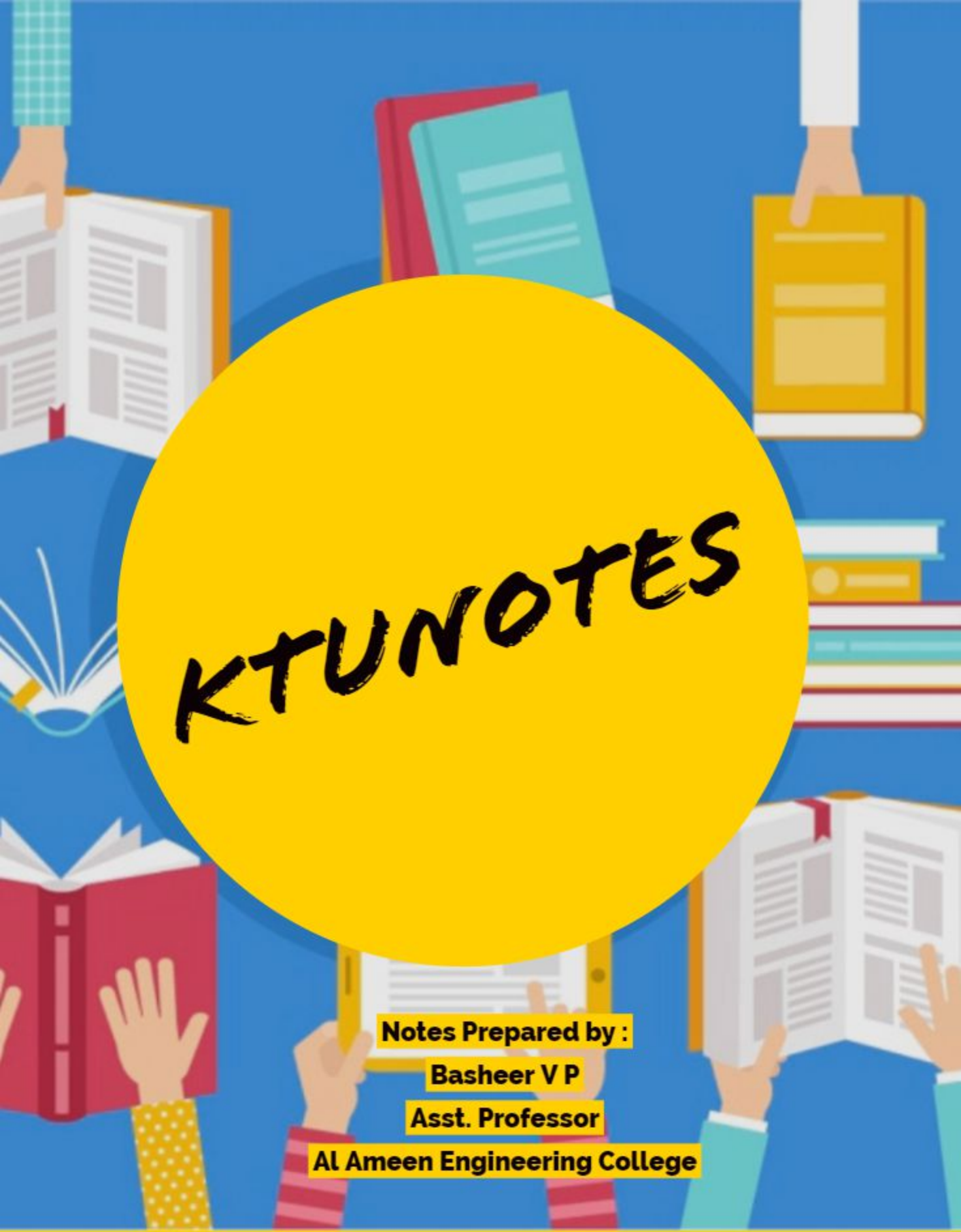


Notes Prepared by :

Basheer V P

Asst. Professor

Al Ameen Engineering College

EC 308 EMBEDDED SYSTEMS MODULE-2

SYLLABUS

Serial Communication Standards and Devices - UART, HDLC, SCI and SPI.

Serial Bus Protocols - I²C Bus, CAN Bus and USB Bus.

Parallel communication standards ISA, PCI and PCI-X

Bus.

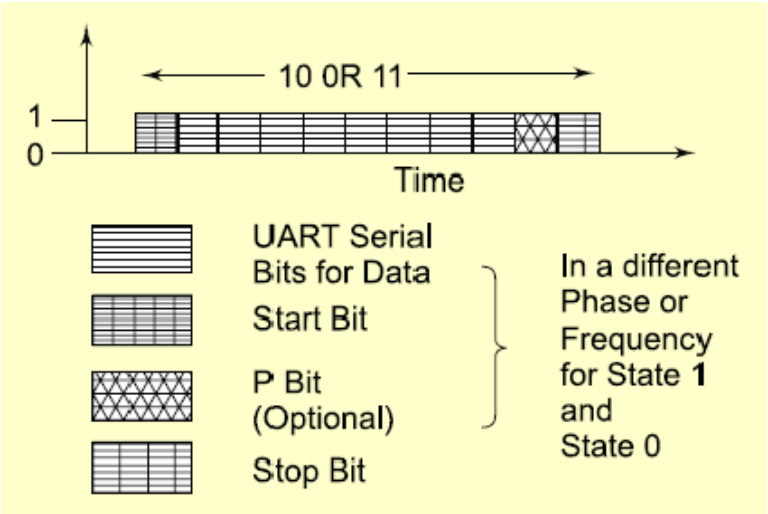
2.1 Communication Protocol

- A protocol is a standard adopted, which tells the way in which the bits of a frame must be sent from a device (or controller or port or processor) to another device or system

2.2 UART (Universal Asynchronous Receiver Transmitter)

UART mode is as follows

1. Idle State
   - A line non-return to zero (NRZ) state. It means in idle state the logic state is 1 at the serial line.
2. Byte start signaling flag bit
   - Start bit 1 to 0 transition, which receiver detect at the middle of bit interval T
3. Data bits
   - After start bit; *8* bits transmitted on TxD line and received on RxD line during period of 8 T (receiver detect at the middle of each bit interval T), In earlier circuits, the number of data bits could also be set 5, 6 or 7 in place of 8
4. Control or error detect bit
   - One bit- P-bit optional
   - P bit can be used to detect parity error
   - P-bit can be used to interpret the preceding byte not as data but as address or command or parity as per the processing circuit for serial bits at receiver
5. Byte end flag bit
   - Minimum one stop bit at Logic 1 [In earlier circuits, the number of stop bits could also be set 1½ or 2 in place of 1]
6. Disconnected State
   - Zero (Z) state
   - Disconnected serial line logic state is 0



UART Frame structure

2.3 HDLC (High-level Data Link Control)

- HDLC is a standard protocol for the data link network.
- For synchronous communication between two data link layers on a network.
- There are two formats Standard HDLC and Extended HDLC for 2^8 and 2^{16} destination devices or systems, respectively .

Sequence of bits in a HDLC frame

1. Frame start signaling flag bits;
   - Flag bits at start are 01111110

2. Address bits for destination
   - 8 bits in Standard HDLC format and 16 bits in extended format

3. Control field
   - The control field distinguishes between the three different types of frames used in HDLC, namely information, control and unnumbered frames.
   - The first one or two bits of the field determine the type of frame.
   - The field also contains control information which is used for flow control and link management.

4. Information field
   - The information field does not have a length specified by HDLC. In practice, it normally has a maximum length determined by a particular implementation.
   - Information frames (also known as I-frames) are the only frames that carry information bits which are normally in the form of a fixed-length block of data of several kilobytes in length.
   - All other types of frames (control and unnumbered frames) normally have an empty information field.

5. Frame check sequence
   - The FCS field contains error-checking bits, normally 16 bit with a provision for increasing this to 32.
   - 8 bits in Standard HDLC format and 16 bits in extended format

HDLC Frame types

- The different types of frame are distinguished by the contents of the control field.
- The structure of all three types of control field is shown in Figure.

Frame type	1	2	3	4	5	6	7	8	Bits
Information	0	N(S)			P	N(R)			
Supervisory	1	0	F		P	N(R)			
Unnumbered	1	1	F		P	F			

N(S) = send sequence number, N(R) = receive sequence number, F = function bits,
P = poll/final bit used for polling in normal response mode

Control Field Structure

Information frames

- An I-frame is distinguished by the first bit of the control field being a binary 0.
- The control field of an I-frame contains both a send sequence number, N(S), and a receive sequence number, N(R), which are used to facilitate flow control.
- N(S) is the sequence number of frames sent and N(R) the sequence number of frames successfully received by the sending node prior to the present frame being sent.
- Thus thefirst frame transmitted in a data transfer has send and receive sequence numbers 0,0.
- I-frames also contain a poll/final (P/F) bit (as do other frames). This acts as a poll bit when used by a primary station and a final bit by a secondary.
- A poll bit is set when a primary is transmitting to a secondary and requires a frame or frames to be returned in response, and the final bit is set in the final frame of a response.

Supervisory frames

- Supervisory frames are distinguished by the first 2 bits of the control field being 10.
- These frames are used as acknowledgements for flow and error control.
- Supervisory frames contain only a receive sequence number since they relate to the acknowledgement of I-frames and not to their transmission.
- They also contain two function bits which allow for four functions as shown in Table 2.1 which lists the supervisory commands/responses.

Table 5.1 Supervisory commands and responses.

Name	Function
Receive Ready (RR)	Positive acknowledgement (ACK), ready to receive I-frame
Receive Not Ready (RNR)	Positive acknowledgement, not ready to receive I-frame
Reject (REJ)	Negative acknowledgement (NAK), go-back-*n*
Selective Reject (SREJ)	Negative acknowledgement, selective-repeat

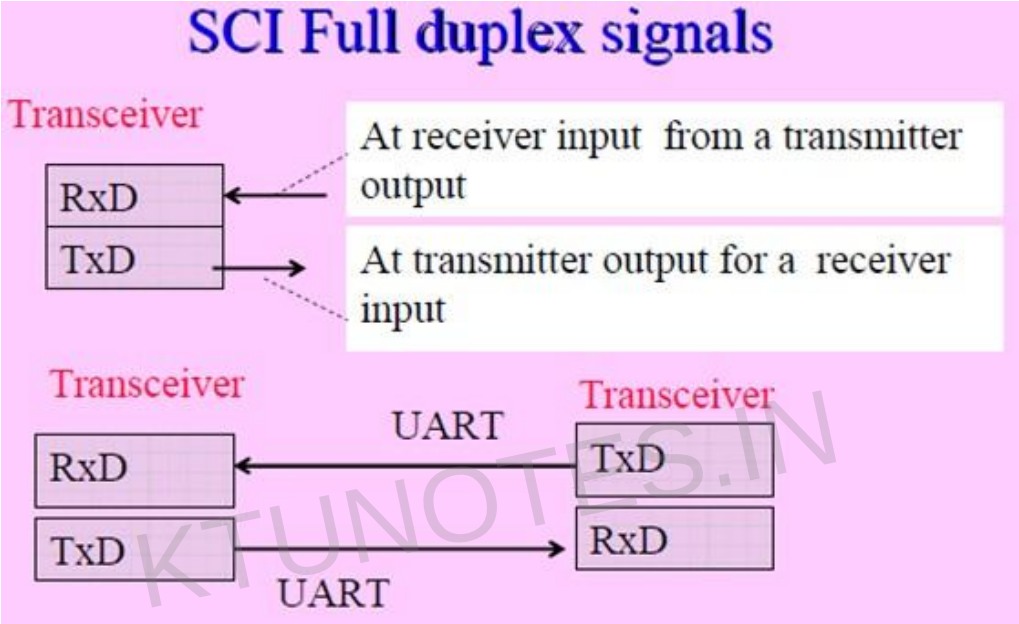
Unnumbered frames

- Unnumbered frames do not contain any sequence numbers (hence their name) and are used for various control functions.

- They have five function bits which allow for the fairly large number of commands and responses.

2.4 SCI (Serial Connect Interface)

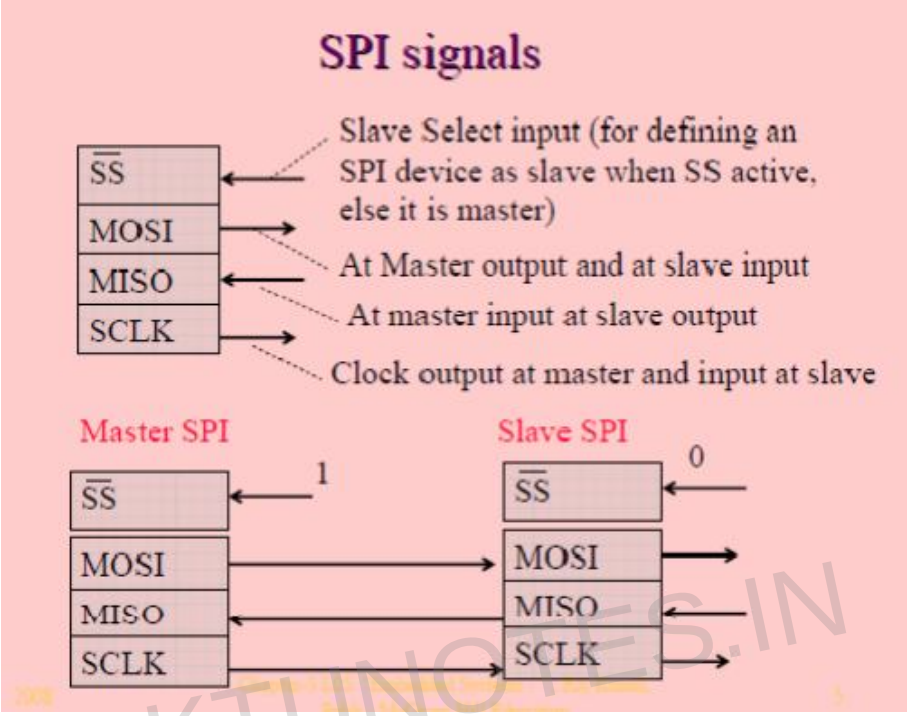
- SCI is a UART asynchronous mode port
- SCI is port with Full-duplex mode
- SCI is programmable for transmission and for reception



- Baud rate is selectable among 32 possible ones by the three- rate bits and two prescaling bits.
- Serial *in* and *out* lines baud rate not separately programmable
- SCI has two control register bits, T8 and R8 for the inter-processor communication in 11-bit format.
- SCI receiver *wake up* feature programmable by RWU (Receiver wakeup Unavailable bit)
- Wake up feature enabled if RWU is set, and is disabled if RWU is reset.
- If RWU if set, then the receiver of a slave does not interrupt by the succeeding frames.
- Number of processors can communicate on the SCI bus using control bits RWU, R8 and T8

2.5 SPI (Serial Peripheral Interface)

- SPI is a Full-duplex Synchronous communication protocol.
- The signals used in SPI are listed in the figure below.



- SCLK is the serial clock from master.
- MOSI is the output from master.
- MISO is the input to the master.
- Device selection as master or slave can be done by applying a signal to input SS pin. When this pin is 0, then the device act as Master. When this pin is 1, then the device act as slave.
- SPI is programmable for defining the occurrence of positive and negative edges within an interval of bits at serial data *out* or *in*
- SPI is also programmable for open-drain or totem pole output from a master to a slave.

2.6 SERIAL BUSCOMMUNICATION PROTOCOLS– I²C

_ Interconnecting number of device circuits, Assume flash memory, touch screen, ICs for measuring temperatures and ICs for measuring pressures at a number of processes in a plant.

_ ICs mutually network through a common synchronous serial bus I²C.

_ An 'Inter Integrated Circuit' (I²C) bus, a popular bus for these circuits.

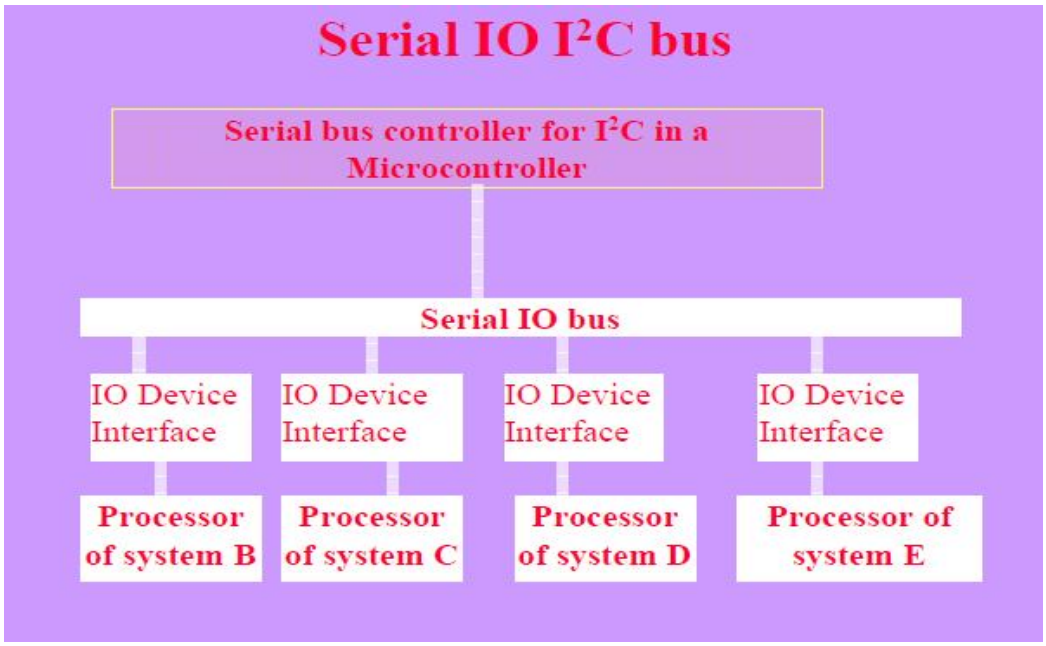
_Synchronous Serial Bus Communication for networking

_ Each specific I/O synchronous serial device may be connected to other using specific interfaces, for example, with I/O device using I²C controller

_ I²C Bus communication− use of only simplifies the number of connections and provides a common way (protocol) of connecting different or same type of I/O devices using synchronous serial communication

IO I²C Bus

_ Any device that is compatible with a I²C bus can be added to the system (assuming an appropriate device driver program is available), and a I²C device can be integrated into any system that uses that I²C bus.



I²C Bus

_ The Bus has two lines that carry its signals— one line is for the clock and one is for bi- directional data.

_ There is a standard protocol for the I²C bus.

Device Addresses and Master in the I²C bus

_ Each device has a 7-bit address using which the data transfers take place.

_ Master can address 127 other slaves at an instance.

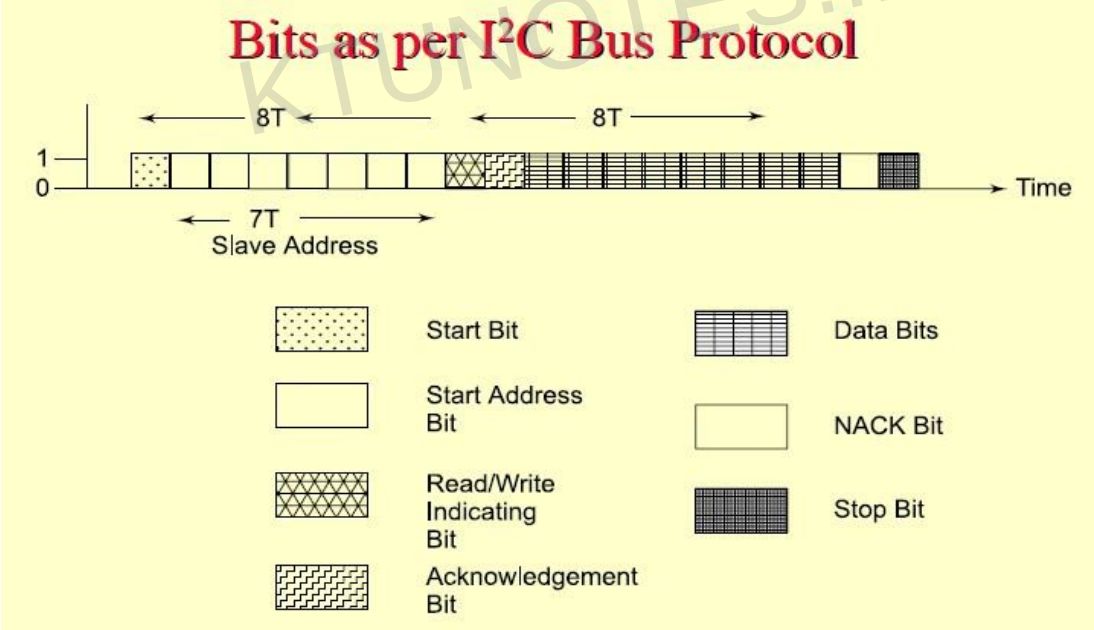
_ Master has at a processing element functioning as bus controller or a microcontroller with I²C (Inter Integrated Circuit) bus interface circuit.

Slaves and Masters in the I²C bus

_ Each slave can also optionally has I²C (Inter Integrated Circuit) bus controller and processing element.

_ Number of masters can be connected on the bus.

_ However, at an instance, master is one, which initiates a data transfer on SDA(serial data) line and which transmits the SCL (serial clock) pulses. From *master*, a data frame has fields beginning from start bit



Synchronous Serial Bus Fields and its length

_ First field of 1 bit– Start bit similar to one in an UART

_ Second field of 7 bits– address field. It defines the slave address, which is being sent the data frame (of many bytes) by the master

_ Third field of 1 control bit– defines whether a read or write cycle is in progress

_ Fourth field of 1 control bit– defines whether is the present data is an acknowledgment (from slave)

_ Fifth field of 8 bits– **I^2C device data byte**

_ Sixth field of 1-bit– bit NACK (negative acknowledgement) from the receiver. If active then acknowledgment after a transfer is not needed from the slave, else acknowledgement is expected from the slave

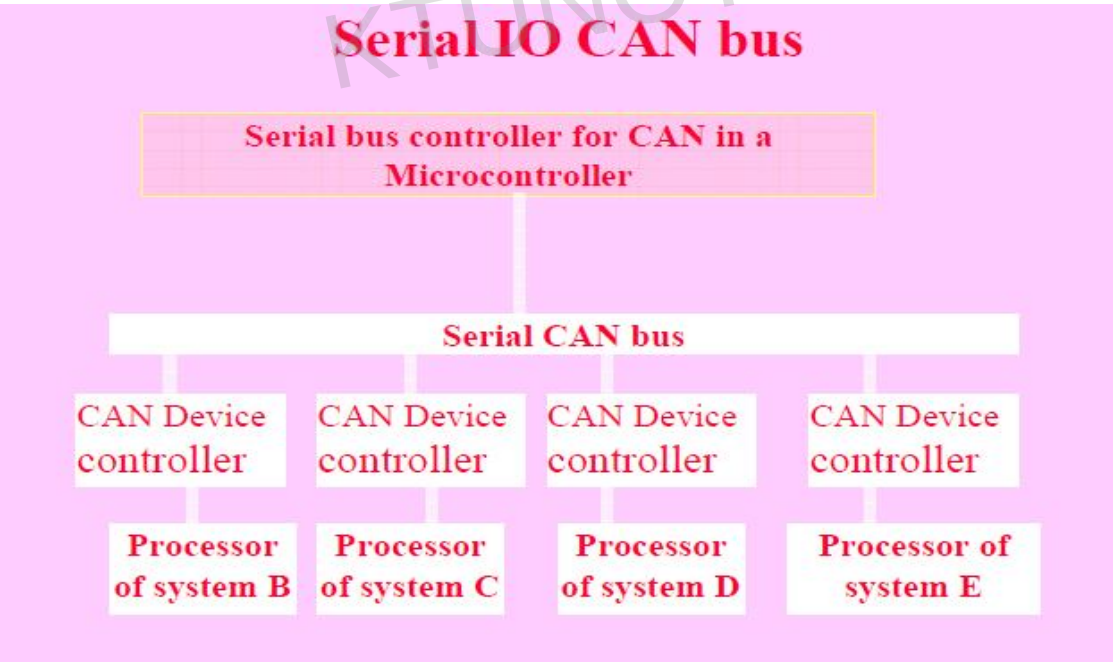
_ Seventh field of 1 bit – stop bit like in an UART

2.7 CAN

_CAN is a Distributed Control Area Network example - a network of embedded systems in automobile.

_ CAN-bus line usually interconnects to a CAN controller between line and host at the node. It gives the input and gets output between the physical and data link layers at the host node.

_ The CAN controller has a BIU (bus interface unit consisting of buffer and driver), protocol controller, status-cum control registers, receiver-buffer and message objects. These units connect the host node through the host interface circuit



CAN protocol

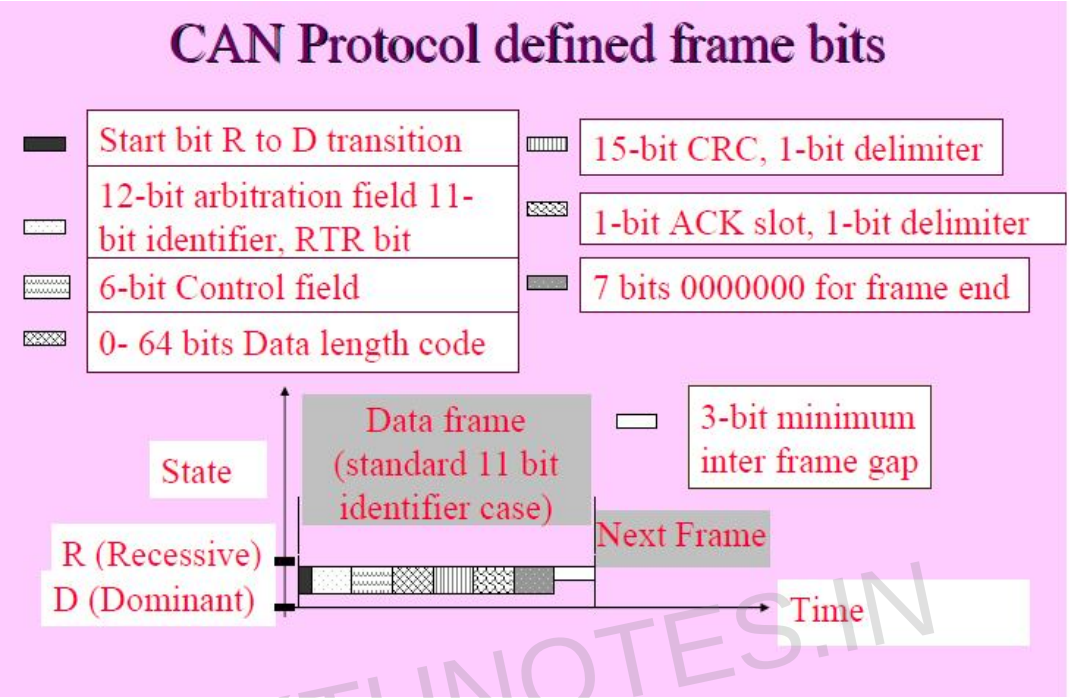
_There is a CAN controller between the CAN line and the host node.

_Protocol defined start bit followed by six fields of frame bits.

_ Data frame starts after first detecting that dominant state is not present at the CAN line with logic 1 (R state) to 0 (D state transition) for one serial bit interval

• After start bit, six fields starting from arbitration field and end with seven logic 0s end-field

• 3-bit minimum inter frame gap before next start bit (R→ D transition) occurs



_ **First field of 12 bits ─'arbitration field.**

_ 11-bit destination address and RTR bit (Remote Transmission Request)

_ Destination device address specified in an11-bit sub-field and whether the data byte being sent is a data for the device or a request to the device in 1-bit sub-field.

_ Maximum 211 devices can connect a CAN controller in case of 11-bit address fieldstandard11-bit address standard CAN

_ Identifies the device to which data is being sent or request is being made.

_ When RTR bit is at '1', it means this packet is for the device at destination address. If this bit is at

'0' (dominant state) it means, this packet is a request for the data from the

device.

_ **Second field of 6 bits─ control field.**

The first bit is for the identifier‘s extension.

_ The second bit is always '1'.

_ The last 4 bits specify code for data Length

_ **Third field of 0 to 64 bits—** Its length depends on the data length code in the control field.

• **Fourth field of 16 bits—** CRC (Cyclic Redundancy Check) bits.

• The receiver node uses it to detect the errors, if any, during the transmission

• **Fifth field of 2 bits—** First bit 'ACK slot'

• ACK = '1' and receiver sends back '0' in this slot when the receiver detects an error in the reception.

• Sender after sensing '0' in the ACK slot, generally retransmits the data frame.

• Second bit 'ACK delimiter' bit. It signals the end of ACK field.

• If the transmitting node does not receive any acknowledgement of data frame within a Specified time slot, it should retransmit.

• **Sixth field of 7-bits —** end- of- the frame specification and has seven '0's

2.8 USB

USB Host Applications Connecting

- flash memory cards,
- pen-like memory devices,
- digital camera,
- printer,
- mouse-device,
- PocketPC,
- video games,
- Scanner

Universal Serial Bus (USB)

_ **USB** is used for serial transmission and reception between host and serial devices

_ The data transfer is of four types: (a) Controlled data transfer, (b) Bulk data transfer, (c) Interrupt driven data transfer, (d) Iso-synchronous transfer

_ A bus between the host system and interconnected number of peripheral devices

USB Protocol Features

_ Maximum 127 devices can connect a host.

_ Three standards: USB 1.1 (a low speed1.5 Mbps 3 meter channel along with ahigh speed 12 Mbps 25 meter channel),USB 2.0 (high speed 480 Mbps 25meter channel), and wireless USB(high speed 480 Mbps 3 m)

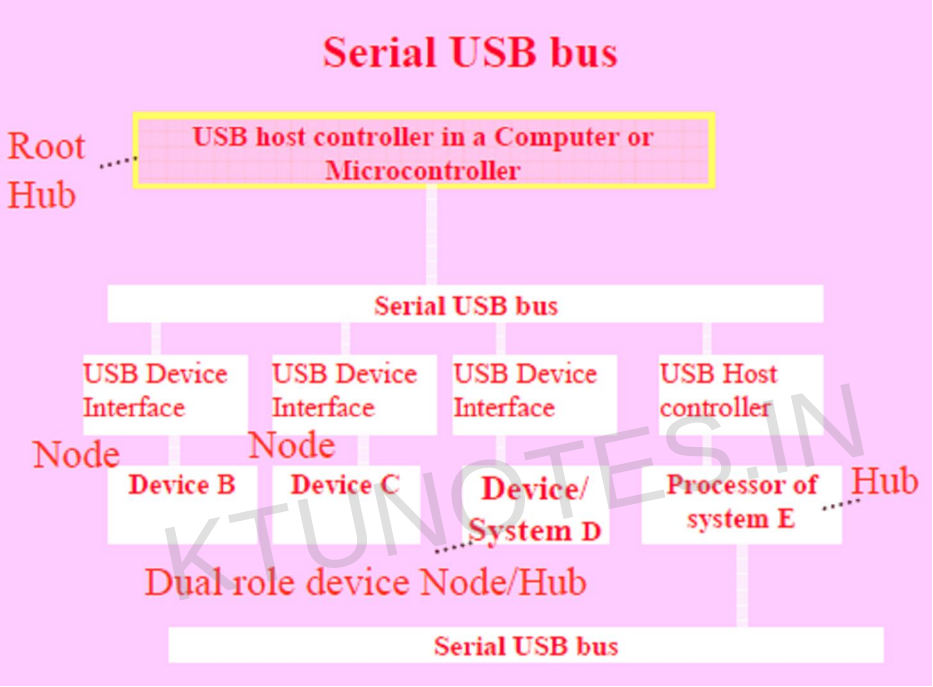
Host connection to the devices or nodes

_ Using USB port driving software and host controller,

_ Host computer or system has a host controller, which connects to a root hub.

_ A hub is one that connects to other nodes or hubs.

_ A tree- like topology



USB Device features

_ Can be hot plugged (attached), configured and used, reset, reconfigured and used

_ Bandwidth sharing with other devices: Host schedules the sharing of bandwidth among the attached devices at an instance.

_ Can be detached (while others are in operation) and reattached.

_ Attaching and detaching USB device or host without rebooting

Powering USB device

_ A device can be either bus-powered or self- powered.

_ In addition, there is a power management by software at the host for USB ports

USB protocol

_ USB bus cable has four wires, one for+5V, two for twisted pairs and one for ground.

_ Termination impedances at each end as per the device-speed.

_ Electromagnetic Interference (EMI)-shielded cable for the 15 Mbps USB devices.

_ Serial signals NRZI (Non Return to Zero (NRZI)

_ The synchronization clock encoded by inserting synchronous code (SYNC) field before each USB packet

_ Receiver synchronizes its bits recovery clock continuously

A polled bus

_Host controller regularly polls the presence of a device as scheduled by the software.

_It sends a token packet.

_The token consists of fields for type, direction, USB device address and device end-point number.

_The device does the handshaking through a handshake packet, indicating

successful or unsuccessful transmission.

_A CRC field in a data packet permits error detection

USB supported three types of pipes

1. 'Stream' with no USB- defined protocol. It is used when the connection is already established and the data flow starts
2. 'Default Control' for providing access.
3. 'Message' for the control functions for of the device.

• Host configures each pipe with the data bandwidth to be used, transfer service type and buffer sizes.

PARALLEL BUS DEVICE PROTOCOLS

2.9 Peripheral Component Interconnect (PCI) Bus

_ Parallel bus enables a host computer or system to communicate simultaneously 32-bit or 64-bit with other devices or systems, for example, to a network interface card (NIC) or graphic card.

_ When the I/O devices in the distributed embedded subsystems are networked all can communicate through a common parallel bus.

_PCI connects at high speed to other subsystems having a range of I/O devices at very short distances (<25 cm) using a parallel bus without having to implement a specific interface for each I/O device.

PCI Bus Feature

_ 32- bit data bus extendible to 64 bits.

_ PCI protocol specifies the ways of interaction between the different components of a computer.

_ A specification version 2.1─synchronous/asynchronous throughput is up to 132/ 528 MB/s [33M× 4/ 66M× 8 Byte/s], operates on 3.3V to 5Vsignals.

_ PCI driver can access the hardware automatically as well as by the programmer assigned addresses.

_ Automatically detects the interfacing systems and assigns new addresses

_ Thus, simplified addition and deletion (attachment and detachment) of the system peripherals.

FIFO in PCI device/card

_ Each device may use a FIFO controller with a FIFO buffer for maximum throughput.

Identification Numbers

_ A device identifies its address space by three identification numbers, (i) I/O port (ii) Memory locations and (iii)Configuration registers of total 256Bwith a four 4-byte unique ID. Each PCI device has address space allocation of256 bytes to access it by the host Computer

PCI device identification

_ A sixteen16-bit register in a PCI device identifies this number to let that device auto- detect it.

_ Another sixteen16-bit register identifies a device ID number. These two numbers let allow the device to carry out its auto-detection by its host computer.

PCI bridge

_ PCI bus interface switches a process or communication with the memory bus to PCI bus.

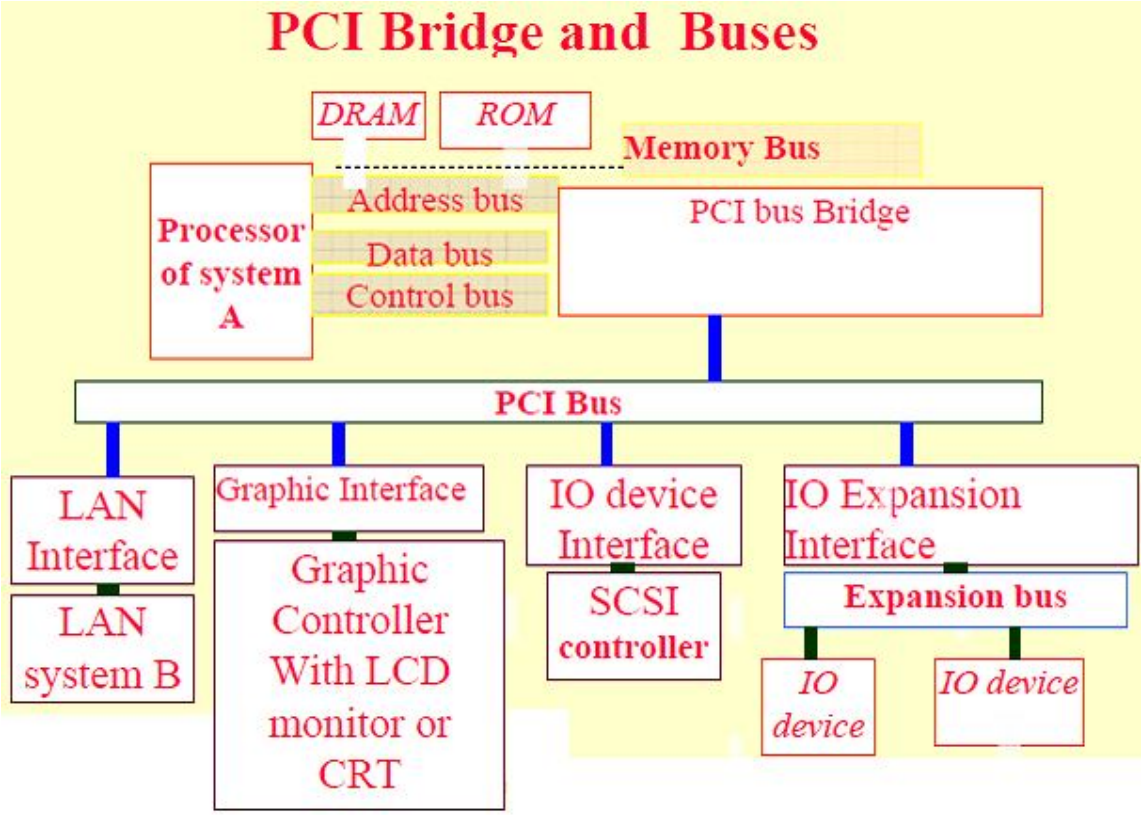
_ In most systems, the processor has a single data bus that connects to a switch module

_ Some processors integrate the switch module onto the same integrated circuit as the processor to

reduce the number of chips required to build a system and thus the system cost.

_ Communicates with the memory through a *memory bus* (a set of address, control and data buses), a dedicated set of wires that transfer data between these two systems.

_ A separate *I/O bus* connects the PCI switch to the I/O devices.



PCI Bus

2.10 PCI-X Bus

- 133 MBps to as much as 1 GBps
- Backward compatible with existing PCI cards
- Used in high bandwidth devices (Fiber Channel, and processors that are part of a cluster and Gigabit Ethernet)
- Maximum 264 MBps throughput, uses 8,, 32, or 64 bit transfers
- 6U cards contain additional pins for user defined I/Os
- Live insertion support (Hot-Swap),
- Supports two independent buses on the back plane (on different connectors)
- Supports Ethernet, Infiniband, and Star Fabric support (Switched fabric based systems) Compact PCI (cPCI)