduling (RMS): This algorithm assigns static priorities to tasks based on their periods, with shorter periods receiving higher priorities. It is suitable for periodic task sets and provides a simple and efficient scheduling mechanism.
- Earliest Deadline First (EDF): In this algorithm, tasks are scheduled based on their absolute deadlines, with the task having the earliest deadline receiving the highest priority. EDF is an optimal scheduling algorithm for preemptive task sets and can achieve higher processor utilization than RMS.
- Least Laxity First (LLF): This algorithm schedules tasks based on their laxity, which is the amount of time remaining before a task's deadline minus its remaining execution time. Tasks with lower laxity values are given higher priorities, as they have less flexibility in meeting their deadlines.
- Priority-based Scheduling: In this approach, tasks are assigned static or dynamic priorities based on various factors, such as importance, criticality, or deadline requirements. Higher-priority tasks are scheduled before lower-priority tasks.

Types of Event to trigger RTES

- Periodic
  - activity occurs repeatedly
  - e.g., to monitor environment values, temperature, etc.



periodic

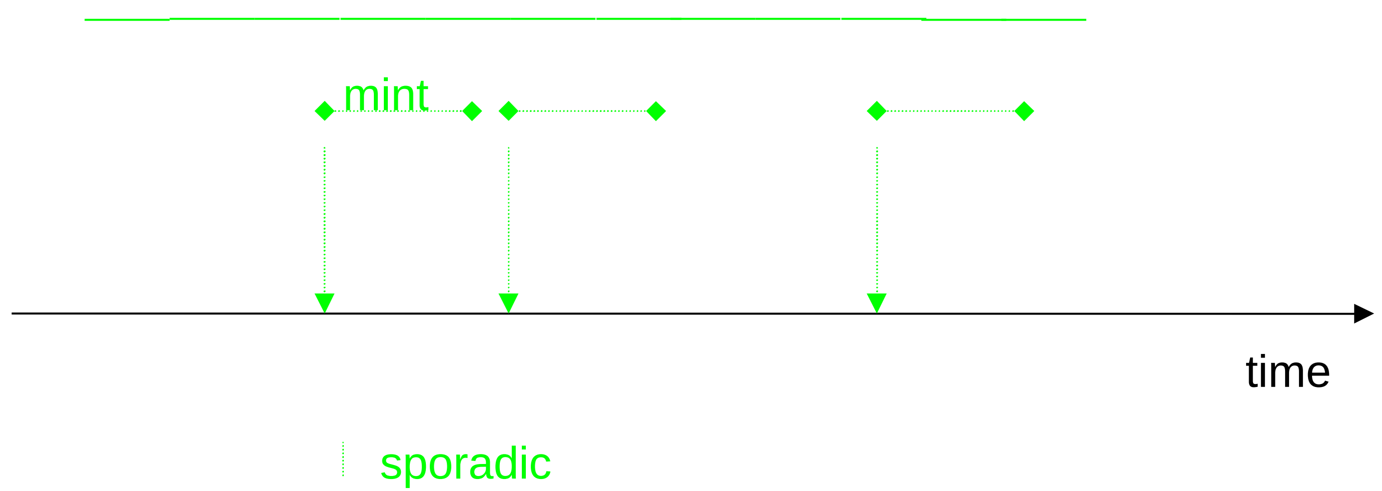
Types of Event to trigger RTES

- Aperiodic
  - can occur any time
  - no arrival pattern given

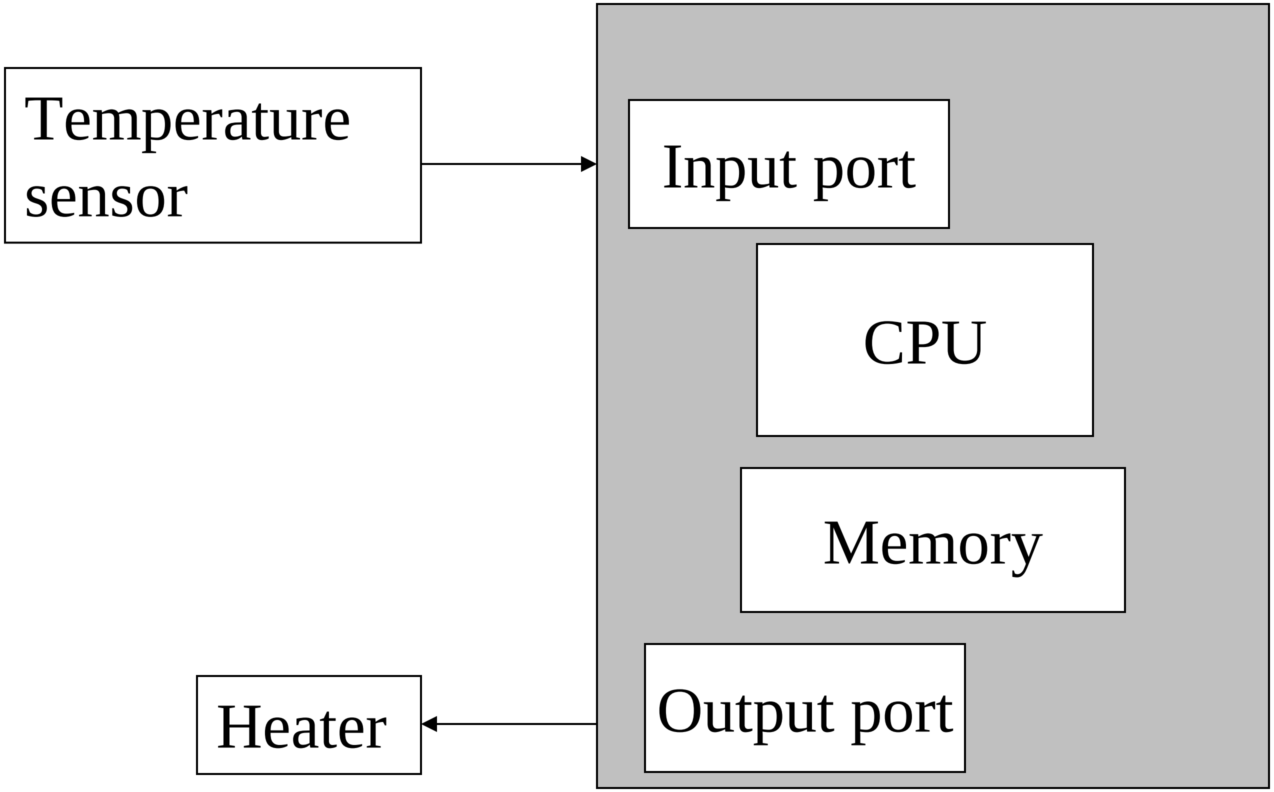
aperiodic

Types of Event to trigger RTES

- Sporadic
  - can occur any time, but
  - minimum time between arrivals



Typical Real time system example



Typical Real time system example

```
While true do
            {
            read temperature sensor
            if temperature too high
               then turn off heater
               else if temperature too low
                    then turn on heater
                else nothing
            }
```

Applications of RTES

Healthcare	Wearables and implantable devices, diagnostic tools and equipment (MRI and CT scanners, ultrasound scans), laboratory analytical applications
Transportation	GPS trackers and control systems, transport telematics
Manufacturing	Industrial robots, monitoring sensors, predictive analytics
Smart homes and cities	Smart lighting, heating and air conditioning, smart parking, communication systems, digital signage with multimedia content, surveillance systems

Thank you