INTRODUCTION TO COMMUNICATION INTERFACES

COMMUNICATION

Communication refers to the process of exchanging data between different components or devices within a system by following defined rules and protocols and allowing various parts to interact and function together such as reading sensor data, sending commands to actuators etc.,

It is the process of exchanging data between different components of an embedded system such as microcontrollers, sensors, actuators, etc., This communication is essential for the system to function correctly such as sending commands to actuators, reading sensor data etc., Communication in embedded systems can be classified into different categories based on how data is transferred and the type of connection used.

1. Based on Data transmission mode
    1. Serial Communication – Data is transmitted bit by bit over a single or few wires. It is slower than parallel communication but more efficient for long distances. Examples – UART, SPI, I2C, CAN, USB
    2. Parallel Communication – Data is transmitted multiple bits at a time using multiple data lines. It is faster but requires more wires and is suitable for short distances. Examples – GPIO, memory bus, LCD/Display interfaces.
2. Based on Connection type
    1. Wired Communication – Physical wires or cables are used for data transmission. It is more reliable and faster but limited by cable length. Examples – UART, SPI, I2C, CAN, Ethernet.
    2. Wireless Communication – Data is transmitted over radio waves, infrared or other wireless technologies. It is suitable for remote or IoT applications but may be affected by interface. Examples – Wifi, Bluetooth, Zigbee, LoRa, NFC.

COMMUNICATION INTERFACES

- Embedded systems often operate in environments where they need to interact with other devices, sensors, or networks.
- To facilitate this interaction, various **communication interfaces** are used.
- These interfaces enable embedded systems to exchange data efficiently, ensuring proper functionality and integration into larger systems.
- It is essential for communicating with various subsystems of the embedded system and with the external world.

Importance of Communication Interfaces

- Communication interfaces are critical in embedded systems for:
- **Data exchange**: Sending and receiving information between microcontrollers, sensors, and other components.
- **Peripheral integration**: Connecting external devices like memory cards, displays, or input devices.
- **Networking**: Enabling embedded devices to connect to larger networks like the Internet or industrial control systems.
- **Real-time control**: Ensuring timely data transfer for applications like automation and robotics.

Types of Communication Interfaces

- Onboard Communication Interface (Device/board level communication interface)
  - It refers to the different communication channels/buses for interconnecting the various integrated circuits and other peripherals within the embedded system.
  - Examples – Serial interfaces like I2C, SPI, UART, 1-Wire, etc., and parallel bus interface.
- External Communication Interface (Product level communication interface)
  - It refers to the different communication channels/buses used by the embedded system to communicate with the external world.
  - Examples – RS-232 C and RS-485, Universal Serial Bus (USB), IEEE 1394 (Firewire), Infrared (IR), Bluetooth (BT), Wi-Fi, ZigBee, GPRS etc.,

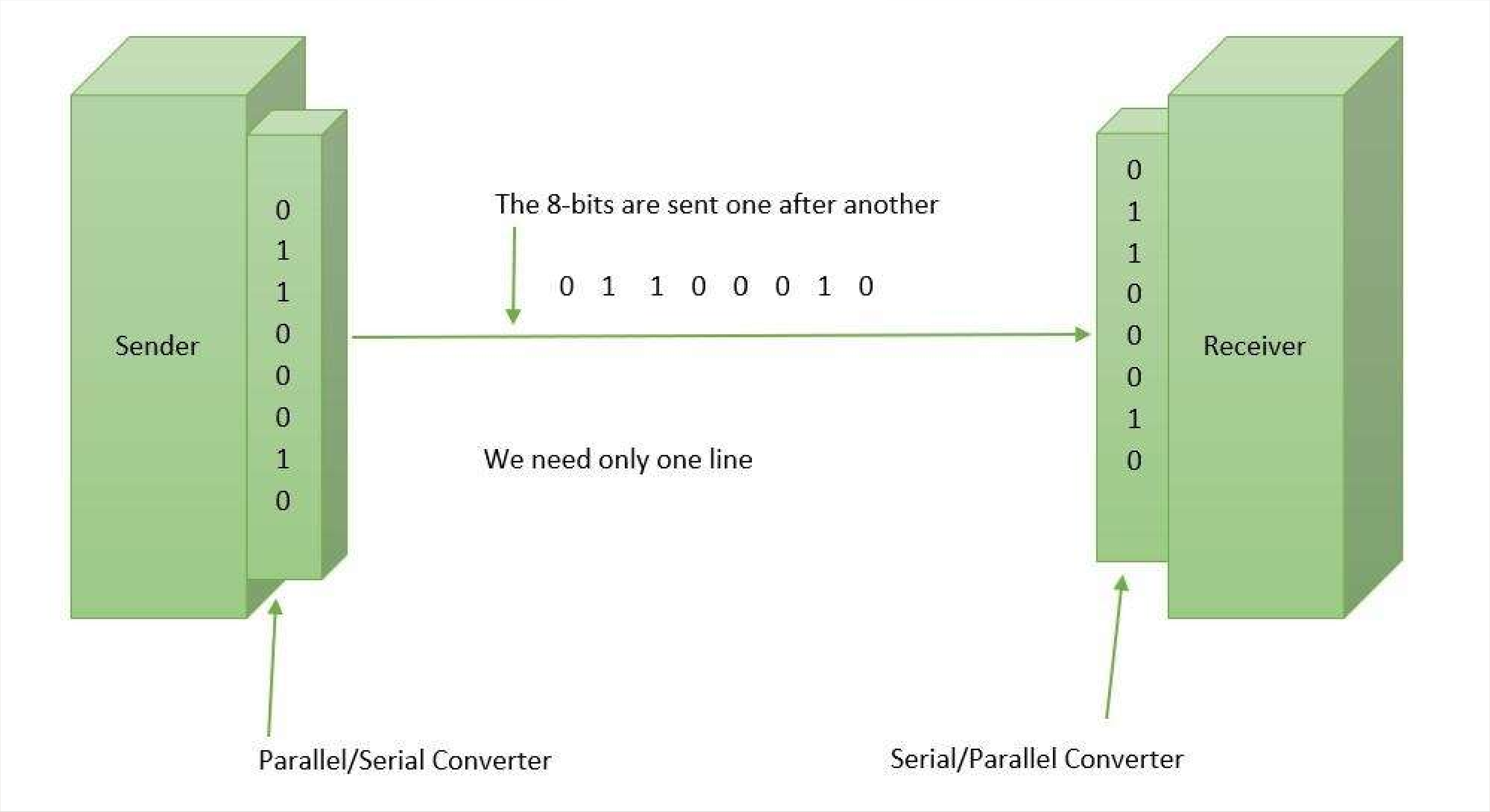
SERIAL VS PARALLEL COMMUNICATION SERIAL COMMUNICATION PROTOCOLS

SERIAL COMMUNICATION

- Serial Communication in a Microprocessor is a bit-by-bit operation.
- In this process, the data transmission takes place through a communication channel in a serial way.
- The microprocessor receives the data individually in the form of bits and makes it a complete data set.
- In simple words, Serial Communication in a Microprocessor takes place serially.

The complete process of Serial Communication in a Microprocessor as mentioned below.

- **Staring Process of Communication**: The data-sending process starts from a transmitter. The transmitter sends the first bit to the destination device or receiver.
- **Data Transmission Process**: The transmitter starts to send the whole data bit-by-bit in a specific order. These messages are long and take a certain time to send.
- **End Of Communication**: When the transmitter finishes its work, it stops the bit transmission process.
- **Optional Error Checking Process**: This step is related to the receiver. The receiver checks the whole data which is known as the parity checking process. If there are some faults, the receiver asks the transmitter to send the data again.


Sender
0
1
1
0
0
0
1
0
The 8-bits are sent one after another
0 1 1 0 0 0 1 0
We need only one line
Receiver
0
1
1
0
0
0
1
0
Parallel/Serial Converter
Serial/Parallel Converter

Serial Communication

Advantages of serial communication

- Serial communication in a microprocessor needs less wired connection than the parallel communication process.
- The structure of the serial communication process has low implementation costs and does not have complex hardware it.
- The data transfer method in the serial communication process is simple and transparent.
- Most of the telecommunication networks use this type of communication process for its simplicity.

Disadvantages of serial communication

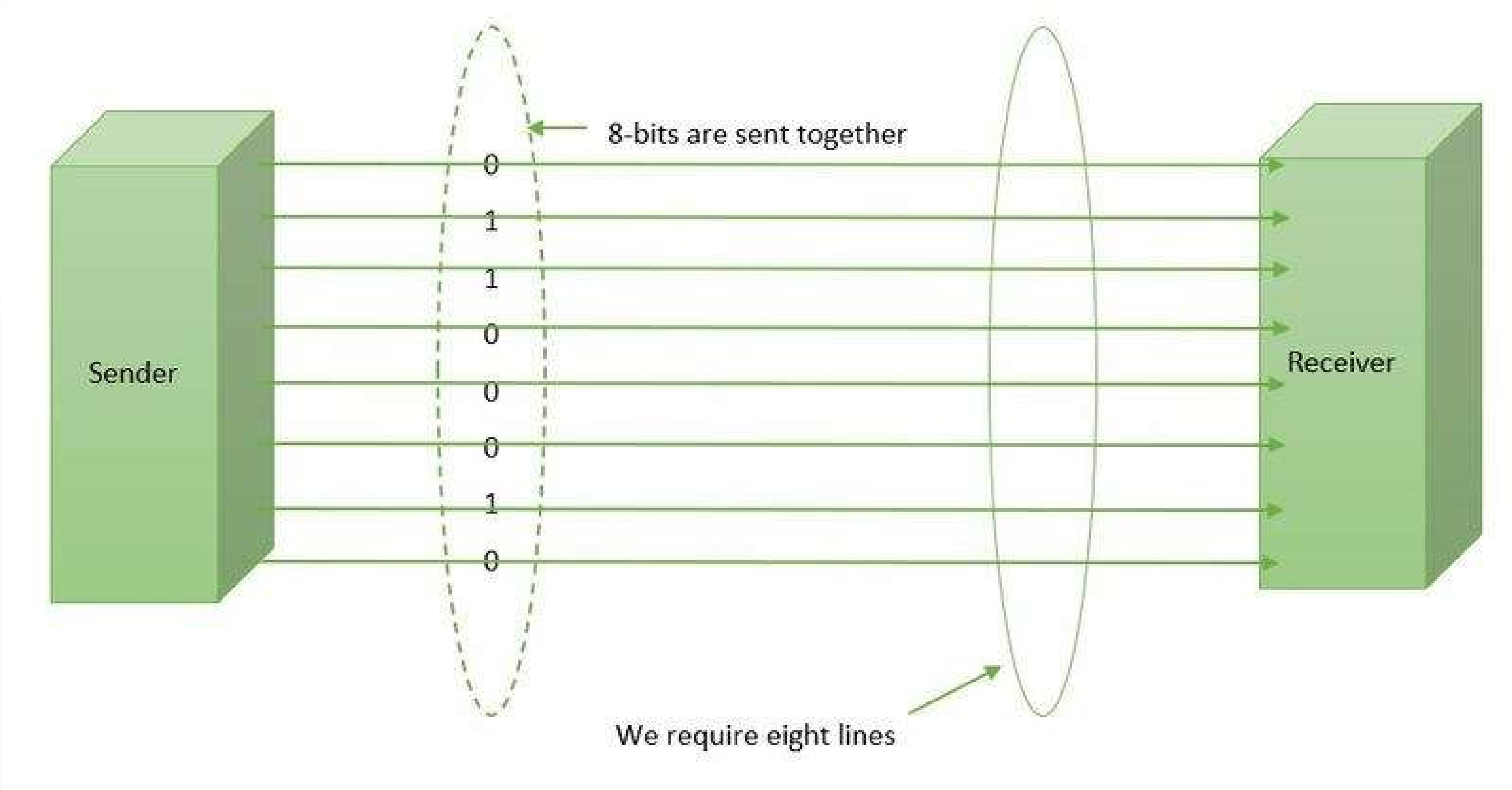
- Serial communication in a microprocessor is slower than the parallel communication process.
- The data transferring capability is low due to the lower bandwidth.
- The data needs to be organized properly as the serial communication process is not able to do that on its own.
- The sent data needs to be decoded properly after it reaches the destination to decrease the chances of error.

PARALLEL COMMUNICATION

- Parallel Communication in a Microprocessor happens when the multiple data transmitted to the destination through the multiple channels.
- The basic bit structure of the Parallel Communication system is 8-bit which is known as byte.
- The data transfer process in parallel communication takes place in a single clock pulse.
- Each bit chooses its cable to travel to the destination.

The complete process of Parallel Communication in a Microprocessor as mentioned below.

- **Staring Process of Communication**: The data-sending process starts from a transmitter. The transmitter sends the first bit to the destination device or receiver.
- **Data Transmission**: The data can be divided into sets of multiple groups. These messages are complex and take a certain time to send through separate cables at the same time.
- **Reception Of Data**: The receiver gets the data and arranges it in the correct order to decode the message.
- **End Of Communication**: After the data transfer process, the receiver decodes the data and closes the process with a complete status.
- **Optional Error Checking Process**: This step is related to the receiver. The receiver checks the whole data which is known as the parity checking process. If there are some faults, the receiver asks the transmitter to send the data again.



Parallel Communication

Advantages of parallel communication

- Parallel communication is faster than serial communication.
- Tt provides a high data transfer rate.
- This type of communication process creates a smaller number of errors.
- The data synchronization process is efficient as it sends multiple bits to the receiver simultaneously.

Disadvantages of parallel communication

- This type of communication process is costly as it needs more complex hardware to execute the process.
- The system transfers the data at a single clock time, but it suffers from the 'skew' due to the involvement of different wires.
- It can create a mess with the high-value data due to the latency factor present in it.
- The process transfers the data at the same time but through different cables. It can create overlapping errors in the receiver data.

The difference between the Serial and Parallel Communication in Microprocessor as mentioned below.

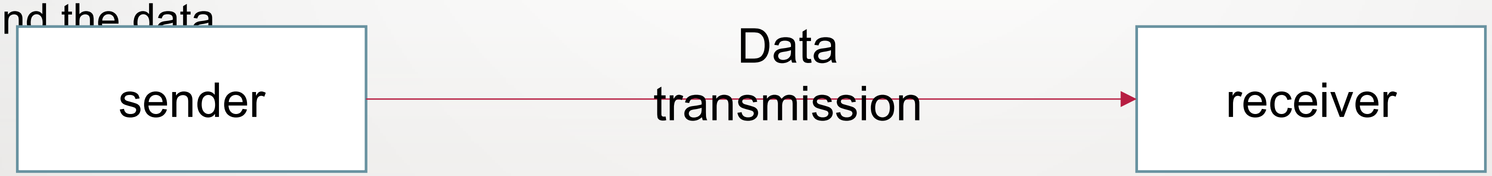
Aspects	**Serial Communication**	**Parallel Communication**
Speed	Slower for short distances	Faster as multiple bits are sent at once
Complexity	Simple for Long distances	Simple for Short distances
Cost	Cheaper for long distances	Expensive for long connections
Reliability	Reliable over long distances	May suffer signal degradation sometimes over long distances
Interference	Less prone to crosstalk	More prone to crosstalk in longer connections
Synchronization	Complex at very high speeds	Easier to synchronize at short distances
Scalability	High speed scalability can be challenging	Can be easily scaled for short distances
Wiring	Requires fewer wires	Requires more wires
Bandwidth	Bandwidth is limited by channel characteristics	High bandwidth potential

SERIAL COMMUNICATION INTERFACE

- A Serial Communications Interface (SCI) is used for serial data transmission between devices.
- It empowers the trade of information by sending bits consecutively over a single channel, which can be a wire or remote medium.
- SCI ordinarily employments a combine of signals for communication: one for transmitting (Tx) and one for getting Reception(Rx).
- This interface is essential in inserted frameworks, permitting microcontrollers to communicate with peripherals, other microcontrollers, and computers.

Types of Serial Communication Interface: It is classified into three types. They are

Simplex: Simplex communication involves one-way transmission of signals, with a clear sender and receiver. It is like a one-lane road, allowing efficient and straightforward communication. The reverse communication is not possible, only the sender alone can send the messages to receiver. The receiver only can read it but cannot send the data.


sender

Data transmission

receiver

Components

Sender/Transmitter: The device or module responsible for sending the data.

Receiver: The device or module that receives and processes the data.

Communication Medium: The physical or wireless medium through which the data is transmitted, such as cables, fiber optics, or radio waves.

Half Duplex: This mode uses a single communication channel for both transmission and reception, but not simultaneously. Devices must take turns to send and receive data.

If sender sends the message to receiver than after the the message is successfully transmitted only the receiver can send the return message to sender. In the same way if receiver sends the message to sender than after the the message is successfully transmitted only the sender can send the return message to receiver.

Sender/Receiver
Data transmission
Receiver/Sender

Components

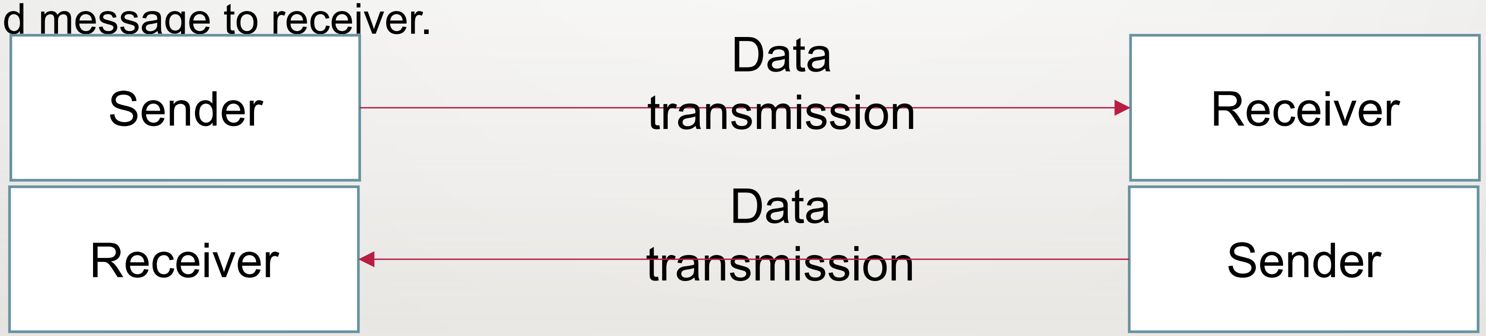
Sender/Transmitter: The device or module responsible for sending the data.

Receiver: The device or module that receives and processes the data.

Communication Medium: The physical or wireless medium through which the data is transmitted, such as cables, fiber optics, or radio waves.

Full Duplex: In this mode, data transmission and reception occur simultaneously on separate channels, allowing continuous bidirectional communication. It involves both side communication same as half duplex but here simultaneous communication is possible.

If sender sends the message to receiver at the same that is possible the receiver can also send message to sender. In the same way if receiver sends the message to sender than at the same that is possible the sender can also send message to receiver.



Components

Sender/Transmitter: The device or module responsible for sending the data.

Receiver: The device or module that receives and processes the data.

Communication Medium: The physical or wireless medium through which the data is transmitted, such as cables, fiber optics, or radio waves.

Advantages of SCI

- Simplicity: Easy to implement and understand.
- Low Cost: Requires less cost to construct.
- Versatility: Suits for various applications and devices.
- Long Distance: It works good for longer communication.
- Error Checking: Parity bits help in detecting errors in transmission.

Disadvantages of SCI

- Slower Speed: It is slower than parallel communication.
- Limited Data Size: Generally transmits one byte at a time.
- Asynchronous Issues: It Requires precise timing for reliable communication.
- Overhead: Start and stop bits add to the data overhead.
- Error Handling: It requires Simple error detection, but not correction.

SERIAL COMMUNICATION PROTOCOLS

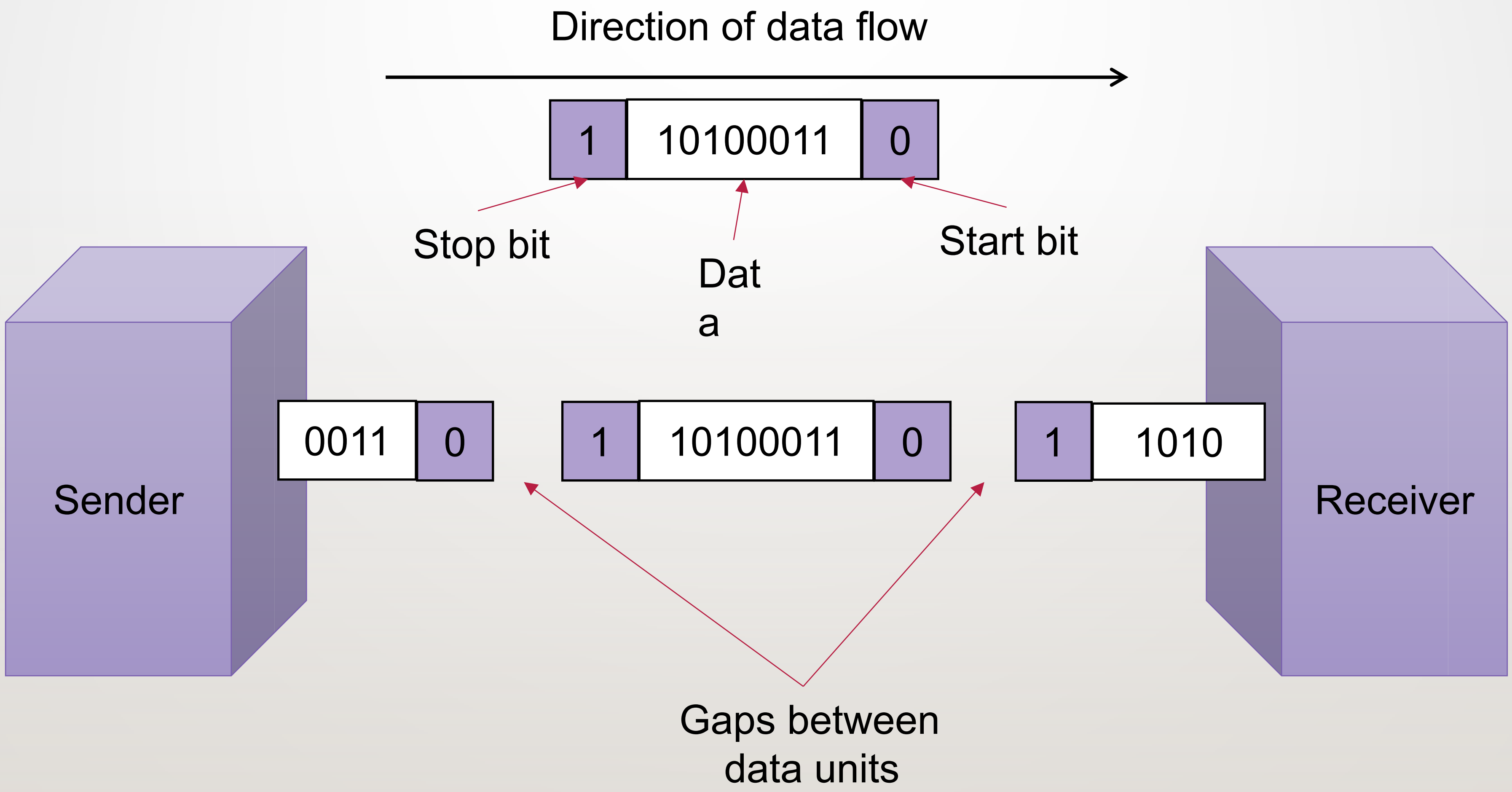
- Serial Communication in a Microprocessor is a bit-by-bit operation.
- In this process, the data transmission takes place through a communication channel in a serial way.
- The microprocessor receives the data individually in the form of bits and makes it a complete data set.
- In simple words, Serial Communication in a Microprocessor takes place serially.
- Serial communication is classified into two categories
  - Asynchronous data transmission – UART, Modem, RS232C
  - Synchronous data transmission – SPI, I2C

Asynchronous Serial Transmission: Asynchronous serial transmission is a widely used communication method in embedded systems.

It enables data transfer between devices without requiring a shared clock signal. The data can be sent and received with mutual agreement of the sender and receiver. The data transmission occurs at byte level only.

It relies on synchronization techniques, such as start and stop bits to manage data flow. There can be a delay between communication of two bytes. Only one byte can be sent at a time after a gap of delay the next byte is sent.

Common protocols using Asynchronous Serial Communication are – UART (Universal Asynchronous Receiver-Transmitter), USART – Universal Synchronous/Asynchronous Receiver-Transmitter), Physical Layer Standards (RS-232, RS-485, RS-422)


Direction of data flow
1
10100011
0
Stop bit
Dat
a
Start bit
Sender
0011
0
1
10100011
0
1
1010
Receiver
Gaps between
data units

Key Features

1. No shared clock – It does not require shared clock between the sender and receiver.
2. Start and stop bits – A start bit (logic low – 0) indicates beginning of a data frame. A stop bit (logic high – 1) indicates end of the frame.
3. Data bits – Typically, 7, 8, or 9 bits represent actual data.
4. Parity Bit (optional) – Used for basic error detection. If disabled error checking relies on higher-level protocols.
5. Baud rate – Both sender and receiver must agree on the baud rate example – 9600, 115200 bps.

Advantages

- Simple hardware implementation – requires TX, RX and GND connections.
- Low-cost and commonly supported by microcontrollers.
- Flexible baud rate selection.

Disadvantages

- Slower compared to synchronous communication.
- Overhead due to start and stop bits.
- Higher susceptibility to timing mismatches at extreme baud rates.

Applications

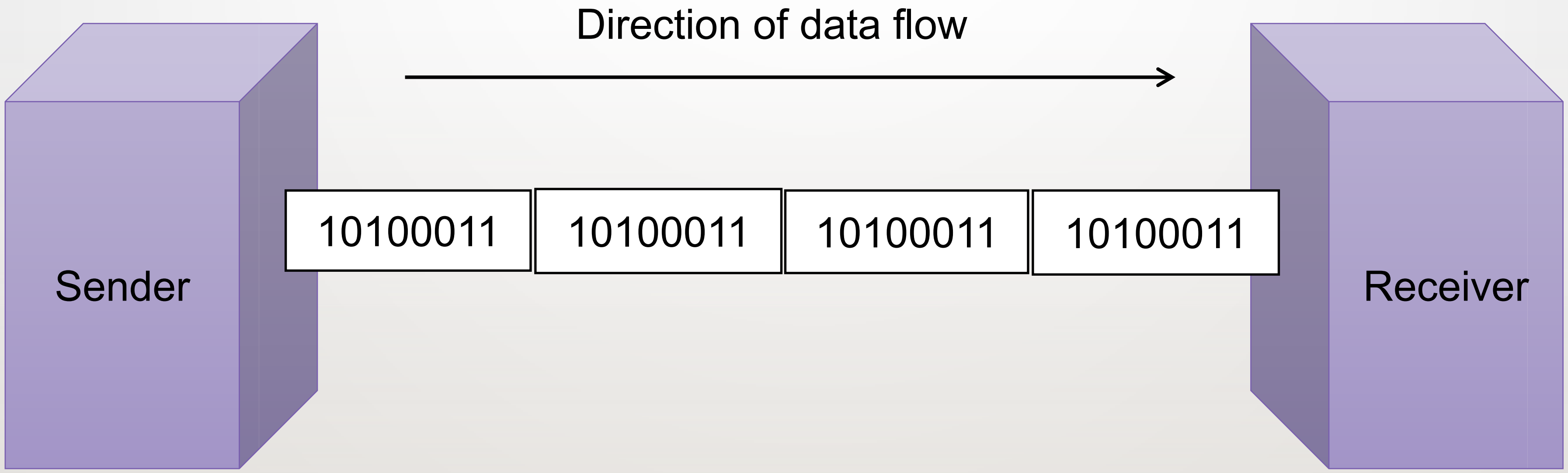
- Serial console communication example – debugging with a PC.
- Data logging and sensor interfacing.
- Wireless modules example – Bluetooth, Zigbee, LoRa.
- Interfacing with GPS, GSM and other external modules.

Synchronous Serial Transmission: Synchronous serial transmission is a method of sending data where the sender and receiver share a common clock signal.

This ensures that data bits are transmitted at a consistent rate, making it more efficient and reliable compared to asynchronous communication.

The stream of bits is combined into bigger frames which may comprise of more than one byte. Each byte is transmitted without gap between bytes. No start and stop bits are used and the data frame is transmitted with fixed time intervals.

Common synchronous serial protocols are – SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit), UART (Universal Asynchronous Receiver-Transmitter) with external clock, CAN (Controller Area Network), USB (Universal Serial Bus)



Direction of data flow

Sender

10100011

10100011

10100011

10100011

Receiver

Key Features

1. Clock synchronization – A shared clock signal ensures proper timing of data transmission.
2. Higher data rates – Since no start/stop bits are needed, more data can be sent per unit of time.
3. Less overhead – Unlike asynchronous transmission, no extra framing bits are required.
4. Multiple Devices communication – Can support multiple devices using bus protocols.
5. Better Error Control – Data integrity is higher due to precise timing.

Advantages

- Higher Efficiency – No start/stop bits and reduced overhead.
- Better Synchronization – Ensures reliable data transfer.
- Supports Higher Speeds – Can handle high data rates.
- Scalability – Suitable for multiple device networks.

Disadvantages

- Complex Hardware – Requires clock signal management.
- Shorter Transmission Distance – Clock skew limits range
- More wiring – Additional clock line increases wiring complexity

Applications

- Communication between microcontrollers and peripherals.
- Data exchange in IoT and sensor networks.
- Industrial automation and automotive networking.
- Memory chip interfacing example – EEPROM, Flash memory.

USART (UNIVERSAL SYNCHRONOUS/ASYNCHRONOUS RECEIVER TRANSMITTER)

USART

- USART is a serial communication protocol used in embedded systems to transfer data between devices.
- It supports both synchronous and asynchronous communication. It is versatile for different applications.
- It is a coordinated circuit that is appended to different specialized gadgets.
- USART changes the parallel information into the sequential structure.
- In USART, the receiver doesn’t have to know the baud rate of the transmitter as it gets information from the master and the clock signal.
- A clock is generated by sender and is received by receiver without knowing the baud rate.

UART

- UART is a large-scale integration gadget used for transferring data asynchronously.
- It provides lower data transfer speed as compared to USART.
- There is no clock signal in UART so, the receiver has to know the baud rate of the transmitter.
- The data clock is generated internally into the microcontroller and synchronized with data stream using a start bit transition.
- It lies between frameworks that deals with parallel communication and the gadgets that handles serial communication.

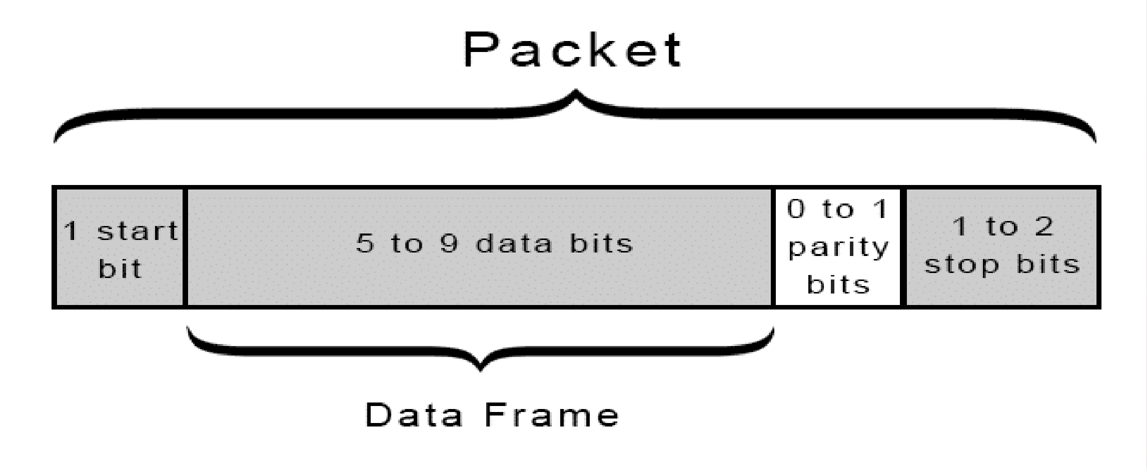
Types of communication in USART

- Synchronous mode – Data is transmitted with a clock signal, ensuring precise timing. It transmits data in frames.
  - In synchronous operation, data must be provided on time until a frame is complete.
  - If the processor does not provide data on time, it shows an error called as “underrun error”, and transmission of the frame is aborted.
  - The devices operating in synchronous mode use either character-oriented or bit-oriented.
  - In character-oriented modes, the device is relied on characters to define frame boundaries.
  - In bit-oriented modes, the earlier device is relied on physical-layer signals and later the devices took over the physical-layer recognition of bit patterns.
  - When the physical layer indicates that the modem is active, USART sends a steady stream of padding either characters or bits as per the device and protocol.

- Asynchronous mode – Data is transmitted without a clock signal, using start and stop bits for synchronization.
  - Both devices i.e., transmitter and receiver in asynchronous mode must be configured to use the same baud rate for successful communication.
  - In this mode data transmission and reception is allowed simultaneously.
  - This mode uses two wires – one for transmitting and one for receiving data.
  - Data is transmitted in frames including start bit, data bits, an optional parity bit, and stop bits.

USART FRAME FORMAT

- A typical USART frame consists of
  - Start Bit (1 bit) – Indicates the beginning of data transmission.
  - Data Bits (5 to 9 bits) – The actual data being transmitted.
  - Parity Bit (optional 1 bit) – Used for error checking.
  - Stop bit (1 to 2 bits) – Indicates the end of the frame.



CONFIGURATION PARAMETERS

- When configuring a USART, some parameters like baud rate, data length, parity, stop bits and data order should be set to ensure proper communication.
- Baud rate
  - This defines the speed at which the data is transmitted. It is measured in bits per second (bps).
  - Ensure both the transmitting and receiving devices use the same baud rate.
  - Common baud rates include 4800, 9600, 19200, 38400, 57600, and 115200.
- Data length
  - Specifies the number of data bits per byte (ex., 8-bit data is common).
- Parity
  - A method for error detection during data transmission.
  - It includes no parity, even parity and odd parity.

- Stop bits
  - Indicate the end of a data frame
  - Common values are 1 or 2 stop bits.
- Data order (Bit order)
  - Determines the order in which data is transmitted. For example, least significant bit first or most significant bit first.
- Other parameters
  - Flow control – Mechanism to manage data flow, preventing the transmitter sending data faster than the receiver can handle.
  - FIFO – Some USART offer FIFO buffers for storing data to improve efficiency.
  - Oversampling – A technique used to improve the accuracy of baud rate generation.
  - Timeout – A mechanism to detect communication errors or inactivity.

Working of USART

- Transmitter Operation –
  - The transmitter sends data through the TX (Transmit) pin.
  - Data is loaded into the Transmit Data Register (TDR).
  - The data is shifted out, adding start, parity, and stop bits.
  - The receiver device receives the data via its RX (Receive) pin.
- Receiver Operation –
  - The receiver detects the start bit.
  - It reads the data bits and checks parity (if enabled).
  - The stop bit marks the end of transmission.
  - The data is stored in the Receive Data Register (RDR) for further processing.

USART Registers – Example: STM32, AVR, PIC

- USART_SR (Status Register): Holds flags like TXE (Transmit Empty), RXNE (Receive Not Empty).
- USART_DR (Data Register): Holds data to be transmitted or received.
- USART_BRR (Baud Rate Register): Sets the baud rate.
- USART_CR1, CR2, CR3 (Control Registers): Enable/disable USART, configure mode, and error control.

Applications of USART in Embedded systems

- Serial Communication with PC (via USB-UART converters)
- Microcontroller-to-Microcontroller Communication
- Data Logging and DebuggingGPS, GSM, and Bluetooth Modules
- IoT Devices and Wireless Communication

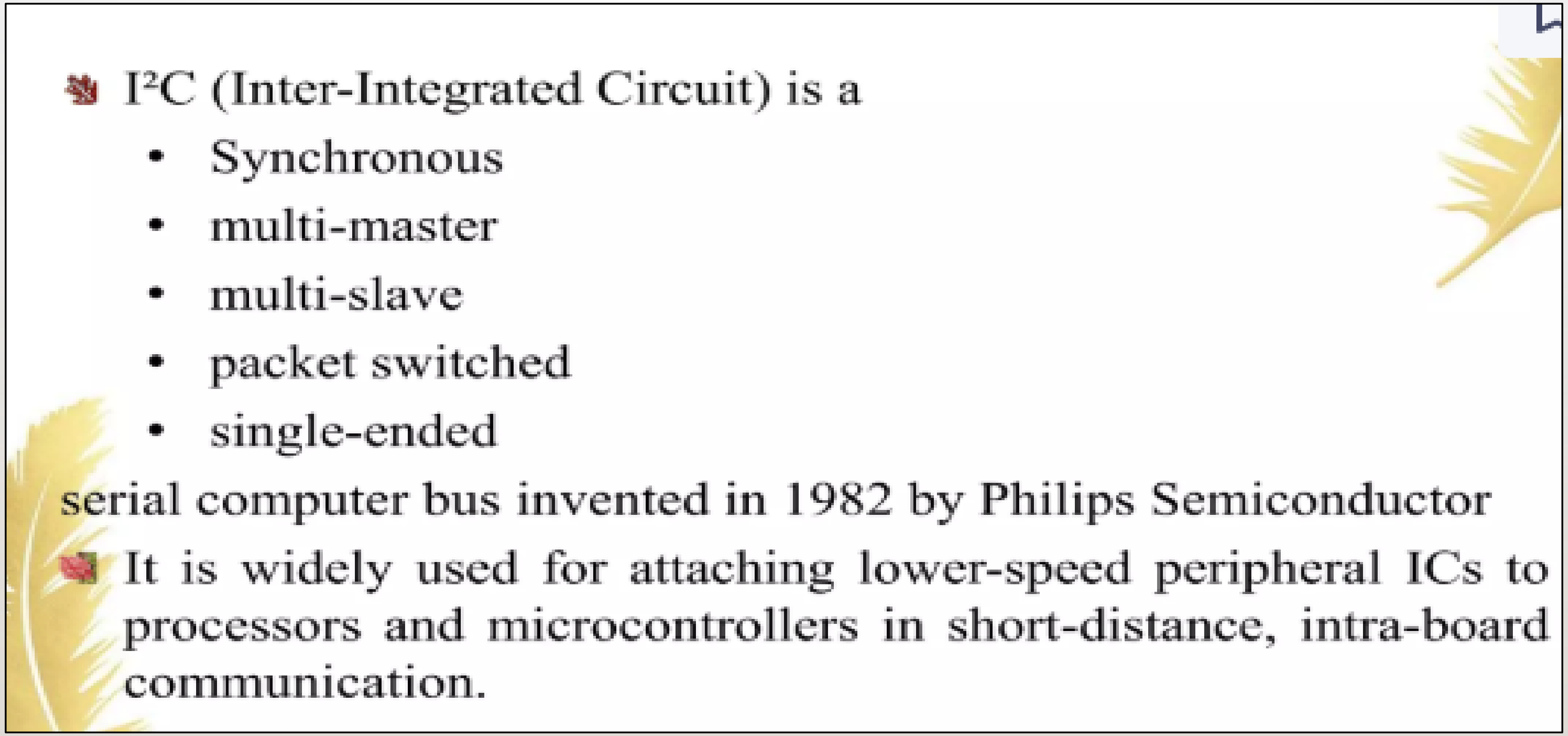
Advantages of USART

- **Supports Both Synchronous and Asynchronous Modes**: This makes USART more versatile than other interfaces that have been described so far in this article.
- **Higher Data Transfer Rate in Synchronous Mode**: Synchronous data transfer has an added advantage over asynchronous communication in a way that a clock signal can be used to speed up data transfer that is in synchronous mode.
- **Error Detection**: USART is usually provided with a means for detecting errors within the transmitted data such as parity.

Disadvantages of USART

- **More Complex to Implement**: This makes USART more complex to set up and use as well because of the dual mode functionality.
- **Requires Additional Clock Lines**: In synchronous mode it needs one more line for the clock signal and this results in a high pin count.

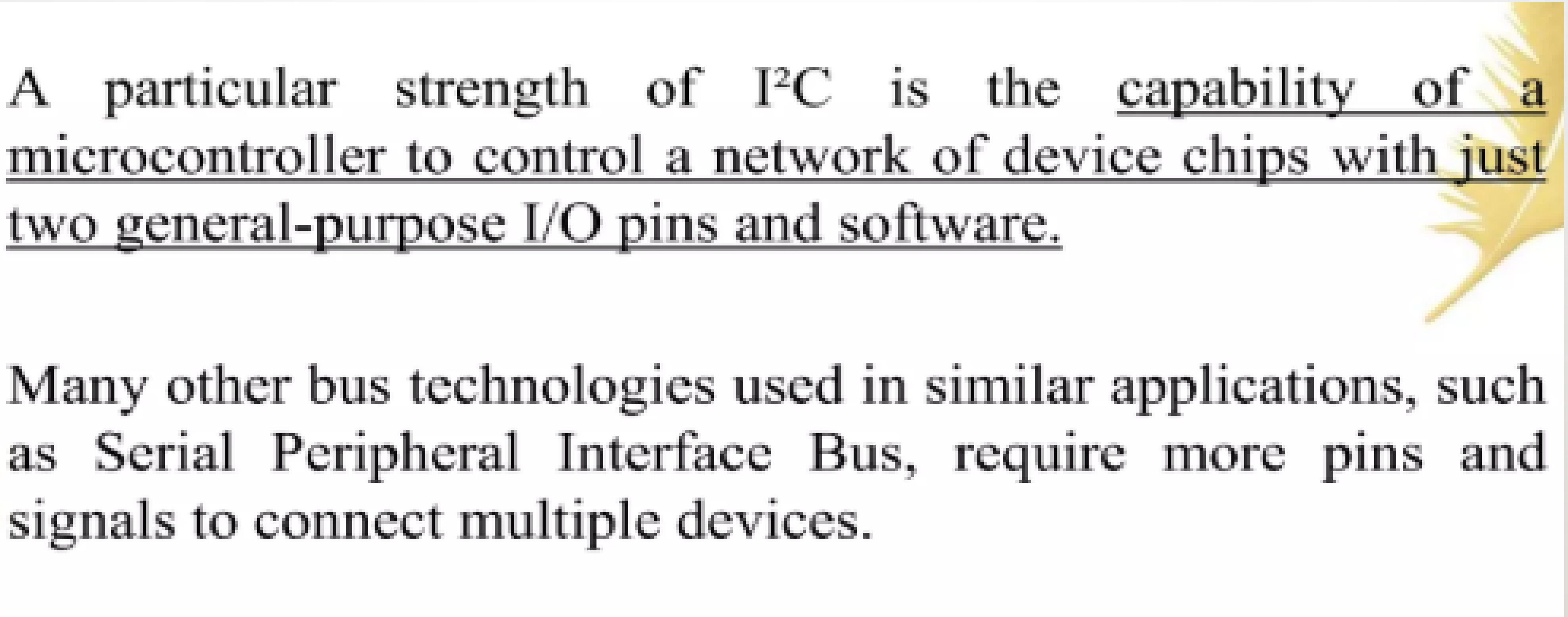
INTRODUCTION-I2C



- I^2C (Inter-Integrated Circuit) is a
  - Synchronous
  - multi-master
  - multi-slave
  - packet switched
  - single-ended

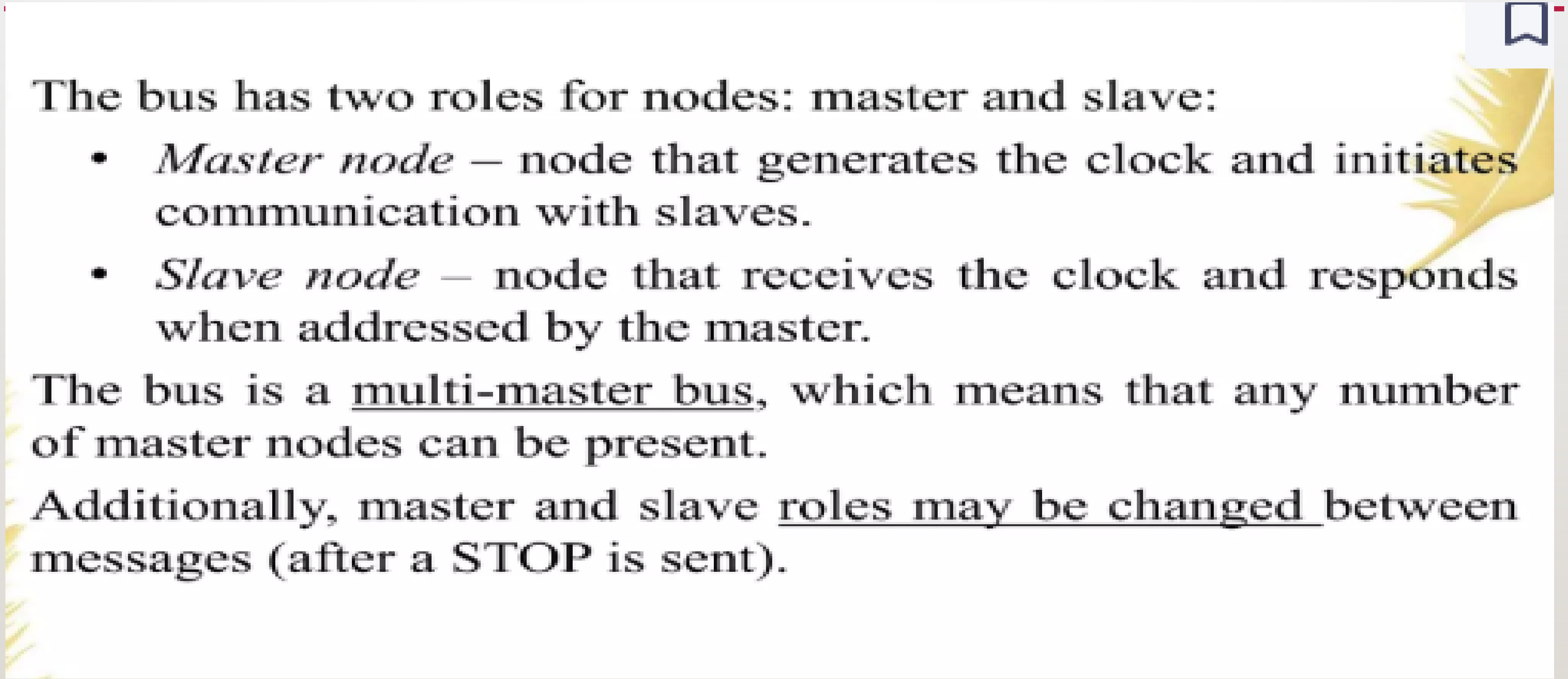
serial computer bus invented in 1982 by Philips Semiconductor

- It is widely used for attaching lower-speed peripheral ICs to processors and microcontrollers in short-distance, intra-board communication.



A particular strength of I^2C is the capability of a microcontroller to control a network of device chips with just two general-purpose I/O pins and software.

Many other bus technologies used in similar applications, such as Serial Peripheral Interface Bus, require more pins and signals to connect multiple devices.

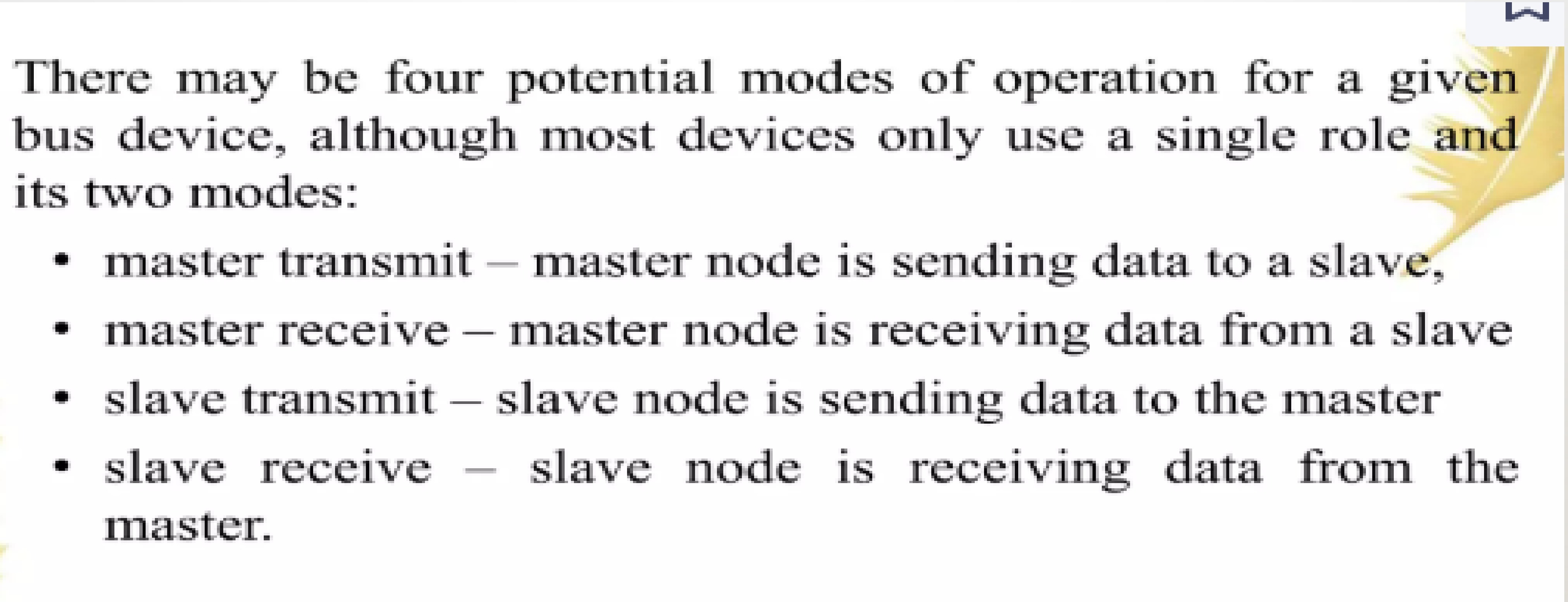


The bus has two roles for nodes: master and slave:

- *Master node* – node that generates the clock and initiates communication with slaves.
- *Slave node* – node that receives the clock and responds when addressed by the master.

The bus is a multi-master bus, which means that any number of master nodes can be present.

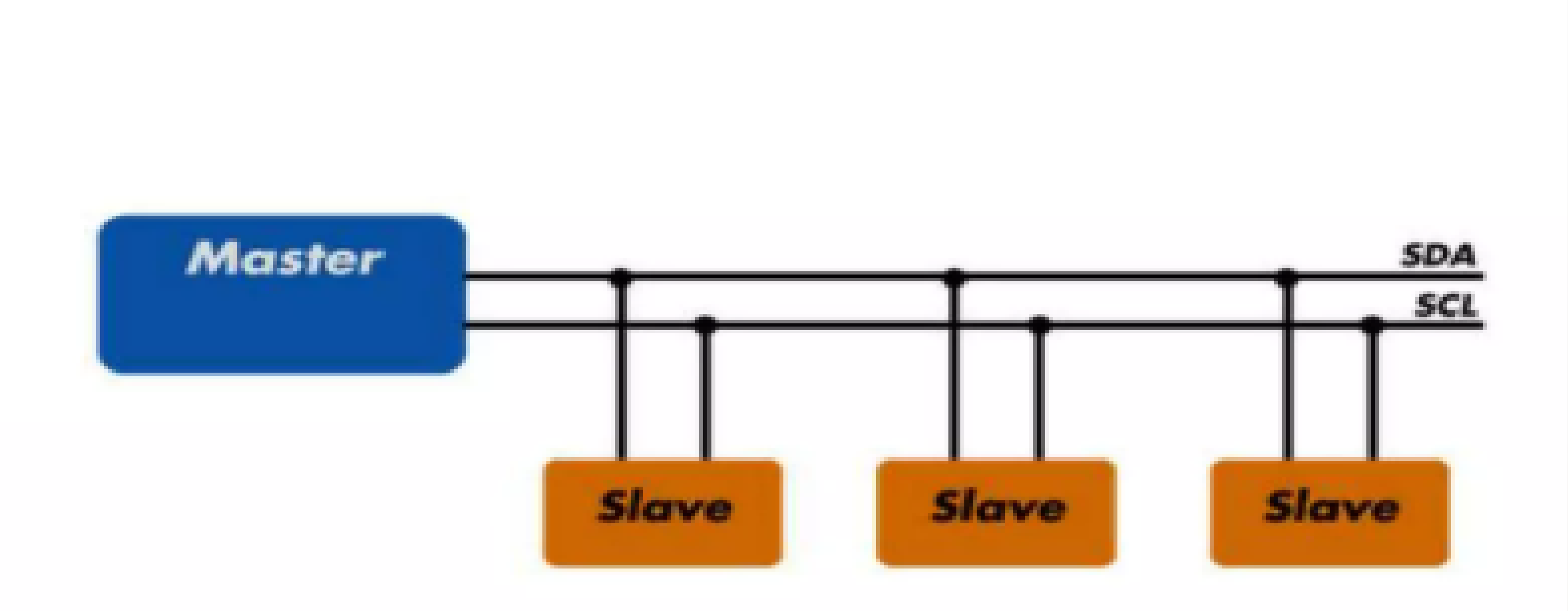
Additionally, master and slave roles may be changed between messages (after a STOP is sent).



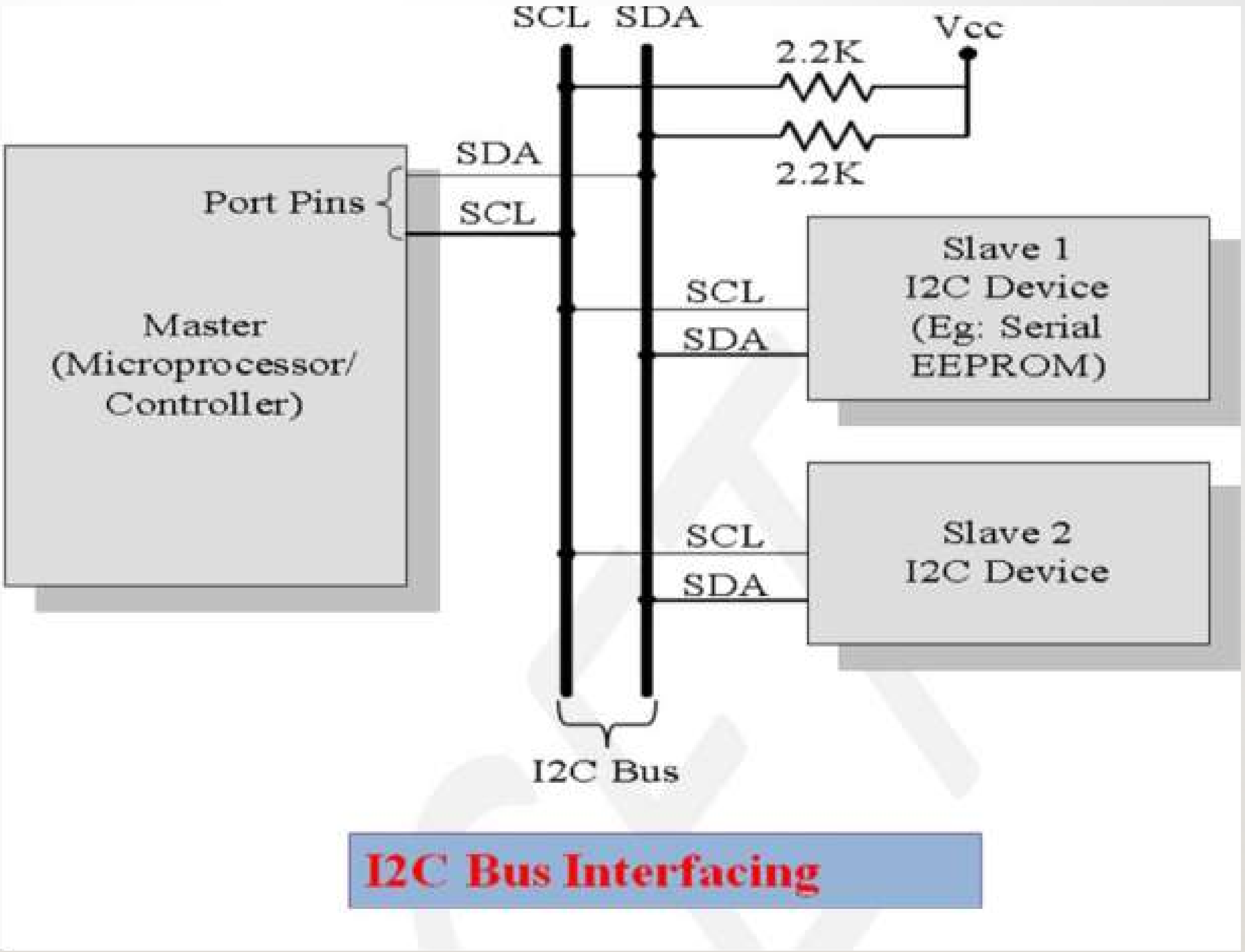
There may be four potential modes of operation for a given bus device, although most devices only use a single role and its two modes:

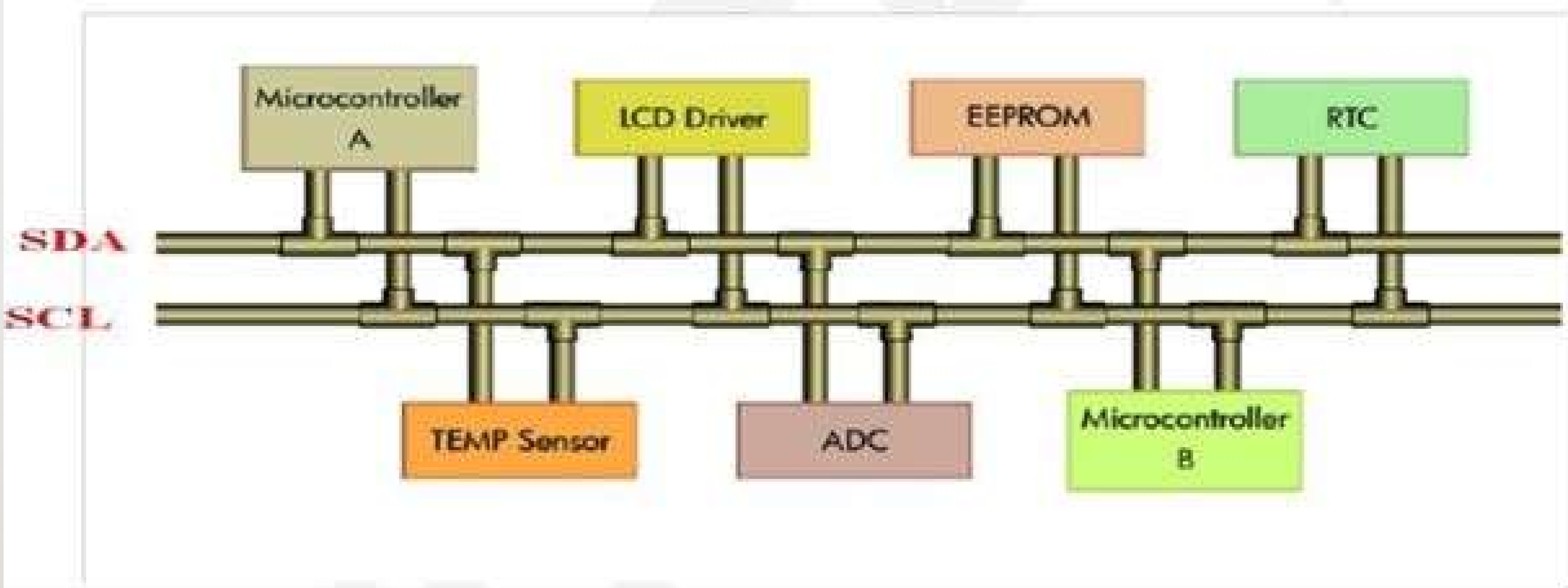
- master transmit – master node is sending data to a slave,
- master receive – master node is receiving data from a slave
- slave transmit – slave node is sending data to the master
- slave receive – slave node is receiving data from the master.

HARDWARE DESIGN



- It has two I2C bus lines
  - SCL-Serial Clock Line
  - SDA-Serial Data Line




Microcontroller A
LCD Driver
EEPROM
RTC
SDA
SCL
TEMP Sensor
ADC
Microcontroller B

- SCL line is responsible for generating synchronization clock
- 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.

- I2C addressing allows multiple devices to be connected to the same bus without conflicts.
- The master sends the address of the slave it wants to communicate with, and only the slave with the matching address responds with an ACK bit.
- This addressing can be either 7-bit or 10-bit, providing a range of unique addresses for different devices on the bus.

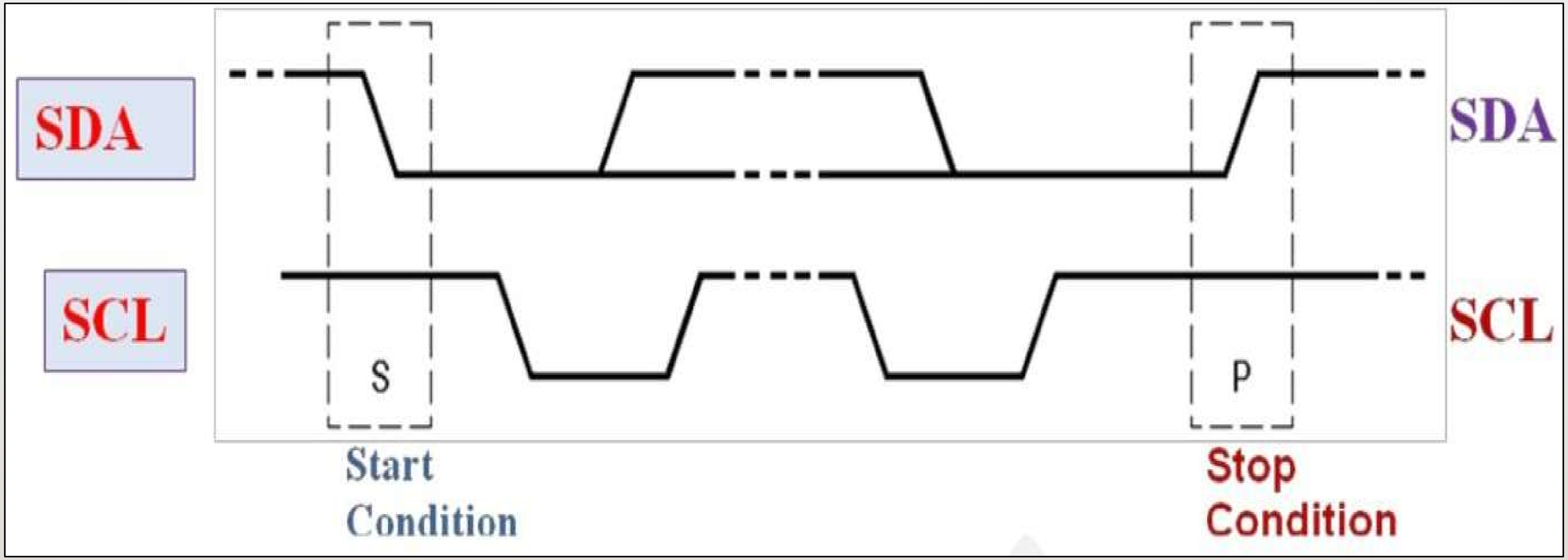
- **I2C Data Transfer Process**
- The data transfer process in I2C involves several steps, each with specific signal conditions and frame structures:

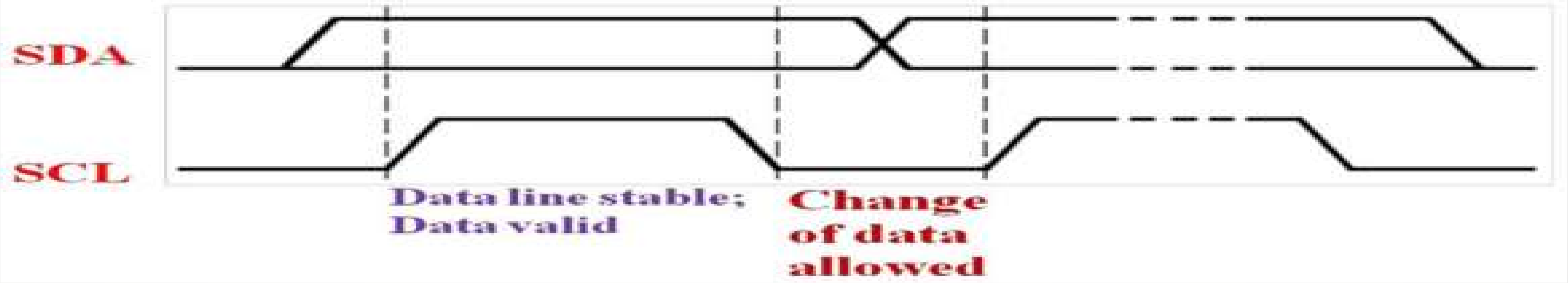
1. **Start Condition**: This is indicated by the SDA (Serial Data) line switching from high to low before the SCL (Serial Clock) line switches from high to low.
2. **Address Frame**: After the start condition, the master sends a 7 or 10-bit address frame that identifies the slave device it wants to communicate with. This frame also includes a read/write bit indicating the operation's direction.
3. **ACK/NACK Bit**: Following each frame, an acknowledge (ACK) or not acknowledge (NACK) bit is sent. A slave device sends an ACK bit if it successfully receives an address or data frame.
4. **Data Frame**: If an ACK is received, the master can proceed to send or receive data frames, which are 8 bits long. Each data frame is followed by an ACK/NACK bit.
5. **Stop Condition**: To end the communication, a stop condition is sent by switching the SDA line from low to high after the SCL line switches from low to high, with the SCL line remaining high.

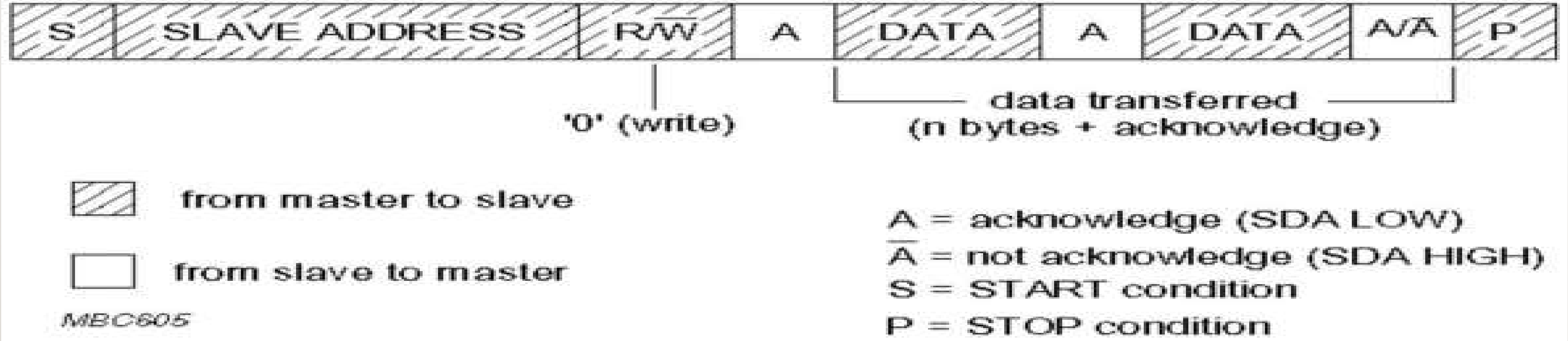
SEQUENCE OF OPERATION FOR COMMUNICATING WITH AN I2C SLAVE DEVICE :

- Master device pulls the clock line (SCL) of the bus to HIGH.
- Master device pulls the data line (SDA) to LOW, when the SCL line is at logic HIGH (This is the Start condition for data transfer).
- Master sends the address (7 bit or 10 bit wide) of the Slave device to which it wants to communicate, over the SDA line.
- Clock pulses are generated at the SCL line for synchronizing the bit reception by the slave device.
- The MSB of the data is always transmitted first.
- The data in the bus is valid during the HIGH period of the clock signal
- In normal data transfer, the data line only changes state when the clock is low.
- Master waits for the acknowledgement bit from the slave device whose address is sent on the bus along with the Read/Write operation command.
- Slave devices connected to the bus compares the address received with the address assigned to them.

- The Slave device with the address requested by the master device responds by sending an acknowledge bit (Bit value =1) over the SDA line.
- Upon receiving the acknowledge bit, master sends the 8bit data to the slave device over SDA line, if the requested operation is “Write to device”.
- 12.If the requested operation is “Read from device”, the slave device sends data to the master over the SDA line.
- 13.Master waits for the acknowledgement bit from the device upon byte transfer complete for a write operation and sends an acknowledge bit to the slave device for a read operation
- 14.Master terminates the transfer by pulling the SDA lines High, when the clock line SCL is at Logic HIGH (indicating STOP condition)


SDA
SCL
S
P
SDA
SCL
Start Condition
Stop Condition





▨ from master to slave

□ from slave to master

A = acknowledge (SDA LOW)

$\overline{A}$ = not acknowledge (SDA HIGH)

S = START condition

P = STOP condition

SPEED OF I2C

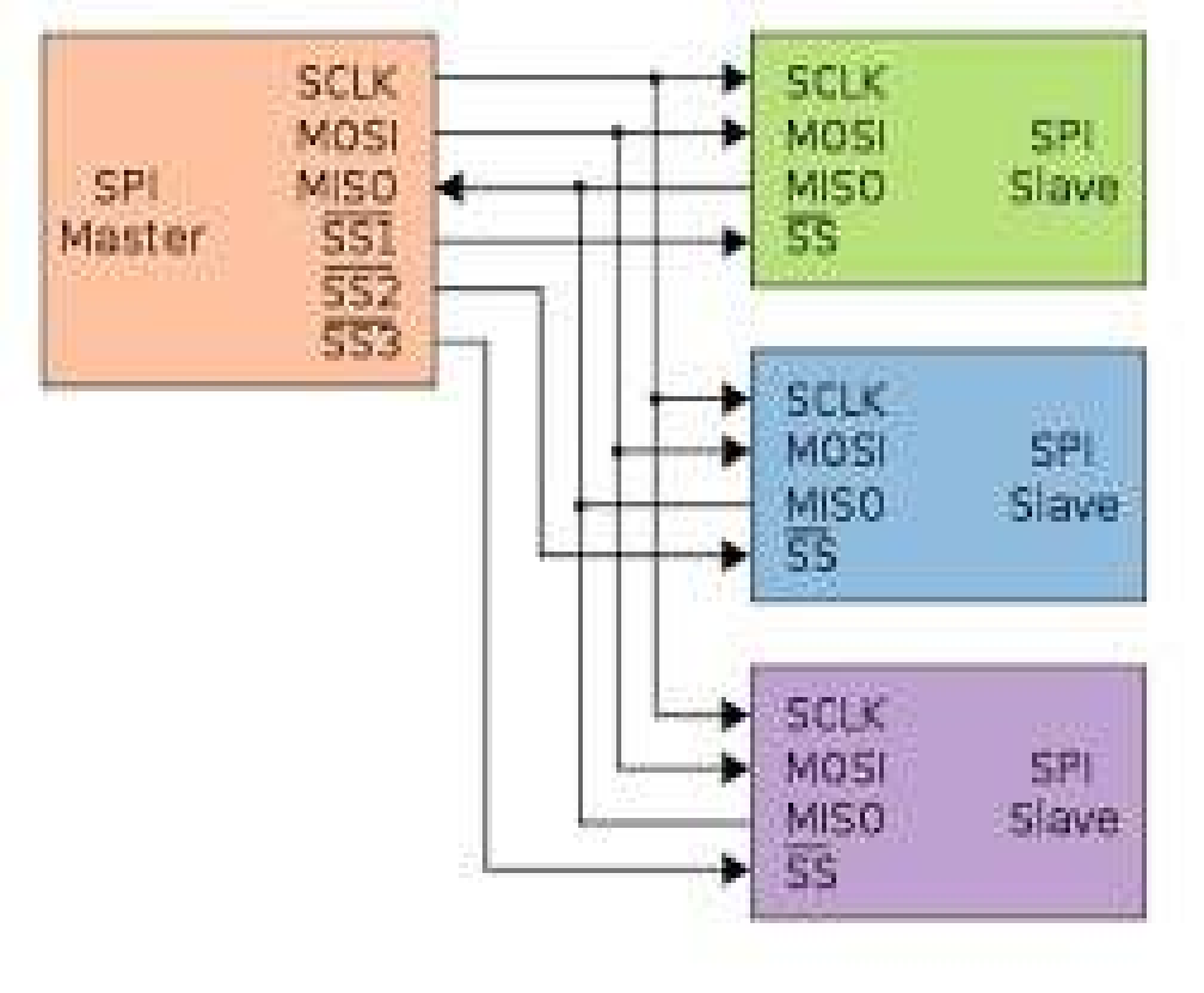
- I2C supports various speed modes, including:
- **Standard Mode**: Up to 100 kbit/s.
- **Fast Mode**: Up to 400 kbit/s.
- **Fast Mode Plus**: Up to 1 Mbit/s.
- **High-Speed Mode**: Up to 3.4 Mbit/s.
- **Ultra-Fast Mode**: Up to 5 Mbit/s.

ADVANTAGE -LIMITATION

- **Advantages** of I2C include its simplicity, the ability to connect multiple masters and slaves, and the low number of required wires (only two).
- **Limitations** such as a slower data transfer rate compared to other protocols like SPI and the potential for address collisions due to the limited address space.

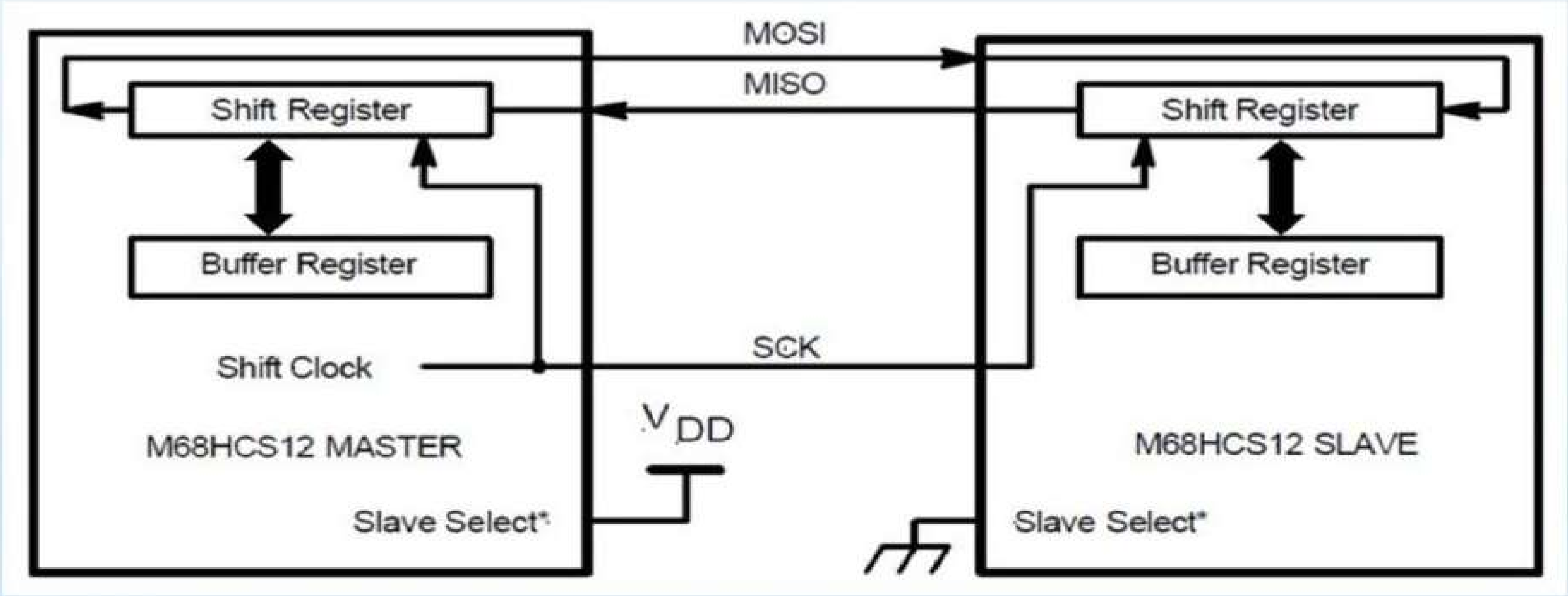
SERIAL PERIPHERAL INTERFACE BUS (SPI)

- 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.

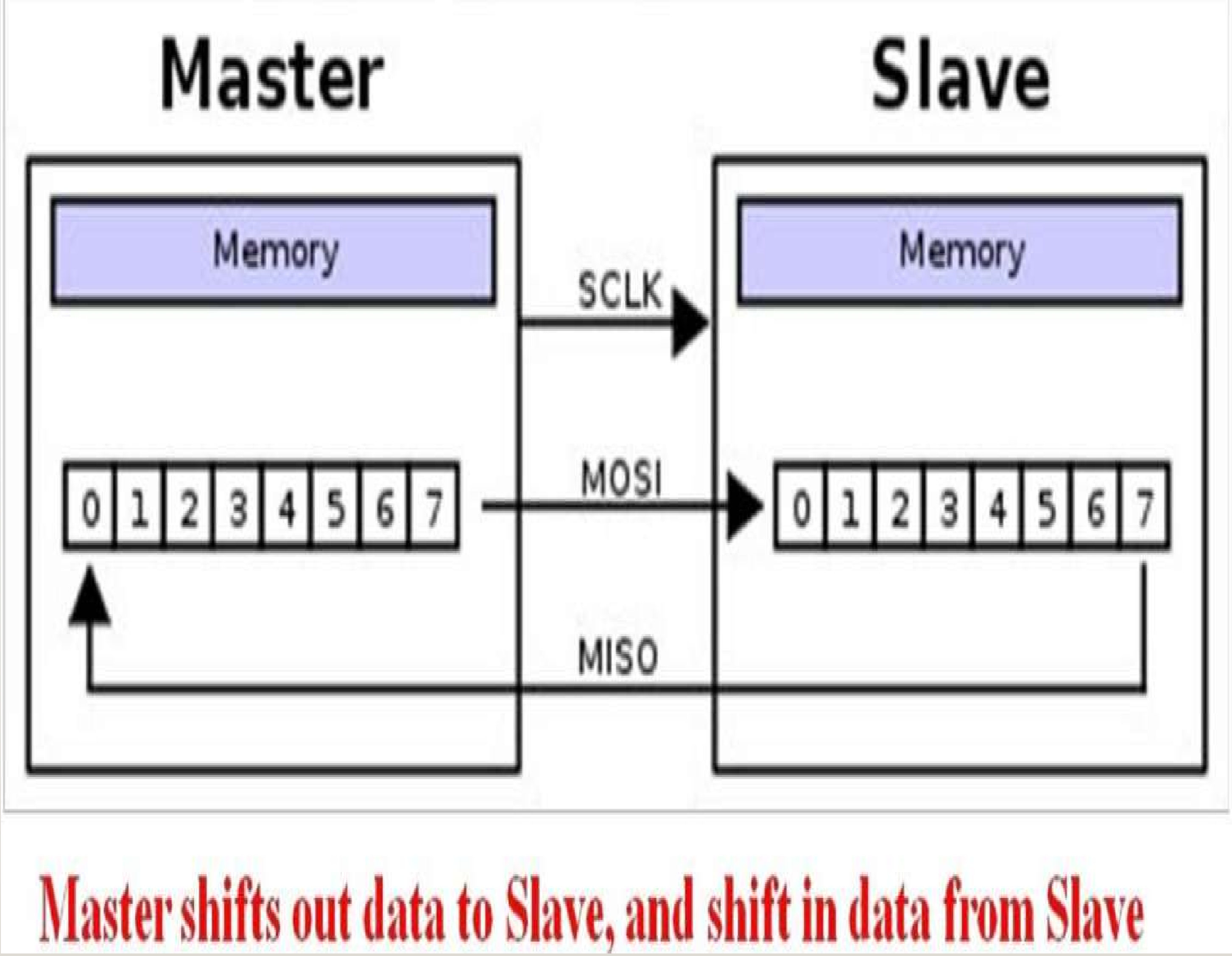

SPI
Master
SCLK
MOSI
MISO
SS1
SS2
SS3
SCLK
MOSI
MISO
SS
SPI
Slave
SCLK
MOSI
MISO
SS
SPI
Slave
SCLK
MOSI
MISO
SS
SPI
Slave

- SPI requires four signal lines for communication. They are:
- Master Out Slave In (MOSI): Signal line carrying the data from master to slave device. It is also known as Slave Input/Slave Data In (SI/SDI)
- Master In Slave Out (MISO): Signal line carrying the data from slave to master device. It is also known as Slave Output (SO/SDO)
- Serial Clock (SCLK): Signal line carrying the clock signals
- Slave Select (SS): Signal line for slave device select. It is an active low signal. The master device is responsible for generating the clock signal.

- Master device selects the required slave device by asserting the corresponding slave devices slave select signal LOW.
- The data out line (MISO) of all the slave devices when not selected floats at high impedance state
- The serial data transmission through SPI Bus is fully configurable.
- SPI devices contain certain set of registers for holding these configurations.
- The Serial Peripheral Control Register holds the various configuration parameters like master/slave selection for the device, baud rate selection for communication, clock signal control etc.
- The status register holds the status of various conditions for transmission and reception.
- SPI works on the principle of Shift Register.
- The master and slave devices contain a special shift register for the data to transmit or receive.
- The size of the shift register is device dependent. Normally it is a multiple of 8.

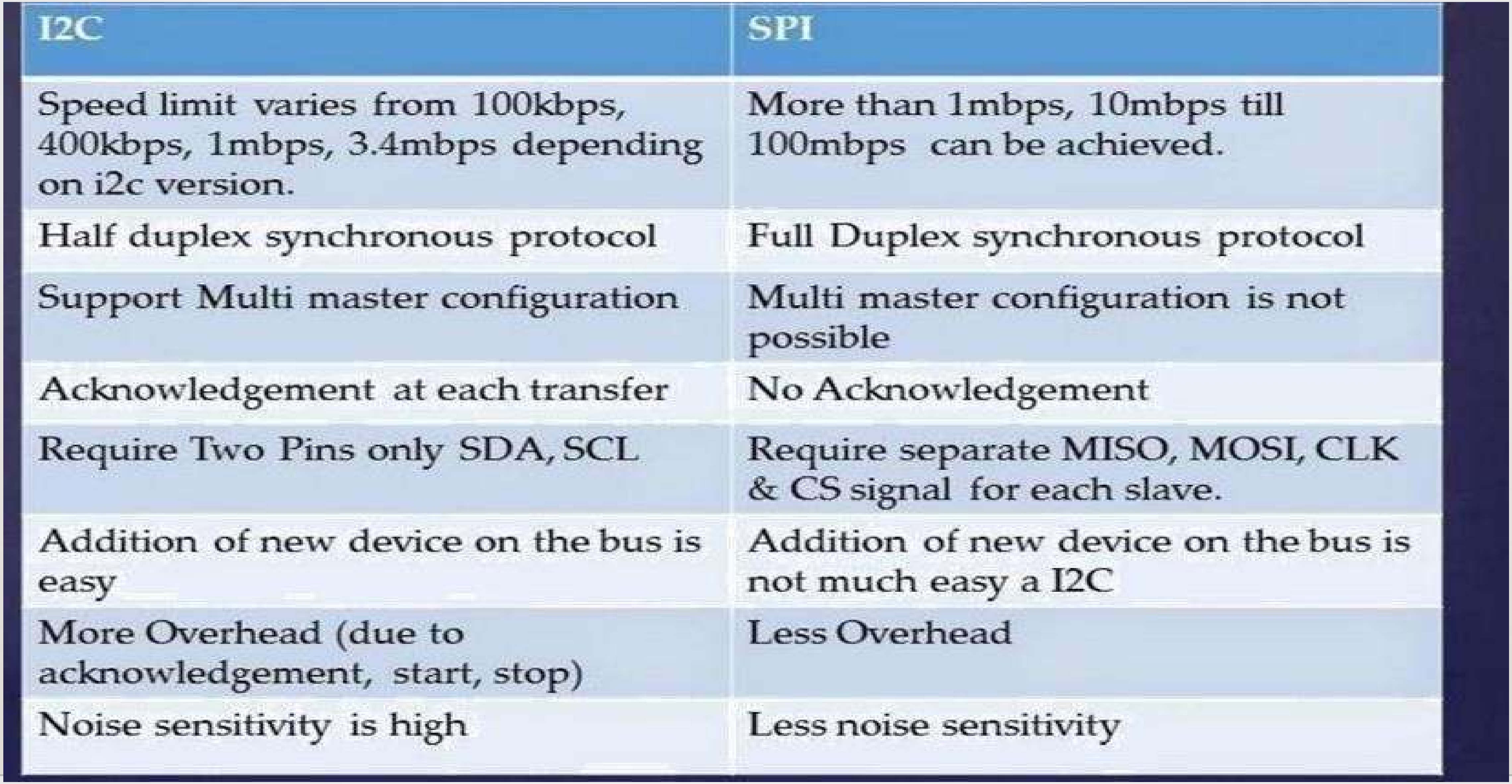


Master/slave serial peripheral interface.



Master shifts out data to Slave, and shift in data from Slave

- During transmission from the master to slave, the data in the master’s shift register is shifted out to the MOSI pin and it enters the shift register of the slave device through the MOSI pin of the slave device.
- At the same time, the shifted-out data bit from the slave device’s shift register enters the shift register of the master device through MISO pin



I2C	SPI
Speed limit varies from 100kbps, 400kbps, 1mbps, 3.4mbps depending on i2c version.	More than 1mbps, 10mbps till 100mbps can be achieved.
Half duplex synchronous protocol	Full Duplex synchronous protocol
Support Multi master configuration	Multi master configuration is not possible
Acknowledgement at each transfer	No Acknowledgement
Require Two Pins only SDA, SCL	Require separate MISO, MOSI, CLK & CS signal for each slave.
Addition of new device on the bus is easy	Addition of new device on the bus is not much easy a I2C
More Overhead (due to acknowledgement, start, stop)	Less Overhead
Noise sensitivity is high	Less noise sensitivity

CONTROLLER AREA NETWORK (CAN)

CAN - INTRODUCTION

- The Controller Area Network (CAN) protocol is a communication protocol that was developed for use in the automotive industry but has also been used in other industries such as industrial automation and medical equipment.
- It is a serial communication protocol that uses a multi-master, distributed control system.
- This means that any device on the network, called a node, can initiate communication and all other nodes on the network can participate in the communication.
- The protocol provides a way for devices to share information and synchronize their actions without the need for a central controller.
- The protocol uses a collision detection and arbitration method to prevent multiple nodes from transmitting at the same time and ensure that only one node can transmit at a time.

WHY CAN IN AUTOMOTIVE INDUSTRY

- The CAN protocol was developed for use in the automotive industry to address several challenges that arose as cars became more complex and incorporated more electronic systems.
- Some of the key reasons for why the CAN protocol was developed include −
- **High reliability:** The CAN protocol is designed to be robust and fault-tolerant, making it suitable for use in critical systems such as the engine control and braking systems in a car.
- **Low cost:** The CAN protocol uses a simple and efficient signaling method that allows for low-cost implementation, which is particularly important in the automotive industry where costs are a major concern.

- **Low weight and minimal wiring**: The CAN protocol uses a two-wire bus, which reduces the amount of wiring needed in a car and makes the vehicle lighter, which can lead to improved fuel efficiency.
- **Scalability:** The CAN protocol is designed to support a large number of devices on a network, making it easy to add new devices or remove existing ones as needed.
- **Multi-master capability**: The ability of the any device (node) to initiate communication in the network, this allows different systems to communicate and act upon their requirement and also enables distributed control which is a major feature of this protocol.

FRAMING

- In the Controller Area Network (CAN) protocol, a message is transmitted using a specific format called a frame.
- The frame consists of several fields that contain information about the message, such as the source address, destination address, and data payload.
- The basic format of a CAN frame includes −
- **Start of Frame (SOF)** − Identifies the start of a frame.
- **Identifier (ID)** − A unique 11 or 29-bit number that identifies the message.
- **Remote Transmission Request (RTR)** − Indicates whether the frame is a data frame (0) or a remote frame (1) requesting data.
- **Identifier Extension (IDE)** − Indicates whether the ID field is 11-bit (0) or 29-bit (1)

- **Data Length Code (DLC)** − Indicates the length of the data payload in bytes.
- **Data** − The payload of the message, which can be up to 8 bytes.
- **Cyclic Redundancy Check (CRC)** − A checksum used to detect errors in the frame.
- **Acknowledge Slot (ACK)** − Acknowledge of the message transmission by the receiver node.
- **End of Frame (EOF)** − Identifies the end of the frame.

SOF	**ID**	**RTR**	**IDE**	**DLC**	**CRC**	**ACK**	**EOF**

A message is transmitted as a series of bits on the bus, with the most significant bit (MSB) transmitted first. Once a message has been transmitted, all nodes on the network will receive the message, but only the node with the matching identifier will process it.

LAYERED ARCHITECTURE

The Controller Area Network (CAN) protocol has a layered architecture that is designed to separate the different responsibilities of the protocol.

The CAN protocol is typically divided into five layers −

- **Physical Layer:**
  - This layer is responsible for the physical transmission of bits over the communication medium, such as a cable or wireless link.
  - It defines the electrical, mechanical, and operational specifications for the interface between the nodes and the communication medium.
- **Data Link Layer:**
  - This layer is responsible for providing reliable data transfer between the nodes. It includes error detection and correction mechanisms, such as bit stuffing and cyclic redundancy check (CRC).
  - It also manages the arbitration process to ensure that only one node can transmit at a time and manages the Acknowledgement of the message.

- **Network Layer:**
  - This layer is responsible for providing a common communication format and addressing scheme for all nodes on the network.
  - It defines the format of the messages, including the identifier, data payload, and priority.
- **Transport Layer:**
  - This layer defines the rules for message transmission and reception, such as message fragmentation and retransmission, flow control, and error handling.
- **Application Layer:**
  - This layer defines the services and interfaces that are available to the application, such as sending and receiving messages, and monitoring the status of the network.
  - It also provides an interface for the Application layer.
- Each layer of the architecture is designed to be independent of the others, so that changes or improvements can be made to one layer without affecting the others.
- This design allows for flexibility and scalability of the protocol, which makes it a well suited for various applications across different industries.

APPLICATIONS

The Controller Area Network (CAN) protocol is widely used in a variety of applications, including −

- Automotive − The CAN protocol was originally developed for use in the automotive industry and is used in a wide variety of systems in modern cars, including engine control, transmission control, anti-lock brakes, and body electronics.
- Industrial automation − The CAN protocol is used in industrial automation systems to allow devices to communicate and coordinate their actions, such as controlling motors, sensors, and other equipment.

- Medical equipment − The CAN protocol is used in medical equipment to control various functions and to transmit data between devices. For example, patient monitoring systems use the protocol to transmit patient vital signs data between devices
- Avionics − The protocol is used in avionics to control and monitor various systems such as engine, navigation, and flight control systems.
- Building automation − The protocol is used in building automation systems to control and monitor various systems such as heating, ventilation, air conditioning (HVAC), lighting, and security systems.
- Robotics − The protocol is used in Robotics to control and monitor various systems such as motors, sensors, and other equipment, thus allowing the robots to communicate and coordinate their actions.

UNIVERSAL SERIAL BUS (USB)

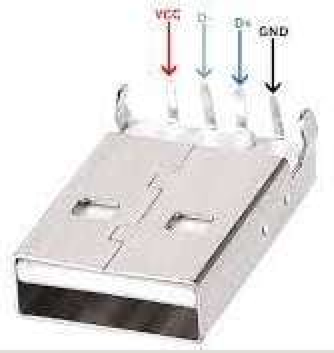
USB - INTRODUCTION

- The USB protocol or universal serial bus was first developed and launched by Ajay V.Bhatt from Intel in the year 1996.
- This USB is replaced different kinds of serial & parallel ports for transferring data in between a computer as well as different peripheral devices like scanners, printers, keyboards, gamepads, digital cameras, joysticks, etc.
- A common interface that is used to allow communication between different peripheral devices like mice, 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.

- A universal serial bus is intended to allow hot swapping & enhance plug-N- play.
- The plug-and-play allows the OS to configure and discover a new peripheral device spontaneously without starting the computer whereas hot swapping removes and replaces a new peripheral device without rebooting.
- 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

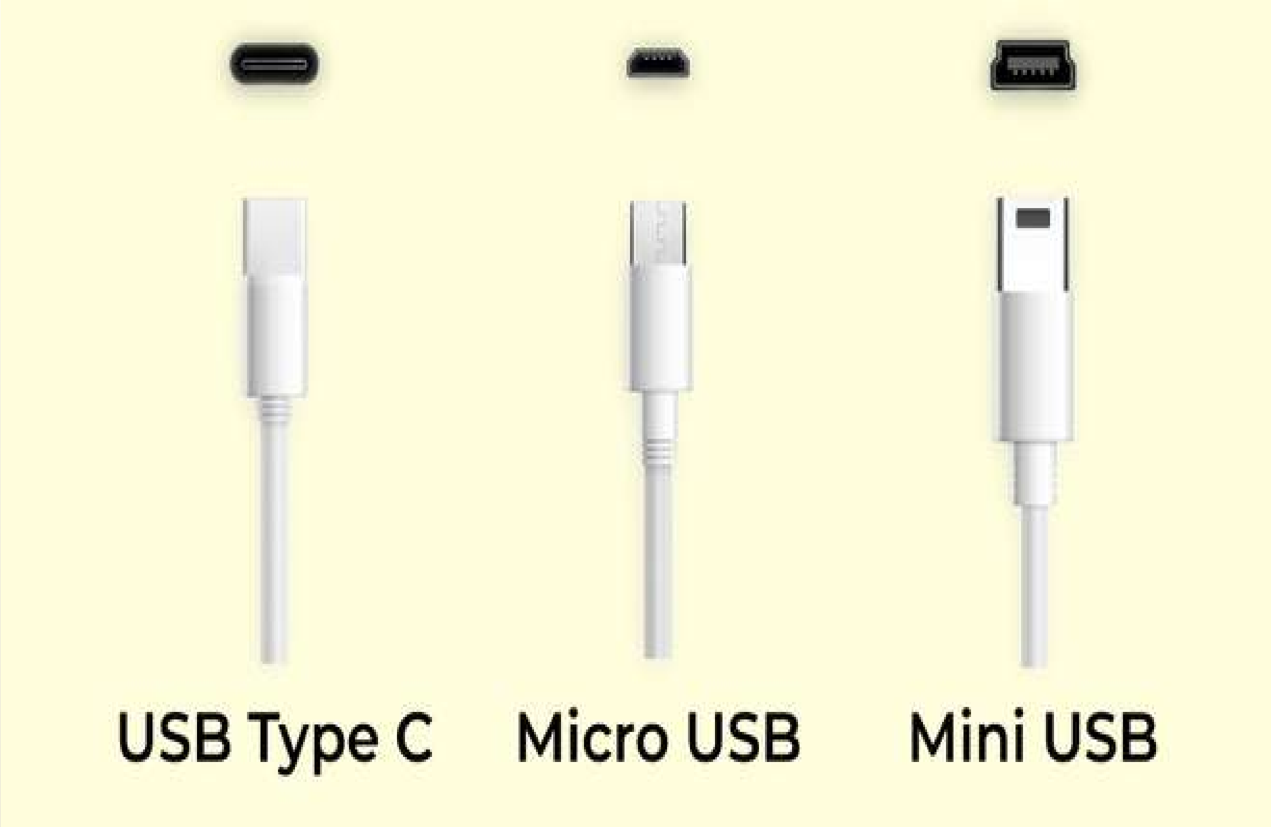
- The typical Type-A USB connector is used in various applications.
- These USBs include 4 pins that are given below.
- Pin1 (VBUS): It is a red color wire, used for providing power supply.
- Pin2 (D-): It is a differential pair pin available in white color, used for connectivity of USB.
- Pin3 (D+): 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.



- 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.
- 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.

USB CONNECTOR TYPES

- USB connectors have different shapes and sizes.
- Most of the USB connectors are
  - USB Type C
  - Mini-USB
  - Micro-USB
- These connectors have two or more variations.

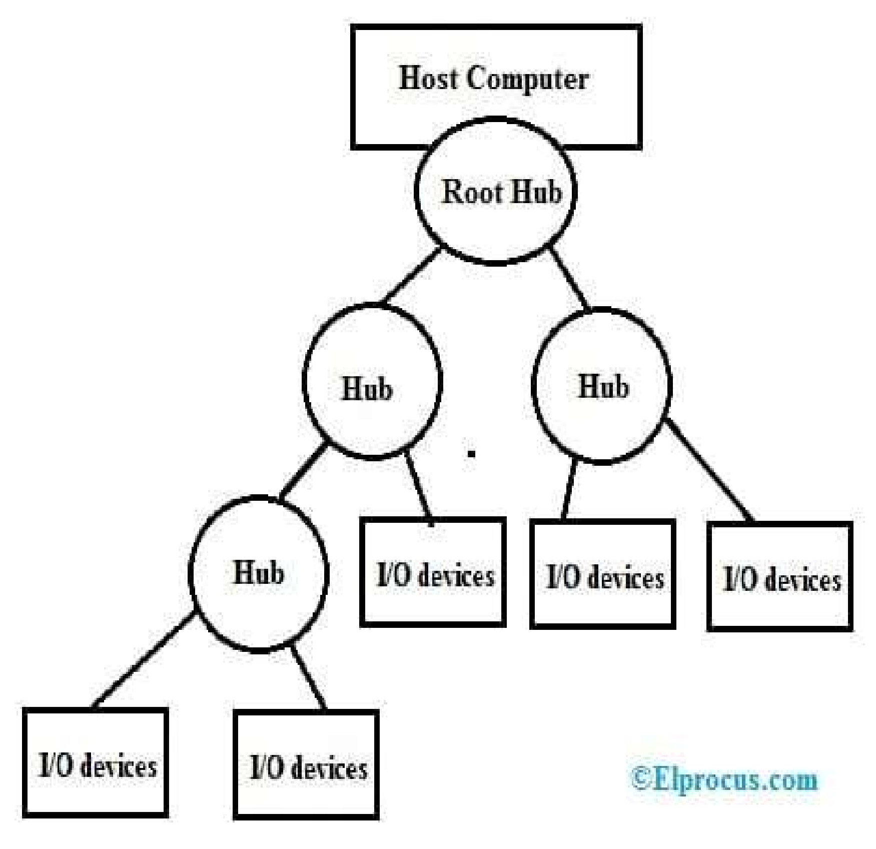


- **Mini USB**
  - Mini USB is available in three different types A type, B type, and AB type. It is used with computer peripherals and digital cameras.
  - It uses coaxial cable to transmit data and power between two devices. it applies to mobile hard drives, digital cameras, and MP3 players.
  - Although the tiny USB is mainly designed for, it can also be used to transfer data between computers having at least one USB port for charging device.
- **Micro USB**
  - A reduced version of the USB (Universal Serial Bus), is the micro-USB.
  - It was created for connecting small and mobile devices including digital cameras, smartphones, GPS components, MP3 players, and photo printers and was first announced in 2007 as a replacement for mini-USB.
  - The three different types of Micro-USB are Micro A, Micro B, and Micro USB 3. The connector size for the type Micro-A and Micro-B is 6.85 x 1.8 mm, while the Micro-A connector has a larger maximum overmild size.

- **USB Type-C**
  - A USB Type-C port is a relatively new type of connector that may be found on most contemporary newer Android smartphones and other USB-connected devices.
  - Data and power are delivered to computing machines using it.
  - In contrast to traditional USB connections, USB-C cables can be connected into devices in either direction, including upside down.

USB PROTOCOL ARCHITECTURE

- Once various I/O devices are connected through USB to the computer then they all are structured like a tree.
- In this USB structure, every I/O device will make a point-to-point connection to transmit data through the serial transmission format.
- In this architecture, I/O devices are connected to the computer through USB which is called as a hub.
- The Hub within the architecture is the connecting point between both the I/O devices as well as the computer.
- The root hub in this architecture is used to connect the whole structure to the hosting computer.
- The I/O devices in this architecture are a keyboard, mouse, speaker, camera, etc.

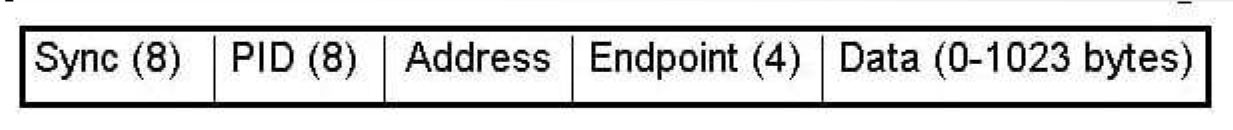


©Elprocus.com

WORKING OF USB PROTOCOL

- The USB protocol simply works on the polling principle.
- In polling, the processor continuously checks whether the input/output device is prepared for transmitting data or not.
- So, 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.
- Whenever a new device is allied to the hub then it is addressed like ‘0’.
- During a normal period, the host computer will poll the hubs to obtain their condition which allows the host to know the I/O devices from the system are attached or detached from the system.
- Once the host becomes responsive to the new device then it knows the device capacities by reading the available data within the memory of the USB interface of the device.
- So, the host uses a suitable driver to communicate with devices.
- Then, the host allocates an address to the new device which is written to the device register.

MESSAGE FORMAT



Sync (8)	PID (8)	Address	Endpoint (4)	Data (0-1023 bytes)

- The data of the USB protocol is transmitted within packets LSB first. There are mainly four types of USB packets Token, Data, Handshake & Start of the Frame. Every packet is designed from various field types which are shown in the following message format diagram.

- SYNC - Every USB packet will begin with a SYNC field which is normally utilized to synchronize the transmitter & the receiver to transmit the data precisely.
- PID - The packer identifier field within the USB protocol is mainly used to recognize the packet type that is being transmitted and thus the packet data format.
- Address - The address field of the USB protocol indicates which packet device is mainly designated for. The 7-bits length simply allows support of 127 devices.
- Endpoint - The endpoint field within the USB protocol is 4-bits long & allows for extra flexibility within addressing. Usually, these are divided for the data moving IN/OUT.
- Data - The length of the data field is not fixed, so it ranges from 0 to 8192 bits long & always an integral the number of bytes.
- CRC - The Cyclic Redundancy Checks (CRC) are executed on the data in the packet payload where all the token packets include 5-bit CRC & the data packets include a 16-bit CRC.
- EOP - Every packet is terminated by an EOP (End of the Packet) field which includes an SE0 or single-ended zero for 2-bit times followed through the J for 1-bit time.

INFRARED, BLUETOOTH, ZIGBEE PROTOCOLS

INFRARED

- Infrared (IR) communication is widely used in embedded systems for wireless, short-range communication.
- It is commonly seen in remote controls, proximity sensors, and some industrial automation systems.
- IR communication uses infrared light (wavelength: 850nm - 950nm) for data transmission.
- Typically, an IR LED (transmitter) sends modulated signals, and an IR receiver detects them.
- The most common modulation frequency is 38 kHz, as it helps reduce noise from ambient light.

- IR communication uses infrared light, which is a type of electromagnetic radiation with longer wavelengths than visible light but shorter than radio waves.
- IR communication requires a direct line of sight between the transmitter (IR LED) and the receiver (IR photodiode).
- IR is typically used for short-range applications, such as remote controls or data transfer between devices within a few meters.
- IR communication is a relatively inexpensive and widely adopted technology.

To implement IR communication in an embedded system, we typically use:

- **IR Transmitter (IR LED)**
- **IR Receiver Module (TSOP1738, VS1838, etc.)**
- **Microcontroller (e.g., Arduino, PIC, STM32, ESP32, etc.)**

Steps for Transmission:

1. Generate a **38 kHz** modulated signal using a **PWM output**.
2. Encode data using the selected protocol.
3. Transmit pulses via the **IR LED**.

Steps for Reception:

1. The IR receiver **demodulates** the 38 kHz signal.
2. The microcontroller **decodes** the received bit stream.
3. The system processes the command accordingly.

IR PROTOCOLS

- There are several standardized IR communication protocols, each with different encoding techniques.
  - IRDA
  - NEC
  - Sony SIRC
  - RC5 (Philips)
  - RC6

IRDA (InfraRed Data Association)

- A standard for wireless infrared communication, providing specifications for a complete set of protocols.
- Designed for wireless data transfer over short distances (e.g., "last one meter").
- Used in consumer electronics, automobiles, computers, medical devices, household appliances, and commercial services.
- IrDA-enabled devices can communicate bidirectionally.

NEC

- Widely used in remote controls.
- Uses pulse distance encoding.
- Each frame consists of:
  - Leader code (9ms HIGH + 4.5ms LOW)
  - Address (8-bit)
  - Inverted Address (8-bit)
  - Command (8-bit)
  - Inverted Command (8-bit)
  - Stop bit (562.5 µs)
  - Bit Encoding:
    - 0: 562.5 µs HIGH + 562.5 µs LOW
    - 1: 562.5 µs HIGH + 1.6875 ms LOW

Sony SIRC

- Used in Sony remote controls.
- Uses pulse-width modulation (PWM).
- Frames include:
  - Start bit (2.4ms HIGH)
  - 7-bit command
  - 5-bit address
  - Bit Encoding:
    - 0: 600 μs HIGH + 600 μs LOW
    - 1: 1.2ms HIGH + 600 μs LOW

RC5

- Uses Manchester encoding.
- Frame structure:
  - Start bits (2 bits)
  - Toggle bit (1 bit)
  - Address (5 bits)
  - Command (6 bits)
  - Bit Encoding:
    - 0: LOW → HIGH transition
    - 1: HIGH → LOW transition

RC6

- Advanced version of RC5,
- Supports higher bit rates.
- Uses bi-phase coding.
- Includes mode bits for different types of devices.

Other Protocols

- There are other protocols used in remote controls, each with its own encoding method and message structure.

APPLICATIONS OF IR

- Remote Control: Controlling devices like TVs, audio systems, and appliances.
- Data Transfer: Transferring data between devices, such as between a computer and a printer or a PDA and a computer.
- Industrial Automation: Monitoring and controlling industrial equipment.
- Consumer Electronics: Used in various consumer electronics devices, such as laptops, PDAs, and cellular phones.

ADVANTAGES OF IR

- Simplicity: IR communication is relatively simple to implement, even with slow microcontrollers, due to the low transmission speeds of popular IR protocols.
- Low Cost: IR components (IR LEDs and receivers) are inexpensive.
- Security: IR communication can be more secure than some other wireless technologies because it requires a direct line of sight.

CHALLENGES AND LIMITATIONS OF IR

- Line-of-sight required: Objects can obstruct signals.
- Limited range (~5m): Compared to RF technologies like Bluetooth.
- Interference from ambient light: Sunlight and fluorescent lights can disrupt IR signals.
- Lower data rate: Suitable for simple commands, not large data transfer.

BLUETOOTH

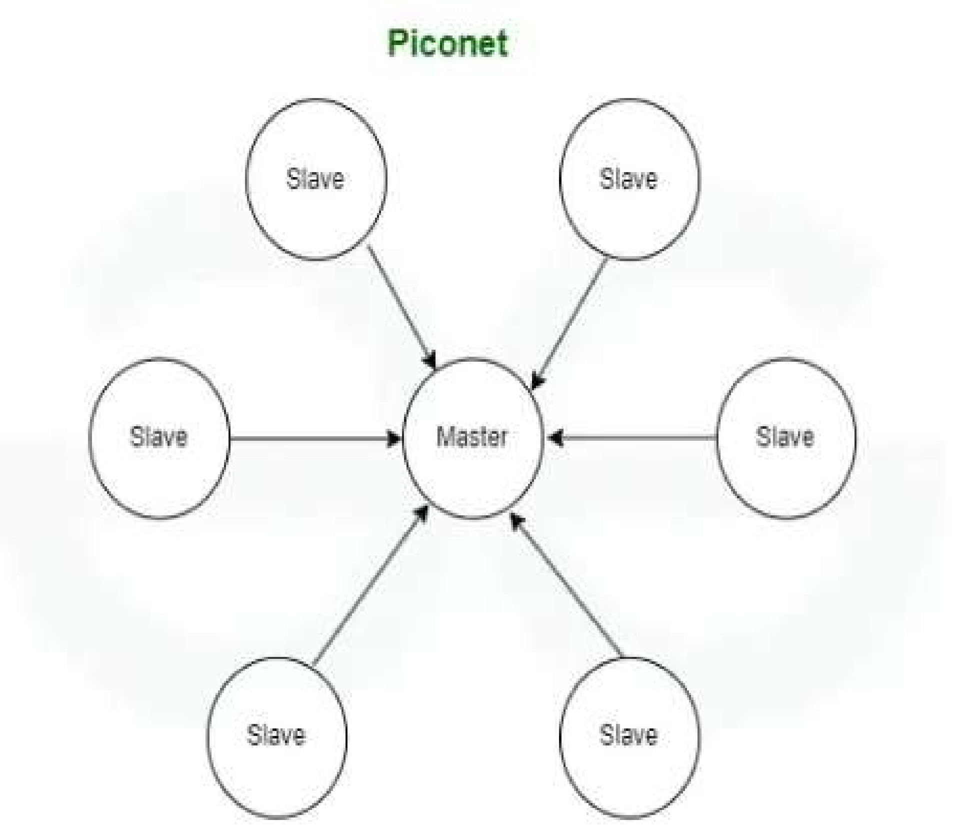
- Bluetooth is used for short-range wireless voice and data communication.
- It is a Wireless Personal Area Network (WPAN) technology and is used for data communications over smaller distances.
- This generation changed into being invented via Ericson in 1994.
- It operates within the unlicensed, business, scientific, and clinical (ISM) bands from 2.4 GHz to 2.485 GHz.
- It makes devices like phones, tablets, and headphones connect to each other and share information without needing cables.
- 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.
- 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 scatter net.

ARCHITECTURE OF BLUETOOTH

The architecture of Bluetooth defines two types of networks: Piconet and Scatternet.

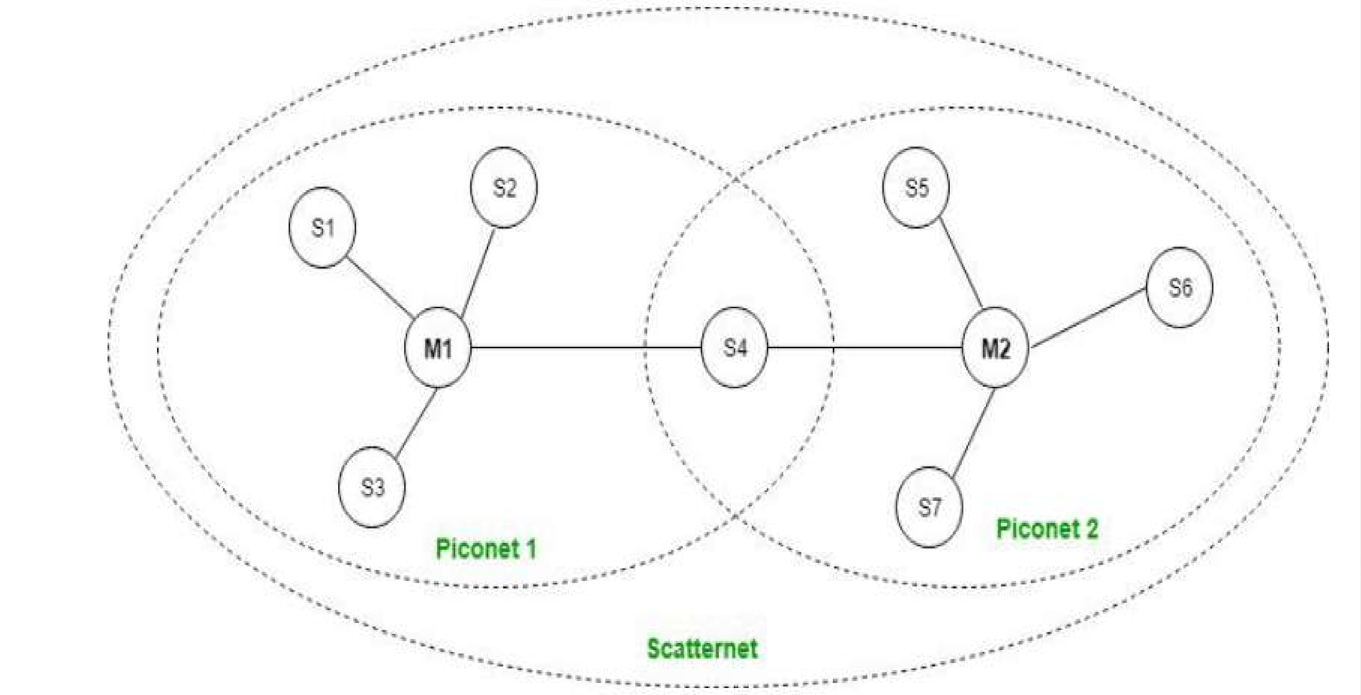
Piconet:

- Piconet is a type of Bluetooth network that contains one primary node called the master node and seven active secondary nodes called slave nodes, i.e., there is a total of 8 active nodes which are present at 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.
- It also has 255 parked nodes; these are secondary nodes and cannot take participation in communication unless it gets converted to the active state.

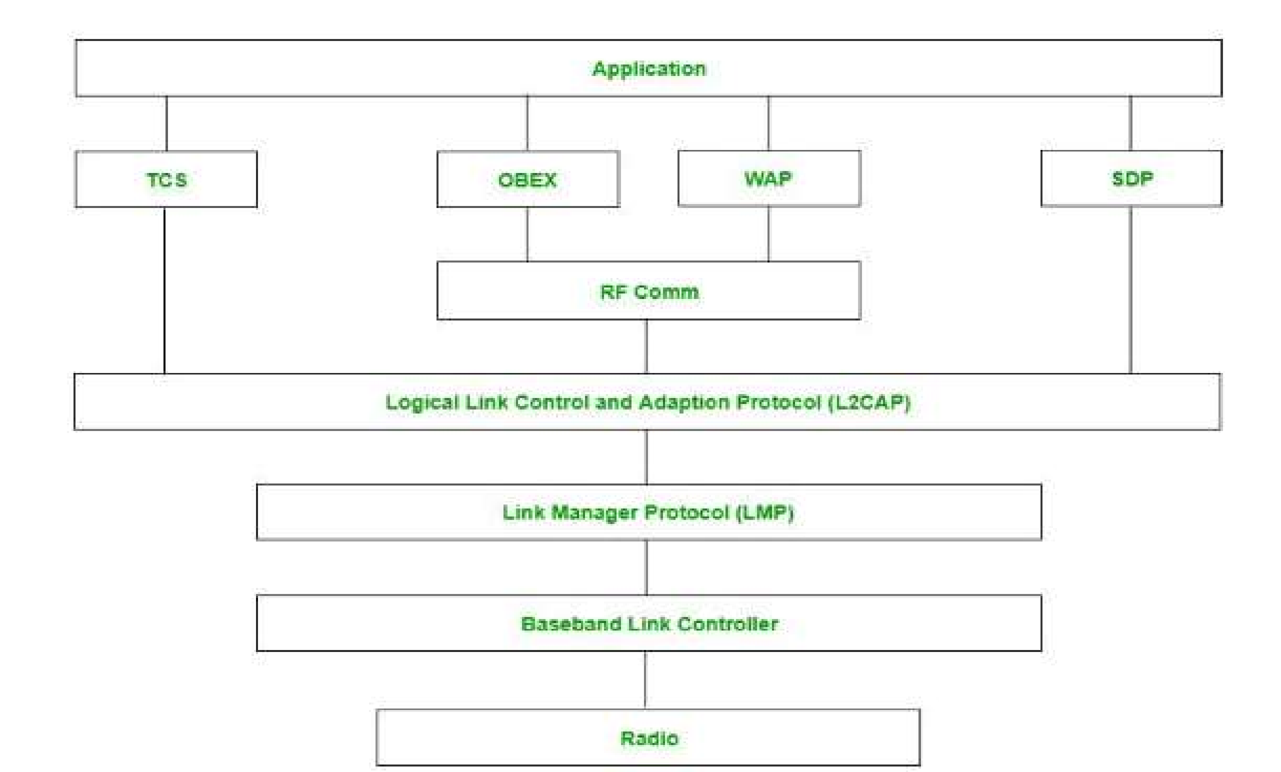


Scatternet

- 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 node cannot be mastered in two piconets.



BLUETOOTH PROTOCOL STACK


Application
TCS
OBEX
WAP
SDP
RF Comm
Logical Link Control and Adaption Protocol (L2CAP)
Link Manager Protocol (LMP)
Baseband Link Controller
Radio

Bluetooth Protocol Stack

- Bluetooth communication is structured into multiple layers:
  - Radio (RF) Layer
  - Baseband Layer
  - Link Manager Protocol Layer
  - Logical Link Control and Adaption Protocol (L2CAP) Layer
  - Service Discovery Protocol (SDP) Layer
  - RF Comm Layer
  - OBEX
  - WAP
  - TCS
  - Application Layer

Radio (RF) Layer:

- It specifies the details of the air interface, including frequency, the use of frequency hopping and transmit power.
- It performs modulation/demodulation of the data into RF signals. It defines the physical characteristics of Bluetooth transceivers.
- It defines two types of physical links: connection-less and connection-oriented.

Baseband Link Layer:

- The baseband is the digital engine of a Bluetooth system and is equivalent to the MAC sublayer in LANs.
- It performs the connection establishment within a piconet, addressing, packet format, timing and power control.

Link Manager Protocol Layer:

- It performs the management of the already established links which includes authentication and encryption processes.
- It is responsible for creating the links, monitoring their health, and terminating them gracefully upon command or failure.

Logical Link Control and Adaption (L2CAP) Protocol Layer:

- It is also known as the heart of the Bluetooth protocol stack.
- It allows the communication between upper and lower layers of the Bluetooth protocol stack.
- It packages the data packets received from upper layers into the form expected by lower layers. It also performs segmentation and multiplexing.

Service Discovery Protocol (SDP) Layer:

- It is short for Service Discovery Protocol. It allows discovering the services available on another Bluetooth-enabled device.

RF Comm Layer:

- It is a cabal replacement protocol. It is short for Radio Frontend Component.
- It provides a serial interface with WAP and OBEX.
- It also provides emulation of serial ports over the logical link control and adaption protocol(L2CAP).
- The protocol is based on the ETSI standard TS 07.10.

OBEX:

- It is short for Object Exchange. It is a communication protocol to exchange objects between 2 devices.

WAP:

- It is short for Wireless Access Protocol. It is used for internet access.

TCS:

- It is short for Telephony Control Protocol. It provides telephony service.
- The basic function of this layer is call control (setup & release) and group management for the gateway serving multiple devices.

Application Layer:

- It enables the user to interact with the application.

APPLICATIONS OF BLUETOOTH

- It can be used in wireless headsets, wireless PANs, and LANs.
- It can connect a digital camera wireless to a mobile phone.
- It can transfer data in terms of videos, songs, photographs, or files from one cell phone to another cell phone or computer.
- It is used in the sectors of Medical healthcare, sports and fitness, Military.

ADVANTAGES AND DISADVANTAGES

Advantages

- 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

- 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 PROTOCOL

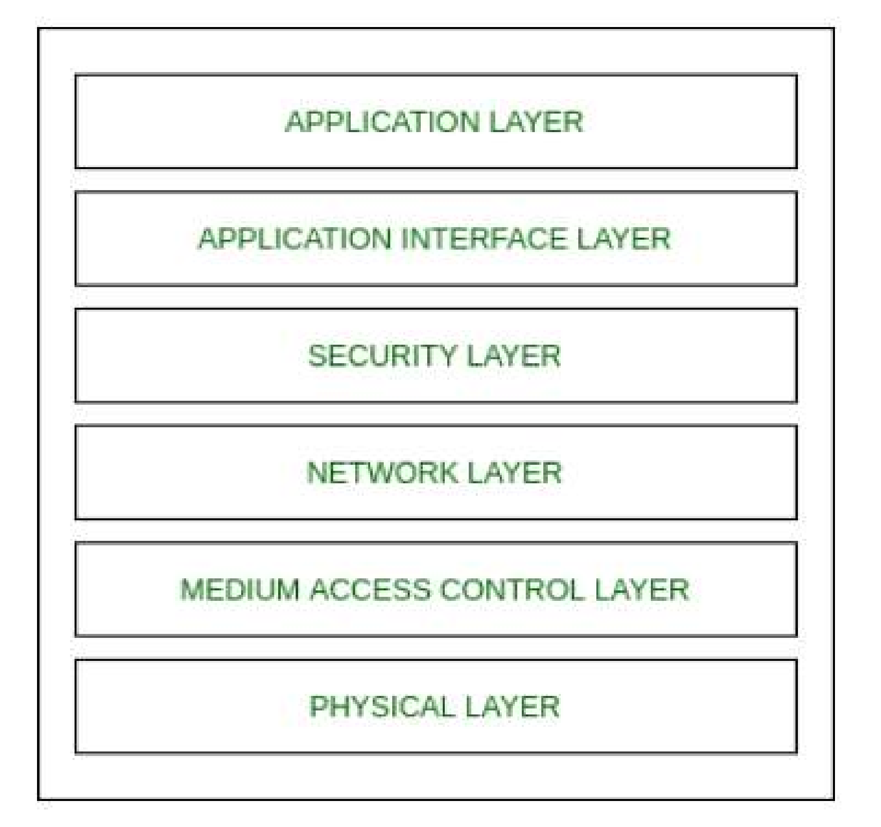
- ZigBee is a Personal Area Network of task group 4, so, it is based on IEEE 802.15.4.
- It is a technology of home networking. ZigBee is a technological standard created for controlling and sensing the network.
- It is created by Zigbee Alliance.
- ZigBee is an open, global, packet-based protocol designed to provide an easy-to-use architecture for secure, reliable, low power wireless networks.
- Flow or process control equipment can be placed anywhere and still communicate with the rest of the system.

- It can also be moved, since the network doesn’t care about the physical location of a sensor, pump or valve.
- 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.

ARCHITECTURE OF ZIGBEE

Zigbee architecture is a combination of 6 layers.

- Application Layer
- Application Interface Layer
- Security Layer
- Network Layer
- Medium Access Control Layer
- Physical Layer



APPLICATION LAYER

APPLICATION INTERFACE LAYER

SECURITY LAYER

NETWORK LAYER

MEDIUM ACCESS CONTROL LAYER

PHYSICAL LAYER

Physical layer:

- The lowest two layers, i.e., the physical and the MAC (Medium Access Control) Layer are defined by the IEEE 802.15.4 specifications.
- The Physical layer is closest to the hardware and directly controls and communicates with the Zigbee radio.
- The physical layer translates the data packets in the over-the-air bits for transmission and vice-versa during the reception.

Medium Access Control layer (MAC layer):

- The layer is responsible for the interface between the physical and network layer.
- The MAC layer is also responsible for providing PAN ID and network discovery through beacon requests.

Network layer:

- This layer acts as an interface between the MAC layer and the application layer. It is responsible for mesh networking.

Application layer:

- The application layer in the Zigbee stack is the highest protocol layer and it consists of the application support sub-layer and Zigbee device object.
- It contains manufacturer-defined applications.

FEATURES OF ZIGBEE

- **Stochastic addressing:** A device is assigned a random address and announced. Mechanism for address conflict resolution. Parents node don’t need to maintain assigned address table.
- **Link Management:** Each node maintains quality of links to neighbors. Link quality is used as link cost in routing.
- **Frequency Agility:** Nodes experience interference report to channel manager, which then selects another channel
- **Asymmetric Link:** Each node has different transmit power and sensitivity. Paths may be asymmetric.
- **Power Management:** Routers and Coordinators use main power. End Devices use batteries.

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.

APPLICATIONS

- Home Automation
- Medical Data Collection
- Industrial Control Systems
- meter reading system
- light control system
- Commercial
- Government Markets Worldwide
- Home Networking

ADVANTAGES OF ZIGBEE

- Designed for low power consumption.
- Provides network security and application support services operating on the top of IEEE.
- Zigbee makes possible completely networks homes where all devices are able to communicate and be
- Use in smart home
- Easy implementation
- Adequate security features.
- Low cost: Zigbee chips and modules are relatively inexpensive, which makes it a cost-effective solution for IoT applications.
- Mesh networking: Zigbee uses a mesh network topology, which allows for devices to communicate with each other without the need for a central hub or router. This makes it ideal for use in smart home applications where devices need to communicate with each other and with a central control hub.
- Reliability: Zigbee protocol is designed to be highly reliable, with robust mechanisms in place to ensure that data is delivered reliably even in adverse conditions.

DISADVANTAGES OF ZIGBEE

- Limited range: Zigbee has a relatively short range compared to other wireless communications protocols, which can make it less suitable for certain types of applications or for use in large buildings.
- Limited data rate: Zigbee is designed for low-data-rate applications, which can make it less suitable for applications that require high-speed data transfer.
- Interoperability: Zigbee is not as widely adopted as other IoT protocols, which can make it difficult to find devices that are compatible with each other.
- Security: Zigbee’s security features are not as robust as other IoT protocols, making it more vulnerable to hacking and other security threats.

WIRELESS COMMUNICATION PROTOCOLS – GSM AND GPRS

GSM

- GSM stands for Global System for Mobile Communication. It is a digital mobile network commonly utilized by cell phone users around the world.
- It is the most popular of the three digital wireless telephony systems (TDMA, GSM, and CDMA) and uses the combination of FDMA and TDMA.
- It uses 4 different frequency bands 850 MHz, 900 MHz, 1800 MHz, and 1900 MHz.
- GSM converts and compresses data before sending it along a channel with two other streams of user data, each with its time slot.
- GSM have 4 different sizes of cells
  - Macro: In this size of the cell, a Base Station antenna is installed.
  - Micro: In this size of cell, antenna height is less than the average roof level.
  - Pico: Small cells' diameter of a few meters.
  - Umbrella: It covers the shadowed (Fills the gaps between cells) regions.

ARCHITECTURE OF GSM

The GSM architecture is made up of three central systems. The following are the primary components of the GSM architecture:

- The network switching system (NSS)
- The base station system (BSS)
- The operations and support system (OSS)

NETWORK SWITCHING SYSTEM (NSS)

NSS is a GSM element that provides flow management and call processing for mobile devices moving between base stations. The switching system consists of the functional units listed below.

- Mobile Services Switching Center (MSC): Mobile Switching Center is integral to the GSM network architecture’s central network space. The MSC supports call switching across cellular phones and other fixed or mobile network users.
- Home Location Register (HLR): It is a set of data items used for storing and managing subscriptions. It provides data for each consumer as well as their last known position. The HLR is regarded as the most significant database because it preserves enduring records about users.

- Visitor Location Register (VLR): VLR is a database that provides subscriber information necessary for the MSC to service passengers. This includes a short-term version of most of the data stored in the HLR.
- Equipment Identity Register (EIR): It is the component that determines if one can use mobile equipment on the system. This consists of a list of every functioning mobile device on the system, with each mobile device recognized by its own International Mobile Equipment Identity (IMEI) number.
- Authentication Center (AuC): The AuC is a unit that offers verification and encryption factors to ensure the user’s identity and the privacy of every call. The verification center is a secure file that contains the user’s private key in the SIM card.

BASE STATION SYSTEM (BSS)

It serves as a connection between the network subsystem and the mobile station. It consists of two parts:

- The Base Transceiver Station (BTS): The BTS is responsible for radio connection protocols with the MS and contains the cell’s radio transceivers. Companies may implement a significant number of BTSs in a big metropolitan area. Each network cell has transceivers and antennas that make up the BTS.
- The Base Station Controller (BSC): The BSC is responsible for managing the radio resources of one or more BTS(s). This manages radio channel configuration and handovers.

OPERATIONS AND SUPPORT SYSTEM (OSS)

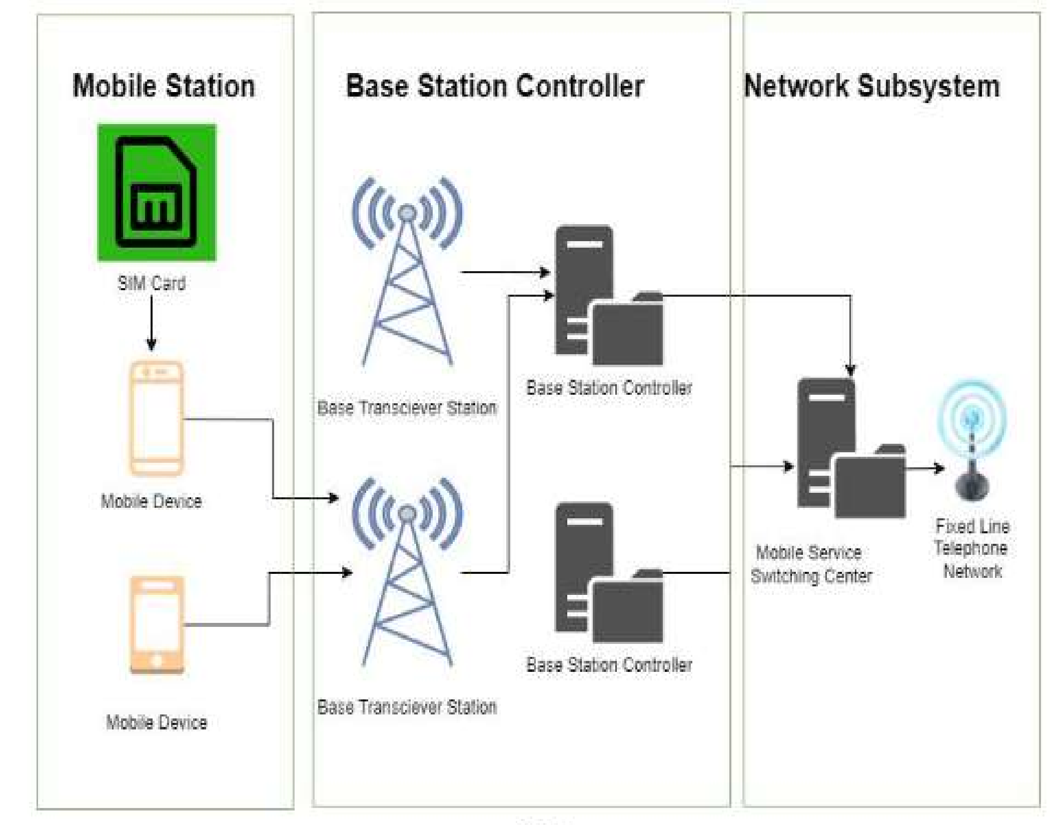
- The operation support system (OSS) is a part of the overall GSM network design.
- This is linked to the NSS and BSC components. The OSS primarily manages the GSM network and BSS traffic load.
- As the number of BS increases due to customer population scaling, a few maintenance duties are shifted to the base transceiver stations, lowering the system's financial responsibility.
- The essential purpose of OSS is to have a network synopsis and assist various services and maintenance organizations with their routine maintenance arrangements.

MOBILE STATION(MS)

- The mobile station is a cell phone with a display, digital signal processor, and radio transceiver regulated by a SIM card that functions on a system.
- Hardware and the SIM card are the two most essential elements of the MS.
- The MS (Mobile stations) is most widely recognized by cell phones, which are components of a GSM mobile communications network that the operator monitors and works.

WORKING OF GSM

- GSM describe the procedures for second-generation digital mobile networks, such as those used by cell phones.
- It is a broad-area communications technology program that uses digital radio channeling to provide audio, information, and multimedia communication systems.
- Every GSM radio channel is 200 kHz broad and is further divided into frames of eight time slots.
- The GSM system consists of mobile stations, base stations, and interweaving switching systems.
- The GSM program allows 8 to 16 audio users to share a single radio channel, and each radio transmission station can have numerous radio channels.
- Because of its simplicity, cost, and accessibility, GSM is now the most often utilized network technology in the Internet of Things (IoT).



GSM

APPLICATIONS, ADVANTAGES AND DISADVANTAGES OF GSM

Applications of GSM

- Mobile Telephony
- VoIP Integration
- SMS (Short Message Service)
- Mobile Banking
- Smart Home Systems
- Surveillance Systems
- Alarm System
- Cell Broadcasting

Advantages of GSM

- Compatibility: GSM is widely used around the world, so it is compatible with many different networks and devices.
- Security: GSM offers enhanced security features such as authentication, encryption and confidentiality which helps to protect the user’s privacy and data.
- Efficient use of bandwidth: GSM uses a time-division multiplexing (TDM) technique which enables many users to share the same frequency channel at different times, making it an efficient use of the available bandwidth.
- Roaming: GSM allows users to roam internationally and use their cell phones in other countries that use the same GSM standard.
- Wide range of features: GSM supports a wide range of features, including call forwarding, call waiting, voicemail, conference calling, and more.

Disadvantages of GSM

- Limited coverage: GSM networks may have limited coverage in some remote areas, which can make it difficult for users to make calls or access the internet.
- Network congestion: GSM networks may become congested during peak hours, which can lead to dropped calls or poor call quality.
- Security vulnerabilities: Although GSM offers enhanced security features, it is still vulnerable to certain types of attacks, such as eavesdropping and spoofing.
- Data transfer speed: GSM networks offer relatively slow data transfer speeds compared to newer technologies such as 3G and 4G.
- Limited capacity: GSM networks have a limited capacity for handling large volumes of data, which can be a disadvantage for users who require high-speed internet access or other data-intensive applications.

GPRS

- GPRS is an expansion Global System for Mobile Communication.
- It is basically a packet-oriented mobile data standard on the 2G and 3G cellular communication network's global system for mobile communication.
- GPRS was built up by European Telecommunications Standards Institute (ETSI) because of the prior CDPD, and I-mode packet switched cell advances.
- GPRS overrides the wired associations, as this framework has streamlined access to the packet information's network like the web.
- The packet radio standard is utilized by GPRS to transport client information packets in a structured route between GSM versatile stations and external packet information networks.
- These packets can be straightforwardly directed to the packet changed systems from the GPRS portable stations.

ARCHITECTURE OF GPRS

GPRS tries to make maximum use of the existing physical structure of GSM. It has introduced a new entity named GPRS support nodes(GSN) whose responsibility is to route and deliver a data packet. GSN is of two types:

- Serving GPRS Support Node (SGSN)
- Gateway GPRS Support Node (GGSN)

Components of GPRS architecture:

- Mobile Station
- Base Station Controller
- GPRS Support Nodes
- Internal Back-bone Network
- Mobility Support
- Routing Area
- SMS in GSM

MOBILE STATION(MS)

- GPRS requires enhanced mobile stations, as existing mobile stations were designed according to the GSM network, and they were unable in handling enhanced data packets.
- A variety of high-speed mobile stations are available to support enhanced data packets.
- These mobile stations are also capable of handling the GSM architecture to make voice calls.

BASE STATION CONTROLLER (BSC)

- In GPRS there is one component is added to BSC called PCU. PCU stands for Packet Control Unit.
- If the signal comes to BSC and that signal contains data, then PCU routes to the SGSN.
- The interface is used between BSC and PCU is the FRI interface.
- After the signal comes to SGSN, it delivers the data packet to the GGSN.
- GGSN routes the data packet to the data network (PDN- Predefined Data Network).

GPRS SUPPORT NODES

GPRS support nodes are of two types:

(a) Serving GPRS Support Node (SGSN): It is responsible for the following tasks:

Packet Delivery

Mobility management

apply/ sign-off of terminals

localization

LLC (Logical Link Control) management

Authentication

Billing

(b) Gateway GPRS Support Node (GGSN): It is responsible for the following tasks:

Mediator between GPRS between backbone and external data networks.

Saves current data for the SGSN address of the participant as well as their profile and data for authentication and invoice.

Internal Backbone Network: It is an IP-based network that is used to support the working of GPRS and is responsible to carry new packets between different GSNs.

The tunneling is used between SGSNs and GGSNs to exchange information without informing the internal backbone.

Mobility Support: GPRS has the following mechanism to support mobility in the network:

- Attachment Procedure
- Location and Handoff Management

Routing Area: This is like the location area in GSM the only difference is routing area use fewer cells as routing areas are smaller than the location area.

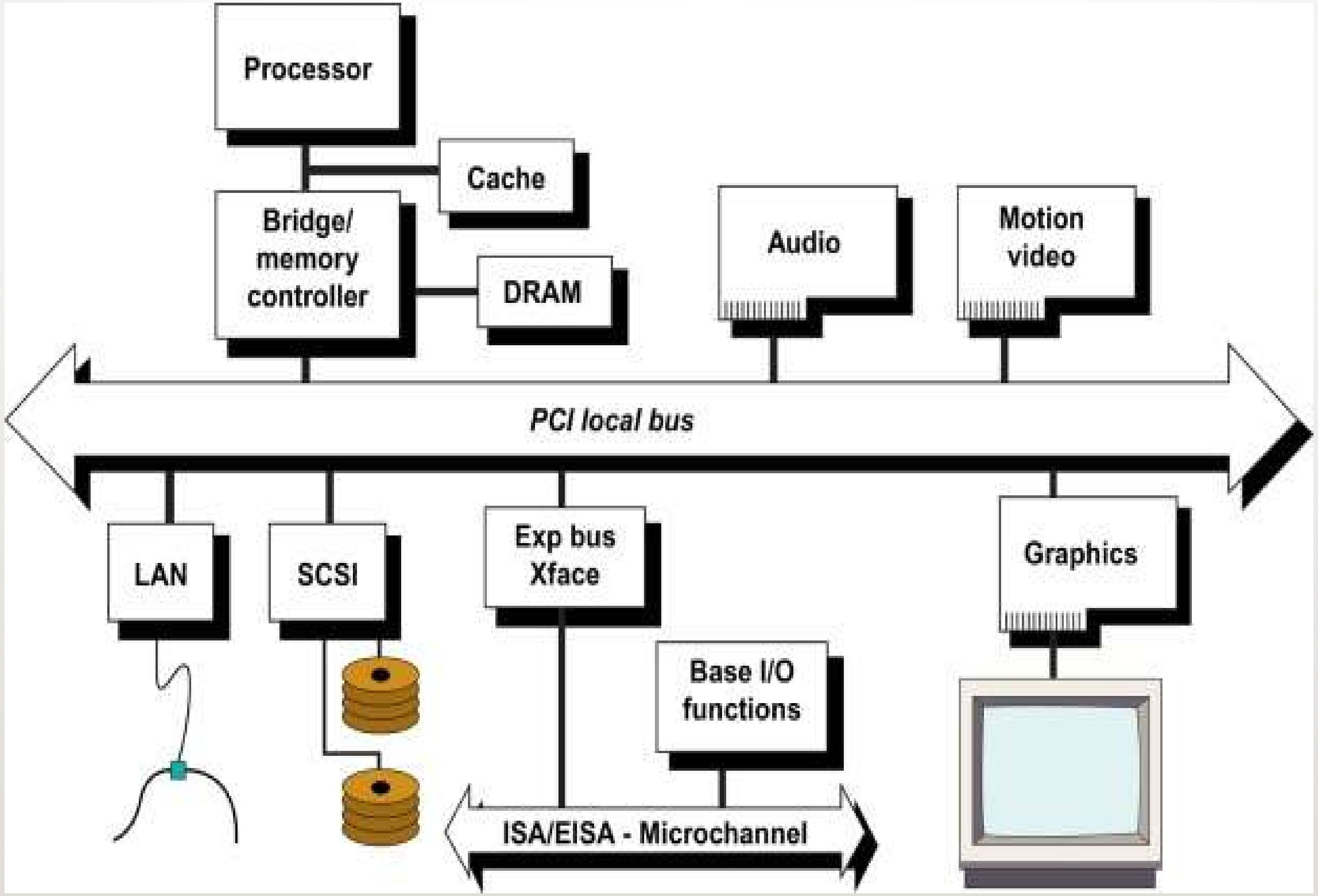
SMS in GSM: GSM introduced a mechanism of Short Messaging Service(SMS) which is similar to peer-to-peer Instant messaging.

BENEFITS OF GPRS

- Mobility: The capacity to keep up consistent voice and information interchanges while moving.
- Cost Efficient: Communication via GPRS is cheaper than through the regular GSM network.
- Immediacy: Allows customers to obtain connectivity when needed, regardless of location and without a lengthy login session.
- Localization: Enables customers to acquire data applicable to their present area.
- Easy Billing: GPRS packet transmission offers an easier to use billing than that offered by circuit switched administrations.

PERIPHERAL COMPONENT INTERCONNECT (PCI)

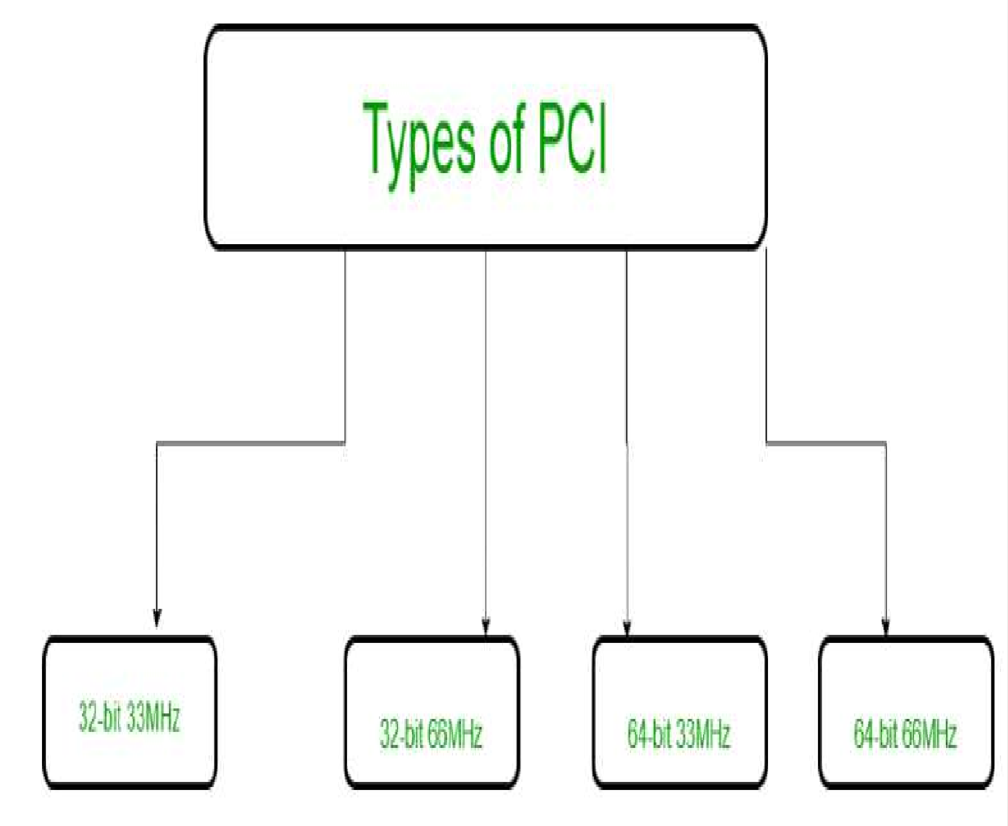
- The Peripheral Component Interconnect (PCI) protocol is a standardized, parallel bus used for connecting peripherals to the motherboard, offering high-speed data transfer and allowing devices to access system memory directly.
- It is a local computer bus standard (or a type of expansion bus) that allows hardware devices (peripherals) to connect to the motherboard.
- It uses a parallel communication protocol, meaning data is transferred across multiple wires simultaneously.
- It provides a high-speed data highway between the motherboard and various hardware components, facilitating efficient communication with peripherals like graphics cards, network adapters, and storage controllers.
- It is developed by Intel as a processor-independent, high-speed replacement


Processor
Cache
Bridge/ memory controller
DRAM
Audio
Motion video
PCI local bus
LAN
SCSI
Exp bus Xface
Graphics
Base I/O functions
ISA/EISA - Microchannel

TYPES OF PCI

There are 4 types of PCI with different speed.

- PCI 32 bits have a transport speed of 33 MHz and work at 132 MBps.
- PCI 64 bits have a transport speed of 33 MHz and work at 264 MBps.
- PCI 32 bits have a transport speed of 66 MHz and work at 512 MBps.
- PCI 64 bits have a transport speed of 66 MHz and work at 1 GBps.



FUNCTIONS OF PCI

- PCI slots are utilized to install sound cards, Ethernet and remote cards and presently strong state drives utilizing NVMe innovation to supply SSD drive speeds that are numerous times speedier than SATA SSD speeds.
- PCI openings too permit discrete design cards to be included to a computer as well.
- PCI openings (and their variations) permit to include expansion cards to a motherboard.
- The extension cards increment the machine capabilities which were created by the motherboard alone, such as: upgraded illustrations, extended sound, expanded USB and difficult drive controller, and extra arrange interface options.

ADVANTAGES OF PCI

- A greatest of five components can be interfaced to the PCI and can be supplant each of them by settled gadgets on the motherboard.
- There are different PCI buses on the same computer.
- The PCI transport will improve the speed of the exchanges from 33MHz to 133 MHz with a transfer rate of 1 gigabyte per second.
- The PCI can handle gadgets employing a greatest of 5 volts and the pins utilized can exchange more than one flag through one stick.

DISADVANTAGES OF PCI

PCI Graphics Card cannot get to system memory.

PCI does not support pipeline.

DMA

DIRECT MEMORY ACCESS

DIRECT MEMORY ACCESS (DMA)

- The processor copies each piece of data from the source to the destination in order to transfer the data.
- This is typically slower than copying normal blocks of memory since access to I/O devices over a peripheral bus is generally slower than normal system RAM.
- During this time, the processor would be unavailable for any other tasks involving processor bus access.
- But it can continue to work on any work which does not require bus access.
- Direct Memory Access (DMA) allows devices to transfer data without subjecting the processor a heavy overhead.
- DMA transfers are essential for high performance embedded systems where large chunks of data need to be transferred from the input/output devices to or from the primary memory.

DMA CONTROLLER

A DMA controller is a device, usually peripheral to a CPU that is programmed to perform a sequence of data transfers on behalf of the CPU.

A DMA controller can directly access memory and is used to transfer data from one memory location to another, or from an I/O device to memory and vice versa.

A DMA controller manages several DMA channels, each of which can be programmed to perform a sequence of these DMA transfers.

Devices, usually I/O peripherals, that acquire data that must be read or devices that must output data and must be written to signal, the DMA controller performs a DMA transfer by asserting a hardware DMA request (DRQ) signal.

DMA CONTROLLER

- A DMA request signal for each channel is routed to the DMA controller. This signal is monitored and responded to in much the same way that a processor handles interrupts.
- When the DMA controller sees a DMA request, it responds by performing one or many data transfers from that I/O device into system memory or vice versa.
- Channels must be enabled by the processor for the DMA controller to respond to DMA requests.
- The number of transfers performed, transfer modes used, and memory locations accessed depends on how the DMA channel is programmed.
- A DMA controller typically shares the system memory and I/O bus with the CPU and has both bus master and slave capability

OPERATION OF DMA

- In bus master mode, the DMA controller acquires the system bus (address, data, and control lines) from the CPU to perform the DMA transfers.
- Because the CPU releases the system bus for the duration of the transfer, the process is sometimes referred to as cycle stealing.
- In bus slave mode, the DMA controller is accessed by the CPU, which programs the DMA controller's internal registers to set up DMA transfers.
- The internal registers consist of source and destination address registers and transfer count registers for each DMA channel, as well as control and status registers for initiating, monitoring, and sustaining the operation of the DMA controller.

DMA TRANSFER TYPES AND MODES

- DMA controllers vary as to the type of DMA transfers and the number of DMA channels they support.
- The two types of DMA transfers are **flyby DMA transfers and fetch-and-deposit DMA transfers.**
- The three common transfer modes are **single, block, and demand transfer modes.**
- The fastest DMA transfer type is referred to as a single-cycle, single-address, or flyby transfer.

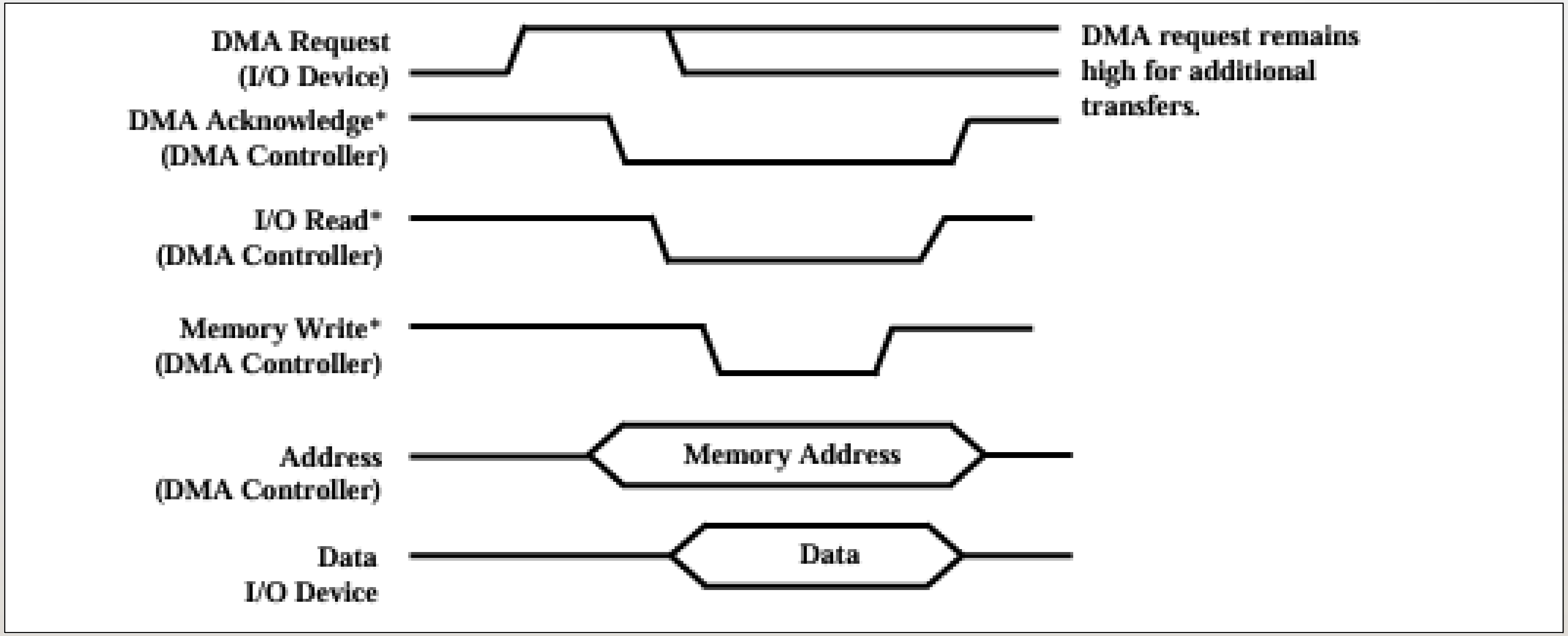
FLYBY DMA TRANSFER

- In a flyby DMA transfer, a single bus operation is used to accomplish the transfer, with data read from the source and written to the destination simultaneously.
- In flyby operation, the device requesting service asserts a DMA request on the appropriate channel request line of the DMA controller.
- The DMA controller responds by gaining control of the system bus from the CPU and then issuing the pre-programmed memory address.
- Simultaneously, the DMA controller sends a DMA acknowledge signal to the requesting device.
- This signal alerts the requesting device to drive the data onto the system data bus or to latch the data from the system bus, depending on the direction of the transfer.

FLYBY OPERATION

- In other words, a flyby DMA transfer looks like a memory read or write cycle with the DMA controller supplying the address and the I/O device reading or writing the data.
- Because flyby DMA transfers involve a single memory cycle per data transfer, these transfers are very efficient. Fig. shows the flyby DMA transfer signal protocol.

FLYBY DMA TRANSFER



FETCH-AND-DEPOSIT DMA TRANSFER

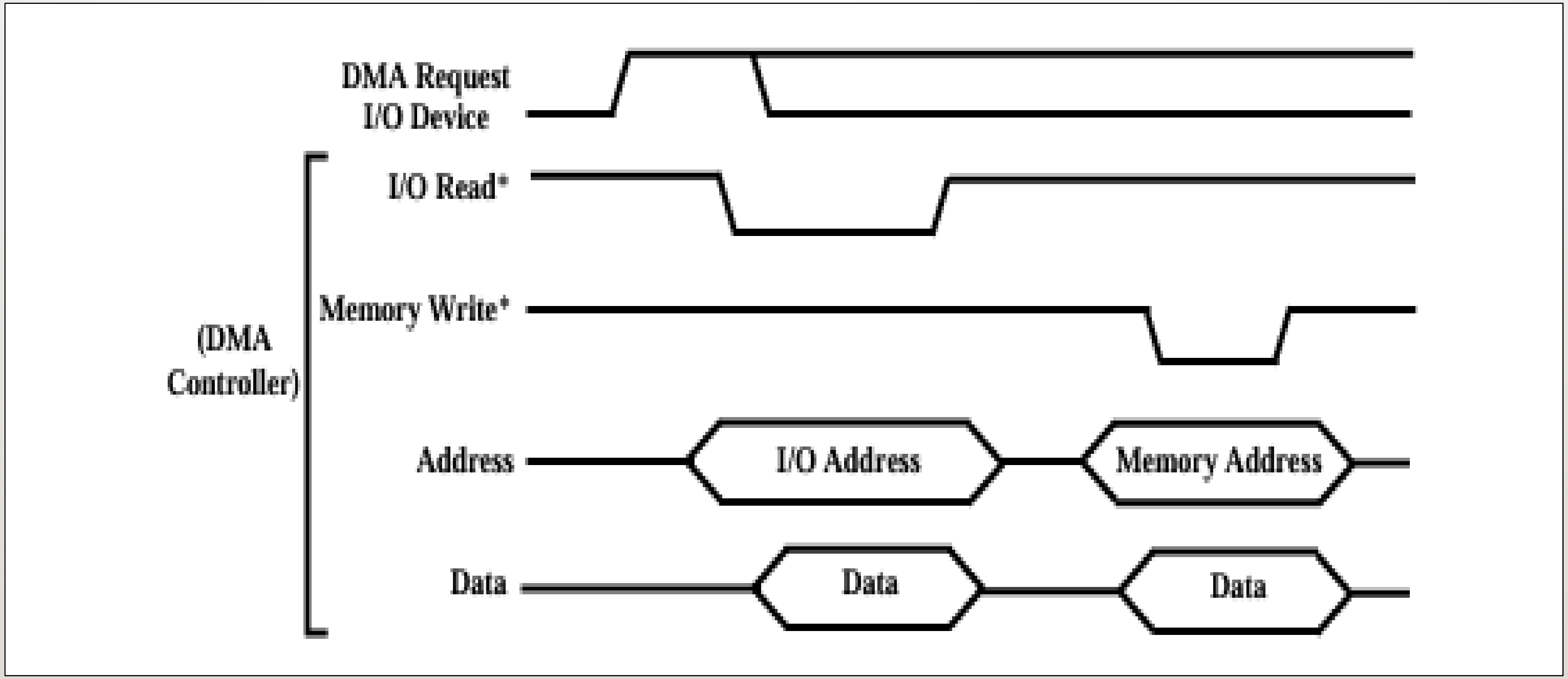
- This type of transfer involves two memory or I/O cycles.
- The data being transferred is first read from the I/O device or memory into a temporary data register internal to the DMA controller.
- The data is then written to the memory or I/O device in the next cycle.
- The DMA controller performs two cycles and thus retains the system bus longer.
- This type of transfer is useful for interfacing devices with different data bus sizes.

For example, a DMA controller can perform two 16-bit read operations from one location followed by a 32-bit write operation to another location.

A DMA controller supporting this type of transfer has –

- Two address registers per channel (source address and destination address).
- Bus-size registers, in addition to the usual transfer count and control registers.
- This type of DMA transfer is suitable for both memory-to-memory and I/O transfers.

FETCH-AND-DEPOSIT DMA TRANSFER

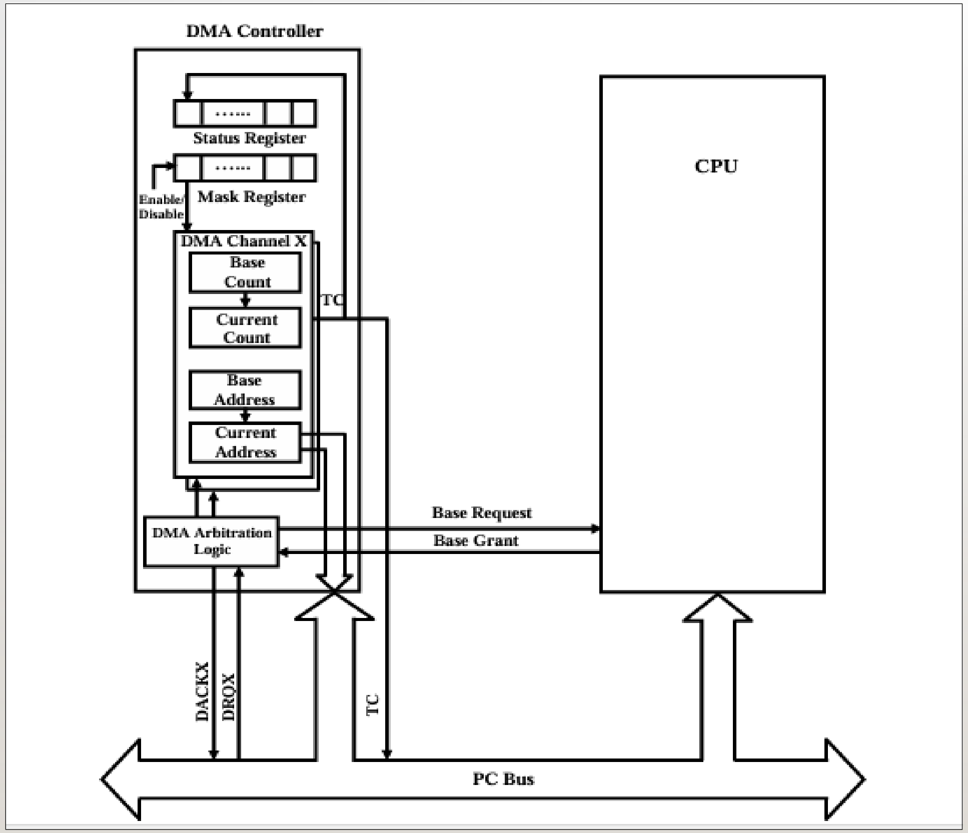


TRANSFER MODES

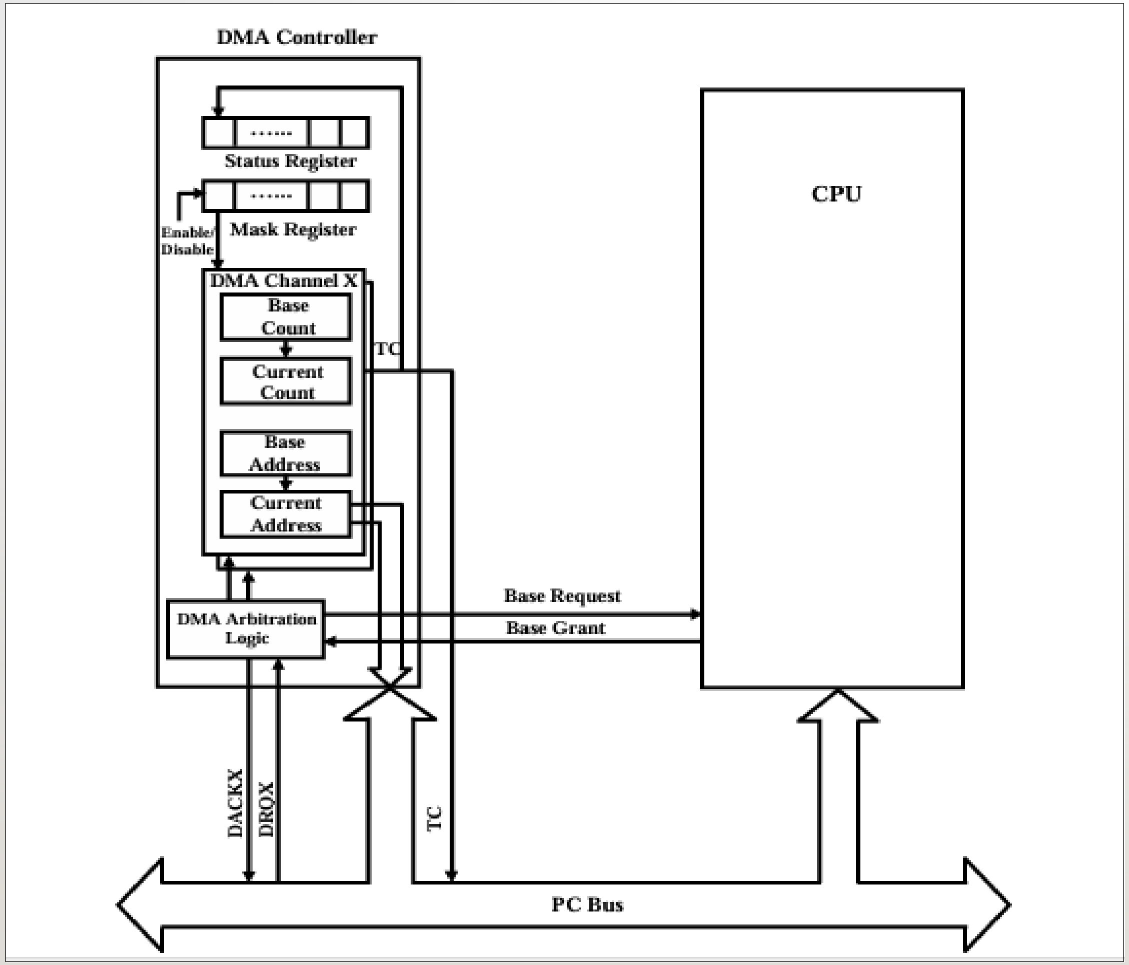
The three most common transfer modes in DMA are Single, block, and demand.

- Single transfer mode - transfers one data value for each DMA request assertion.
- This mode is the slowest method of transfer because it requires the DMA controller to arbitrate for the system bus with each transfer.
- This arbitration is not a major problem on a lightly loaded bus, but it can lead to latency problems when multiple devices are using the bus.
- Block and demand transfer modes - increase system throughput by allowing the DMA controller to perform multiple DMA transfers when the DMA controller has gained the bus.
- For block mode transfers, the DMA controller performs the entire DMA sequence as specified by the transfer count register at the fastest possible rate in response to a single DMA request from the I/O device.
- For demand mode transfers, the DMA controller performs DMA transfers at the fastest possible rate as long as the I/O device asserts its DMA request. When the I/O device unasserts this DMA request, transfers are held off.

DMA CONTROLLER OPERATION

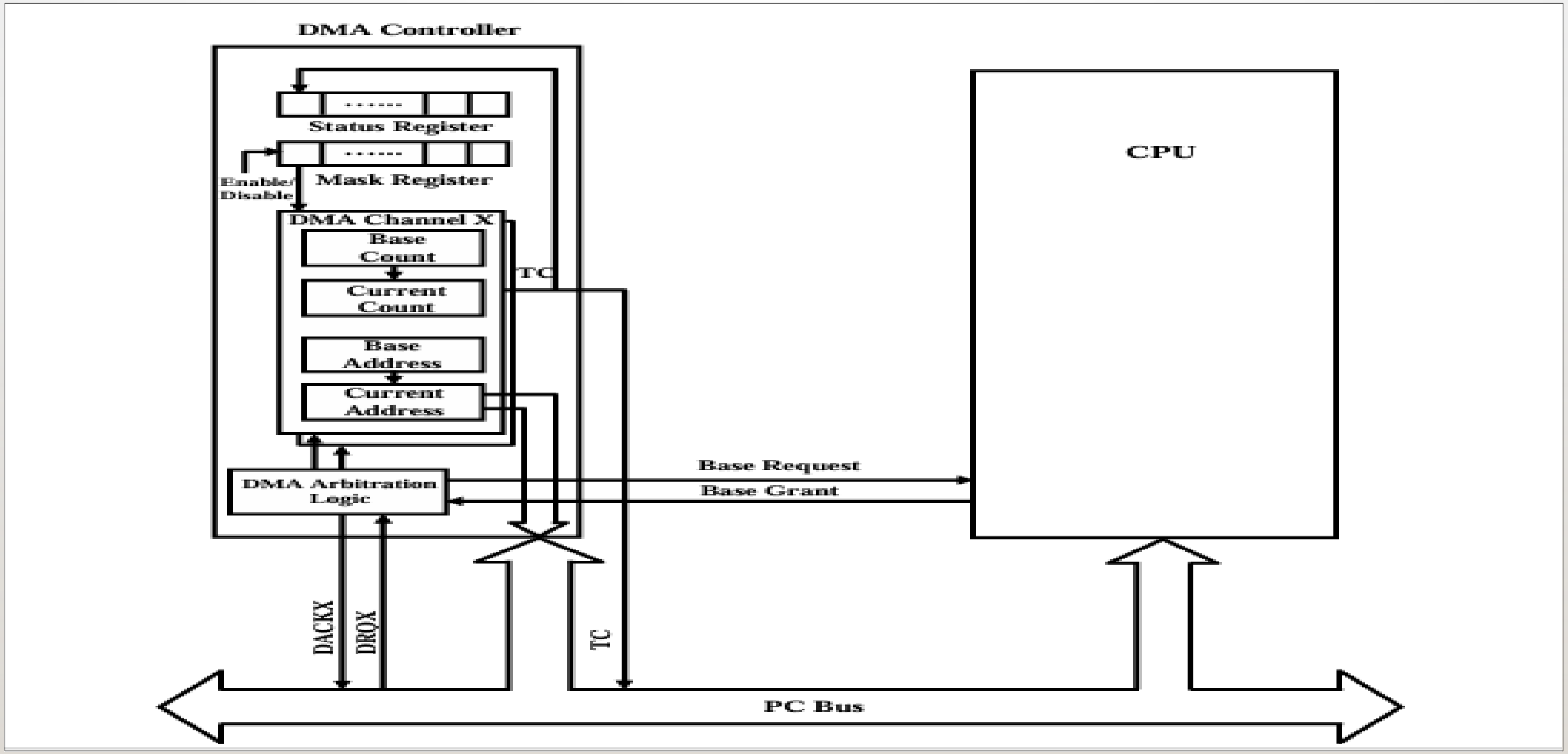


- For each channel, the DMA controller saves the programmed address and count in the base registers and maintains copies of the information in the current address and current count registers.
- Each DMA channel is enabled and disabled via a DMA mask register.
- When DMA is started by writing to the base registers and enabling the DMA channel, the current registers are loaded from the base registers.
- With each DMA transfer, the value in the current address register is driven onto the address bus, and the current address register is automatically incremented or decremented.
- The current count register determines the number of transfers remaining and is automatically decremented after each transfer.
- When the value in the current count register goes from 0 to -1, a terminal count (TC) signal is generated, which signifies the completion of the DMA transfer sequence



- This termination event is referred to as reaching terminal count. DMA controllers often generate a hardware TC pulse during the last cycle of a DMA transfer sequence.
- This signal can be monitored by the I/O devices participating in the DMA transfers. DMA controllers require reprogramming when a DMA channel reaches TC.
- Thus, DMA controllers require some CPU time, but far less than is required for the CPU to service device I/O interrupts.
- When a DMA channel reaches TC, the processor may need to reprogram the controller for additional DMA transfers.
- Some DMA controllers interrupt the processor whenever a channel terminates.

ARCHITECTURE OF DMA CONTROLLER AND DMA INTERACTION WITH CPU



- DMA controllers also have mechanisms for automatically reprogramming a DMA channel when the DMA transfer sequence completes.
- These mechanisms include auto initialization and buffer chaining. The auto initialization feature repeats the DMA transfer sequence by reloading the DMA channel's current registers from the base registers at the end of a DMA sequence and re-enabling the channel.
- Buffer chaining is useful for transferring blocks of data into noncontiguous buffer areas or for handling double buffered data acquisition.
- With buffer chaining, a channel interrupts the CPU and is programmed with the next address and count parameters while DMA transfers are being performed on the current buffer.
- Some DMA controllers minimize CPU intervention further by having a chain address register that points to a chain control table in memory.
- The DMA controller then loads its own channel parameters from memory. Generally, the more sophisticated the DMA controller, the less servicing the CPU has to perform.

- A DMA controller has one or more status registers that are read by the CPU to determine the state of each DMA channel.
- The status register typically indicates whether a DMA request is asserted on a channel and whether a channel has reached TC.
- Reading the status register often clears the terminal count information in the register, which leads to problems when multiple programs are trying to use different DMA channels.

STEPS IN A TYPICAL DMA CYCLE

Device wishing to perform DMA asserts the processors bus request signal.

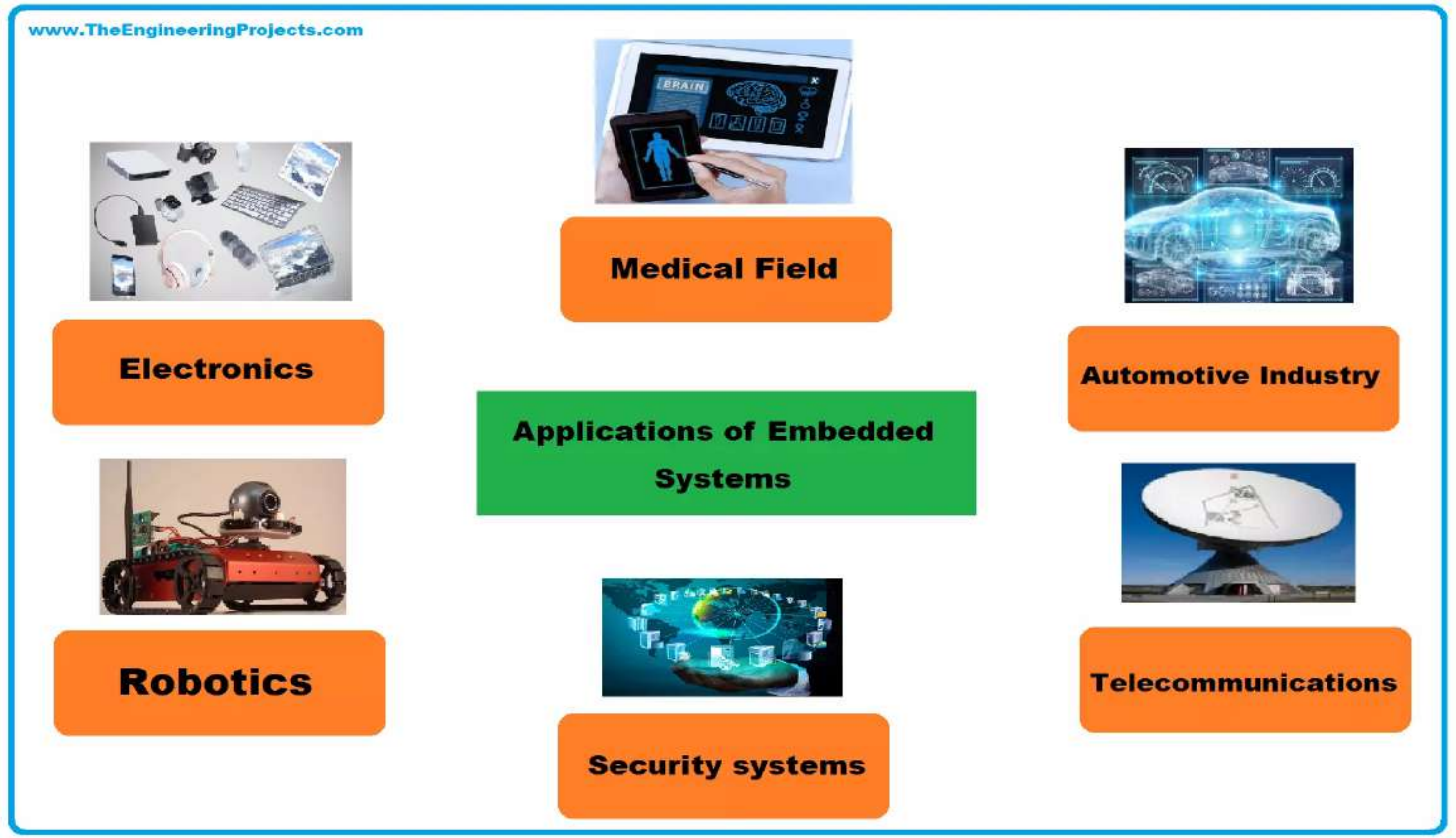
- Processor completes the current bus cycle and then asserts the bus grant signal to the device.
- The device then asserts the bus grant ack signal.
- The processor senses in the change in the state of bus grant ack signal and starts listening to the data and address bus for DMA activity.
- The DMA device performs the transfer from the source to destination address.
- During these transfers, the processor monitors the addresses on the bus and checks if any location modified during DMA operations is cached in the processor.

- If the processor detects a cached address on the bus, it can take one of the two actions:
  - Processor invalidates the internal cache entry for the address involved in DMA write operation
  - Processor updates the internal cache when a DMA write is detected
- Once the DMA operations have been completed, the device releases the bus by asserting the bus release signal.
- Processor acknowledges the bus release and resumes its bus cycles from the point it left off.

APPLICATIONS/ EXAMPLES OF EMBEDDED SYSTEMS

APPLICATIONS AND EXAMPLES

- Embedded systems are specialized computing systems that perform dedicated functions within larger systems.
- They are widely used across various industries due to their efficiency, real-time processing capabilities, and compact design.



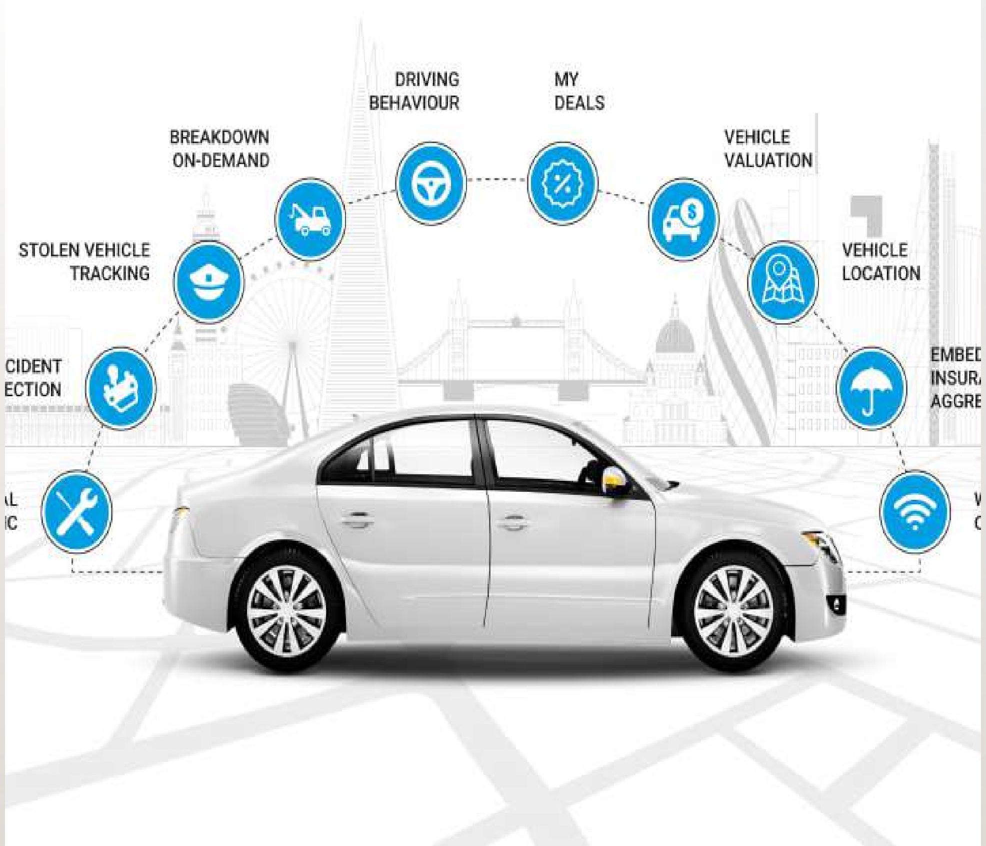
CONSUMER ELECTRONICS

- Smartphones & Tablets – Embedded processors manage touchscreen interfaces, sensors, and power efficiency.
- Smart TVs & Set-Top Boxes – Handle video decoding, user interface, and connectivity.
- Digital Cameras – Manage image processing, autofocus, and storage.
- Smart Home Devices – Thermostats (e.g., Nest), smart lights, and voice assistants (Alexa, Google Home).

Embedded Systems
Telephone
Television
Palm Top
Cell Phone
Washing Machine
Set Top Box
MP3 Players
PSP

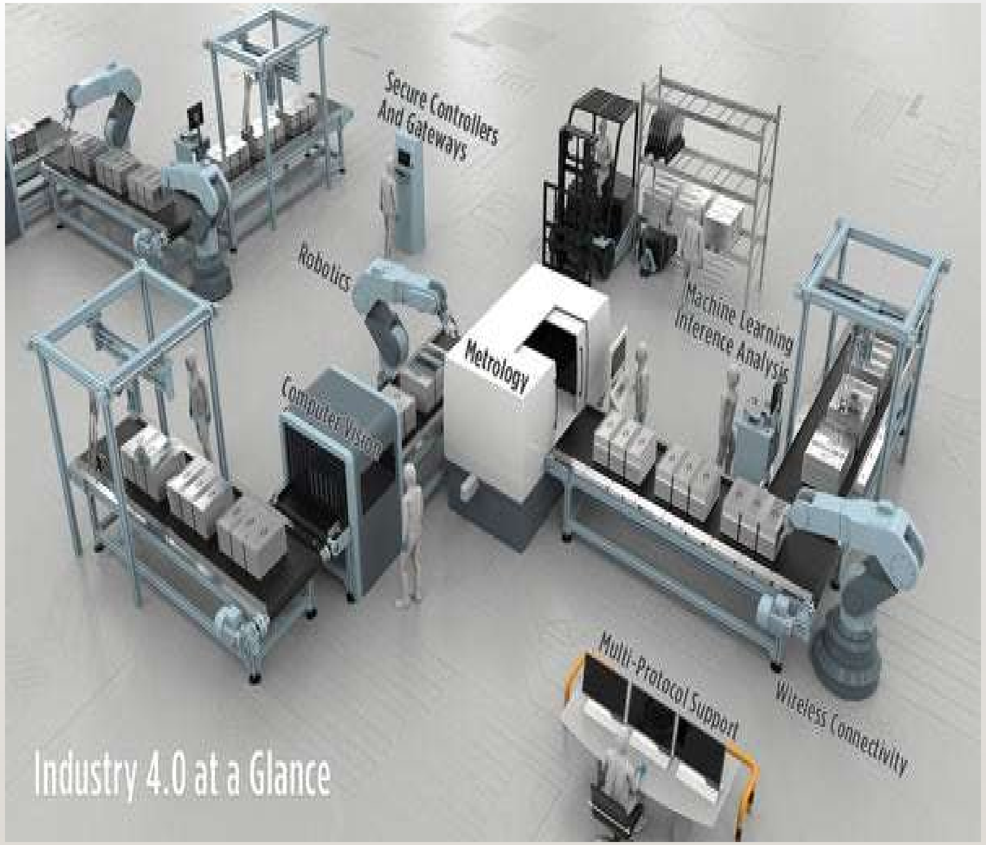
AUTOMOTIVE SYSTEMS

- Engine Control Units (ECUs) – Regulate fuel injection, emissions, and engine performance.
- Anti-lock Braking Systems (ABS) – Enhance vehicle safety by preventing wheel lock.
- Airbag Systems – Use embedded sensors to deploy airbags upon impact.
- Infotainment Systems – Navigation, Bluetooth, entertainment controls.



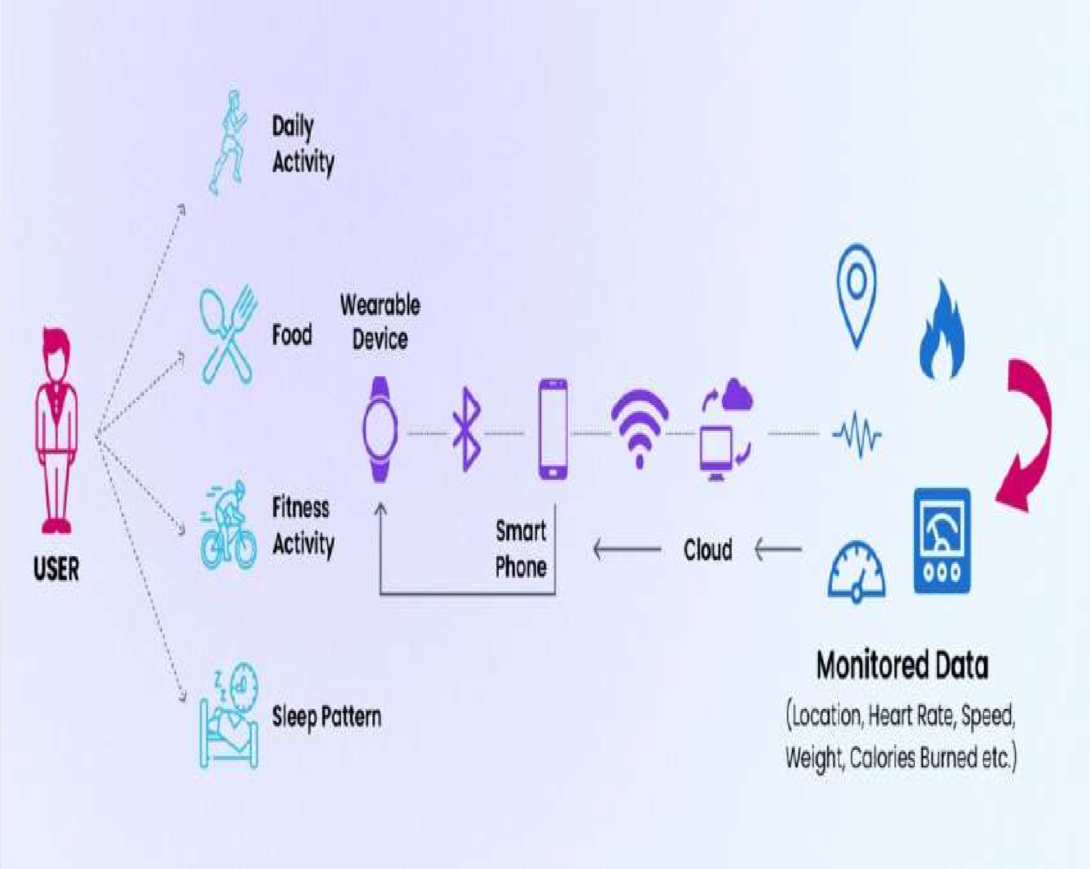
INDUSTRIAL AUTOMATION

- Programmable Logic Controllers (PLCs) – Control machinery in manufacturing plants.
- Robotics – Used in automated assembly lines and quality control.
- SCADA Systems – Monitor and control industrial processes remotely.



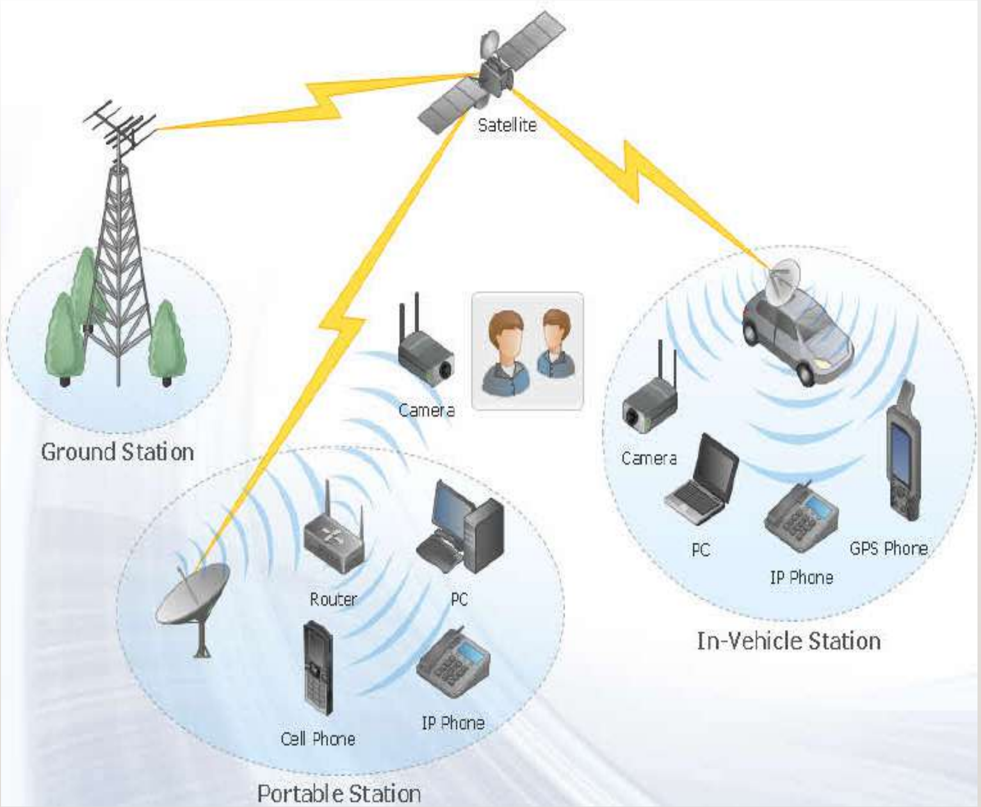
HEALTH CARE AND MEDICAL DEVICES

- Pacemakers – Monitor and regulate heartbeats.
- MRI & CT Scanners – Process medical imaging data.
- Blood Pressure Monitors & Glucose Meters – Portable health monitoring devices.



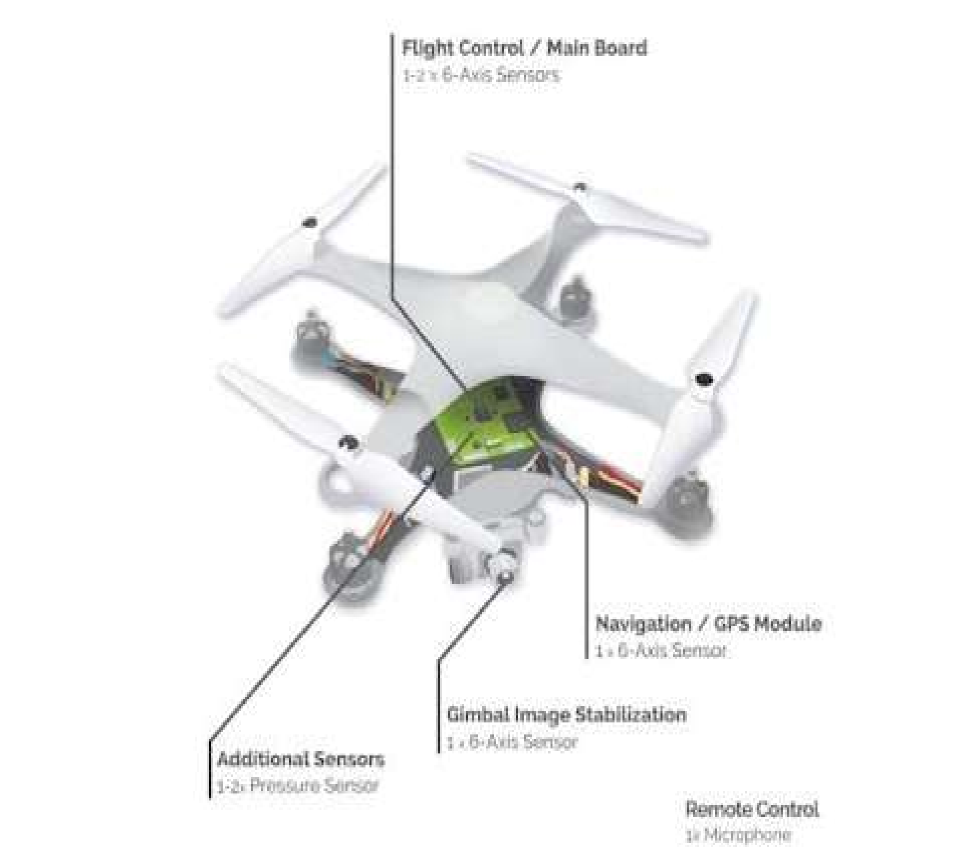
TELECOMMUNICATIONS

- Routers & Modems – Manage network traffic and connectivity.
- Base Station Controllers (BSCs) – Handle cellular communications.
- Satellite Communication Systems – Process signals for GPS and broadcasting.



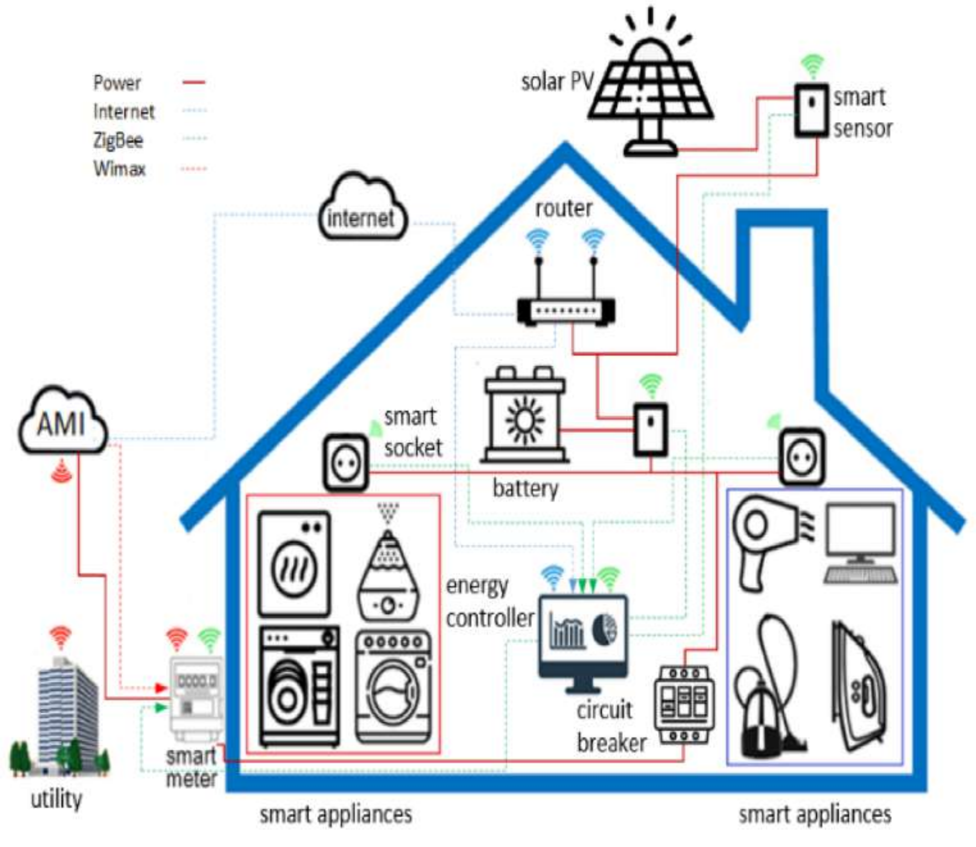
AEROSPACE AND DEFENSE

- Drones & UAVs – Autonomous navigation and surveillance.
- Missile Guidance Systems – Use real-time processing for targeting.
- Flight Control Systems – Manage aircraft stability and autopilot functions.



SMART APPLIANCES AND IOT DEVICES

- Smart Refrigerators – Monitor temperature and manage inventory.
- Washing Machines & Microwaves – Use embedded controllers for automation.
- Home Security Systems – Motion detectors, smart locks, and surveillance cameras.



BANKING AND RETAIL

- ATMs – Secure embedded systems for financial transactions.
- Point-of-Sale (POS) Terminals – Process credit/debit card transactions.
- Barcode Scanners – Used in supermarkets and warehouses.