Sahrdaya Tech Talks

CS 404 Embedded System

- **NETWORKS FOR EMBEDDED SYSTEMS**
  - **I_2C BUS**
  - **ETHERNET**
- **NETWORK BASED DESIGN**

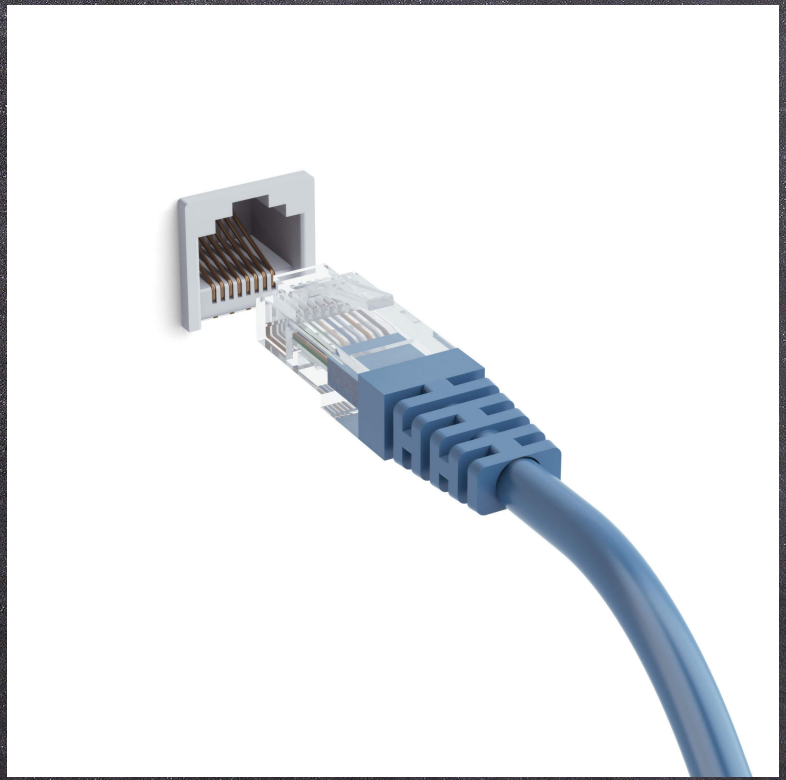
Fact you should know before getting started

I_2C bus Cable

EXAMPLE : ARDUINO TO SENSORS CONNECTION

CAN Cable

EXAMPLE : VGI PORT CABLES



Ethernet Cable

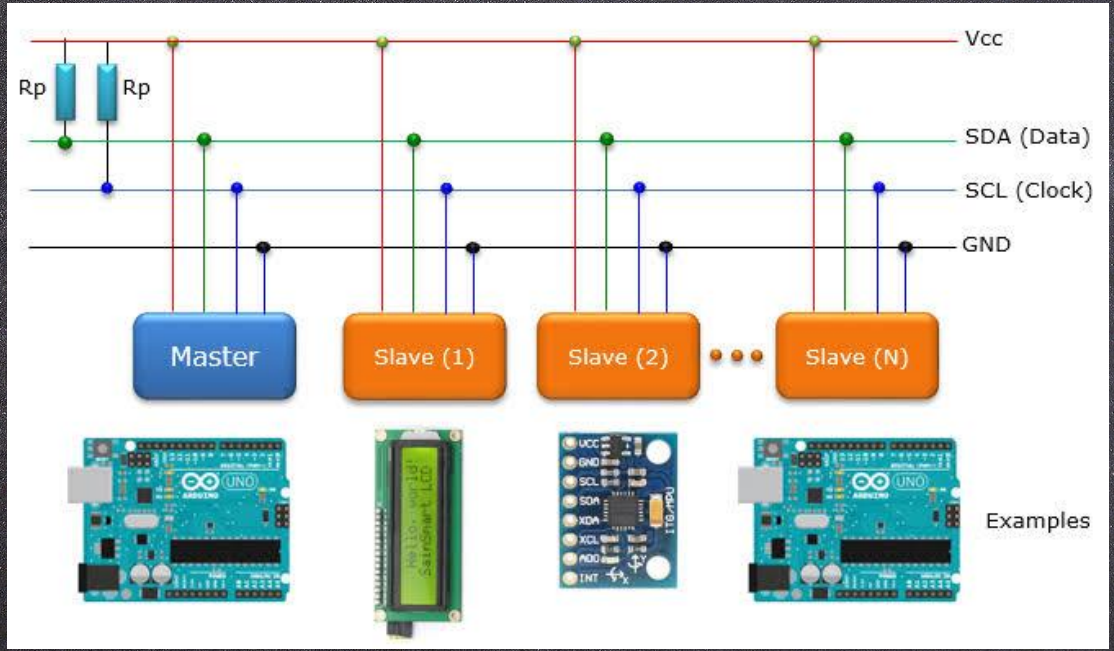
EXAMPLE : LAN PORT CABLE

CHAPTER 8.2

NETWORK FOR EMBEDDED SYSTEM

?...

The field of **embedded networking** deals with the **network** design and topology, hardware devices, and communication / data exchange protocols needed to connect and exchange information between **embedded systems**.



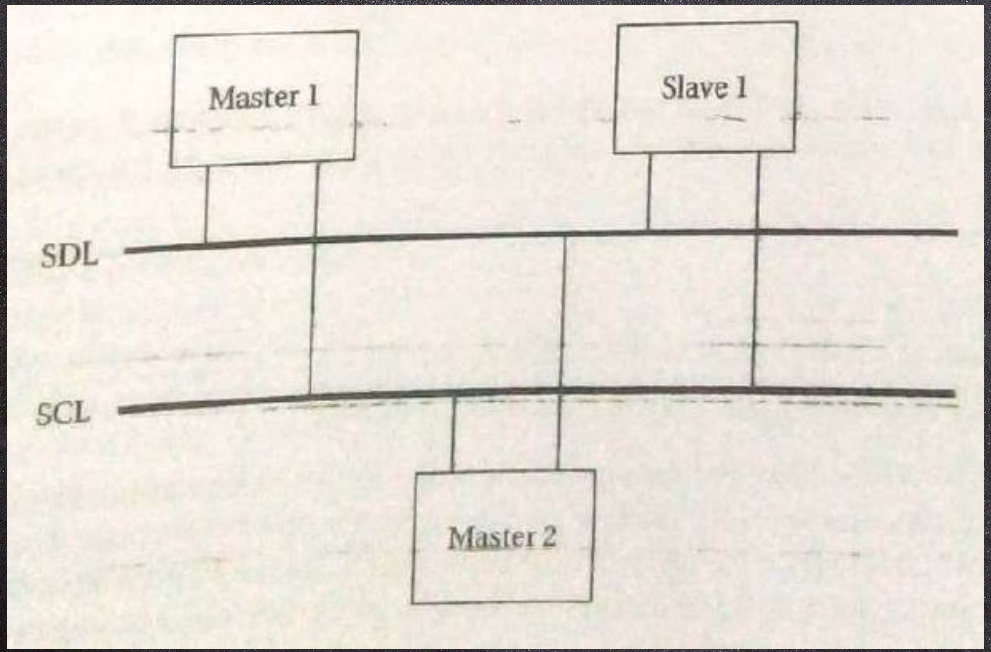
I2C Bus Communication

I2C Bus Communication (Short notes)

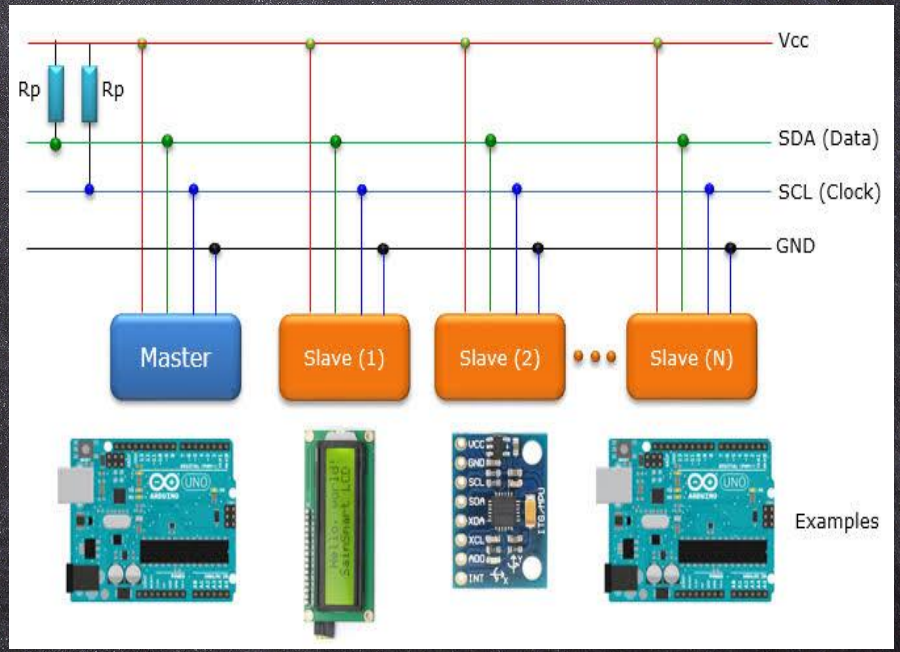
- The I2C bus is a well-known bus commonly used to link microcontrollers into systems.
- I2C is designed to be low cost, easy to implement, and of moderate speed up to 100 KB/s for the standard bus and up to 400 ICB/s for the extended bus
- As a result, it uses only two lines: the **serial data line (SDL)** for data and the **serial clock line (SCL)**, which indicates when valid data are on the data line
- Every node in the network is connected to both SCL and SDL
- Some nodes may be able to act as bus masters and the bus may have more than one master.
- Where , Other nodes may act as slaves that only respond to requests from masters.

I_2C Bus Communication (Short notes)

ORIGINAL



FOR REFERENCE ONLY



Structure of I_2C Bus system

I_2C Bus Communication (Short notes)

- The bus does not define particular voltages to be used for high or low so that either bipolar or MOS circuits (transistors) can be connected to the bus. (eg : digital format 10101001 etc..)

- A pull-up resistor keeps the default state of the signal high, and transistors are used in each bus device to pull down the signal when a 0 is to be transmitted.

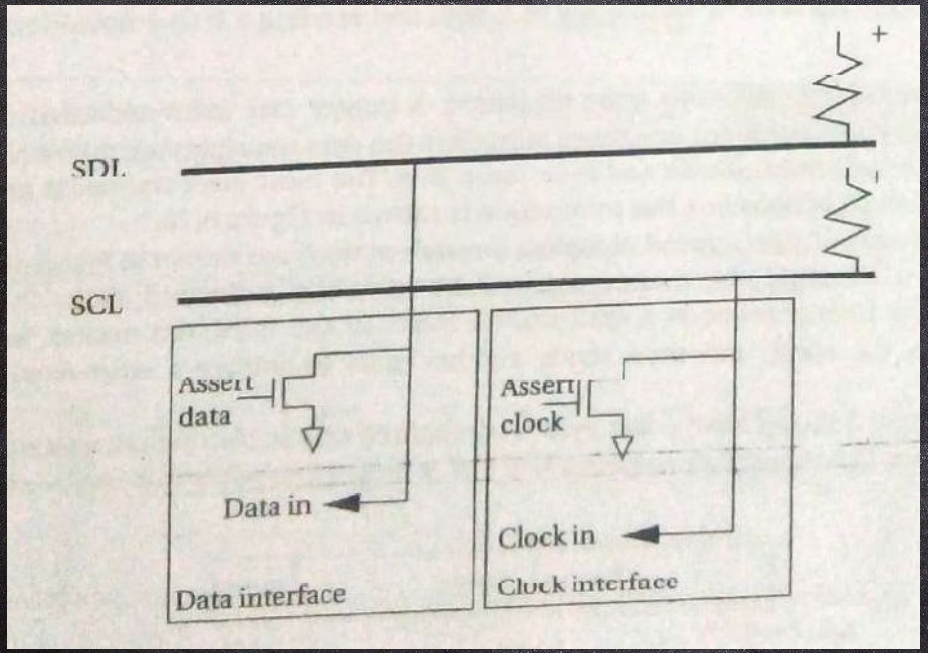
- Open collector / open drain signaling allows several devices to simultaneously write the bus without causing electrical damage.

- The master is responsible for generating the SCL clock

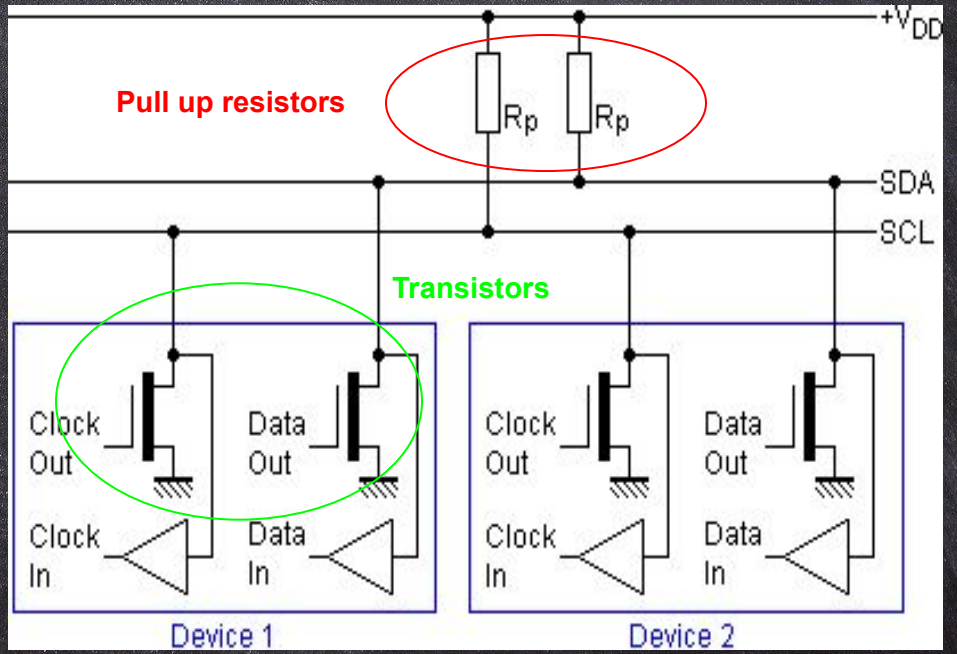
- The 1^2C bus is designed as a multimaster bus . any one of several different devices may act as the master at various times. As a result, there is no global master to generate the clock signal on SCL.

I_2C Bus Communication (Short notes)

ORIGINAL



FOR REFERENCE ONLY



Electrical interface to I_2C Bus system

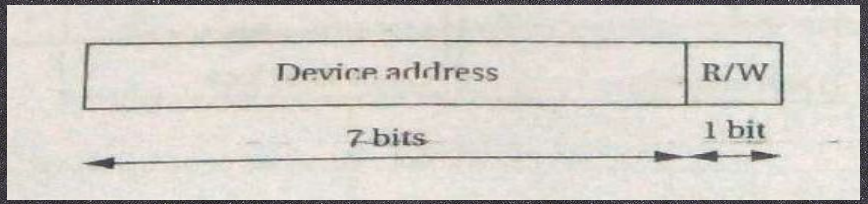
I_2C Bus Communication (Short notes)

- Instead, a master drives both SCL and SDL when it is sending data.
- When the bus is idle, both SCL and SDL remain high. When two devices try to drive either SCL or SDL to different values, the open collector/ open drain circuitry prevents errors,
- but each master device must listen to the bus while transmitting. to be sure that it is not interfering with another message
- if the device receives a different value than it is trying to transmit, then it knows that it is interfering with another message.
- Every I_2C device has an address. The addresses of the devices are determined by the system designer, usually as part of the program for the I2C driver.
- The addresses must of course be chosen so that no two devices in the system have the same address.

I_2C Bus Communication (Short notes)

- A device address is 7 bits in the standard 12c definition (the extended I2C allows 10-bit addresses).
- The address 0000000 is used to signal a general call or bus broadcast, which can be used to signal all devices simultaneously.
- The address 11110XX is reserved for the extended 10-bit addressing scheme; there are several other reserved addresses as well.
- A bus transaction comprised a series of 1-byte transmissions and an address followed by one or more data bytes.
- 1^2C encourages a data-push programming style. When a master wants to write a slave, it transmits the slave's address followed by the data.
- Since a slave cannot initiate a transfer, the master must send a read request with the slave's address and let the slave transmit the data.

I2C Bus Communication (Short notes)



Format of an I2C Address Transmission

an address transmission includes the 7-bit address and 1 bit for data direction: 0 for writing from the master to the slave and 1 for reading from the slave to the master.

- TO READ (from slave and send to master)
  // get info from slave

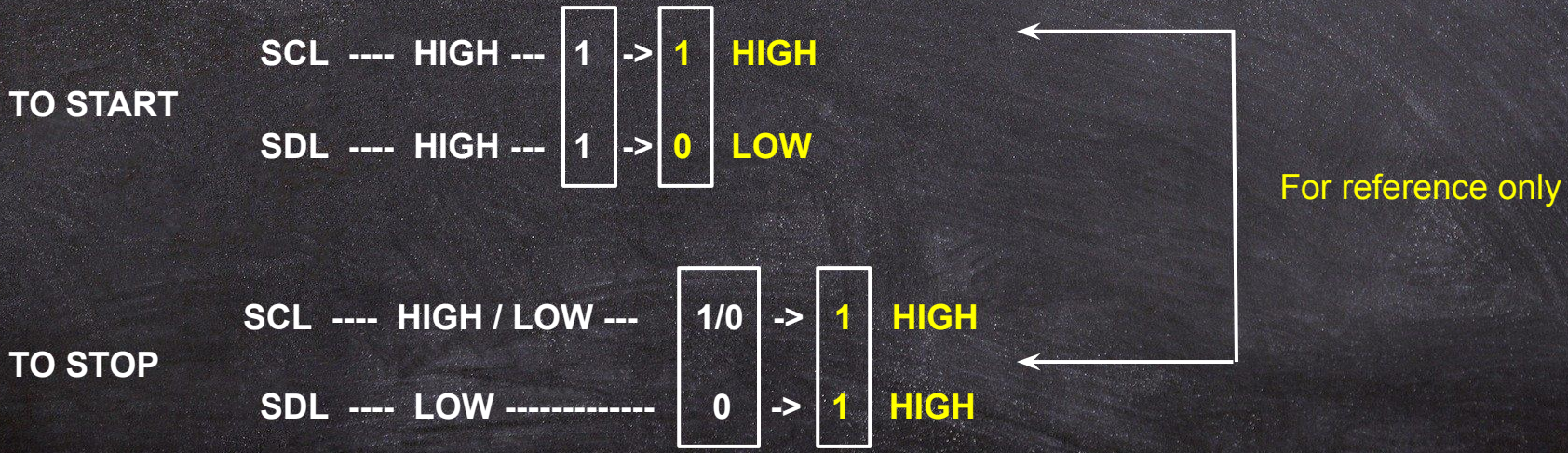
1 0 1 1 0 1 0	1
Device 7 bit address	*1 bit for Data Direction*

- TO WRITE (from master to slave)
  // give command to slave

1 0 1 1 0 1 0	0
Device 7 bit address	*1 bit for Data Direction*

I_2C Bus Communication (Short notes)

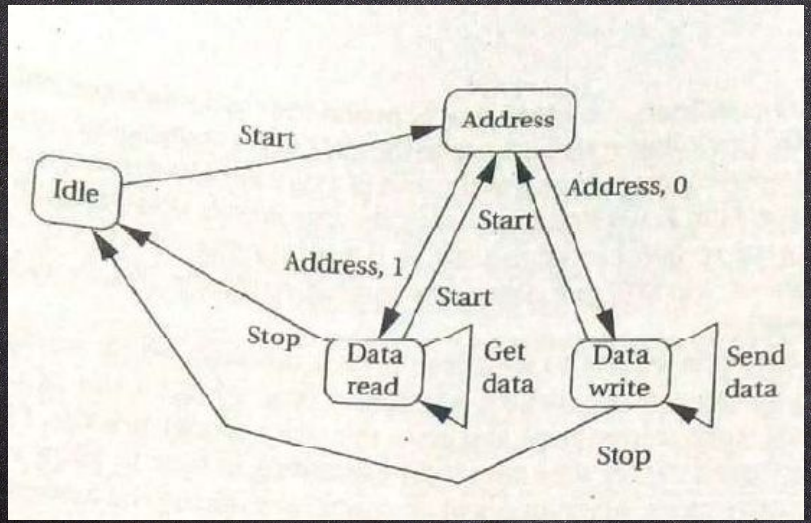
- A bus transaction is initiated by a start signal and completed with an end signal as follows:
- A start is signaled by leaving the SCL high and sending a 1 to 0 transition on SDL
- A stop is signaled by setting the SCL high and sending a 0 to 1 transition on SDL.



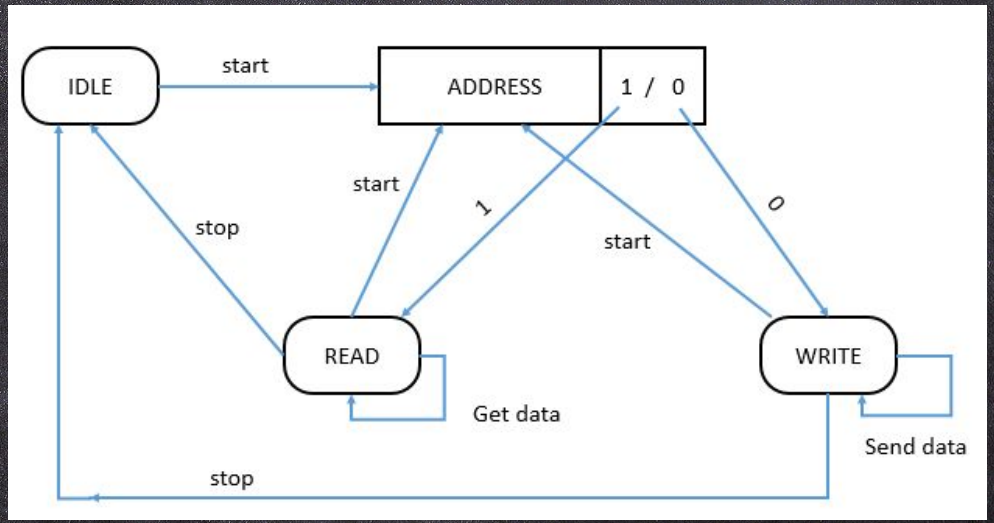
I_2C Bus Communication (Short notes)

- However, starts and stops must be paired. A master can write and then read (or read and then write) by sending a start after the data transmission, followed by another address transmission and then more data.

ORIGINAL



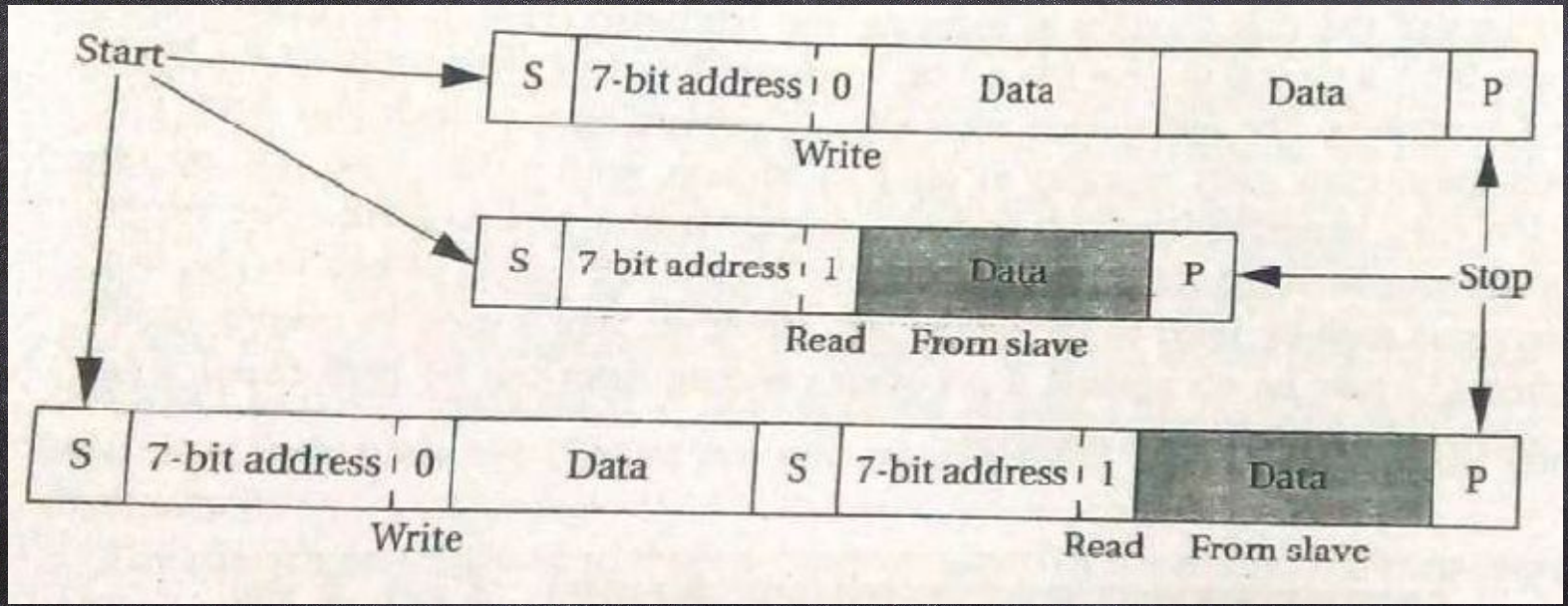
FOR REFERENCE ONLY



State transition graph for I_2C Bus Master

I_2C Bus Communication (Short notes)

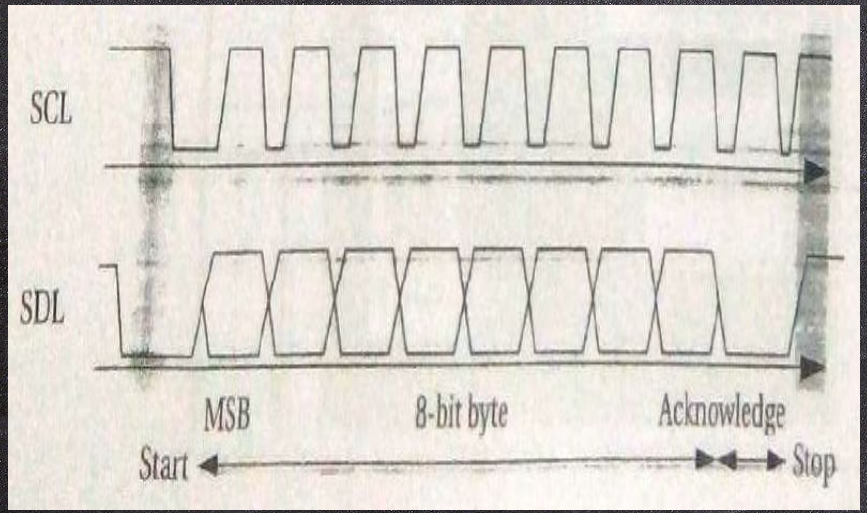
- In the first example,the master writes 2 bytes to the addressed slave. In the second, the master requests a read from a slave. In the third, the master writes 1 byte to the slave, and then sends another start to initiate a read from the slave.



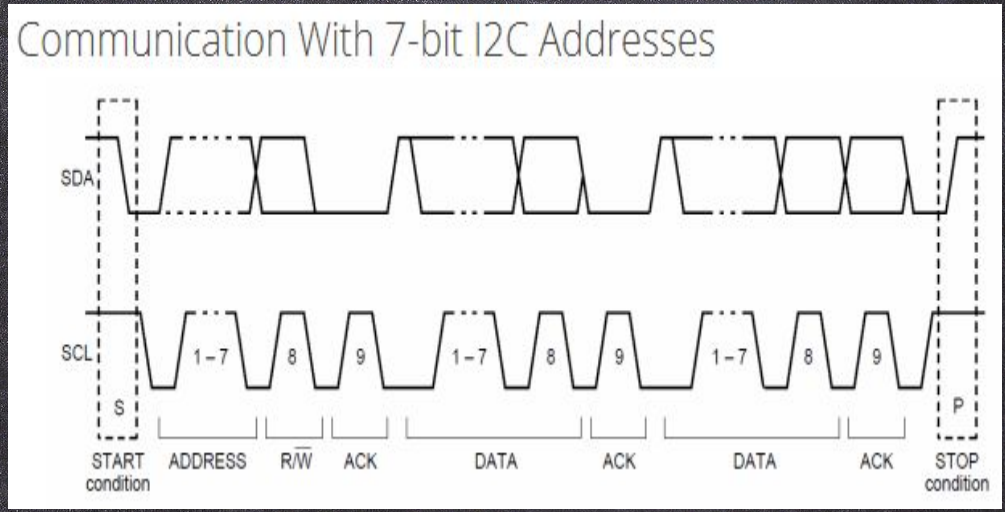
Typical bus transaction on the I_2C Bus

I_2C Bus Communication (Short notes)

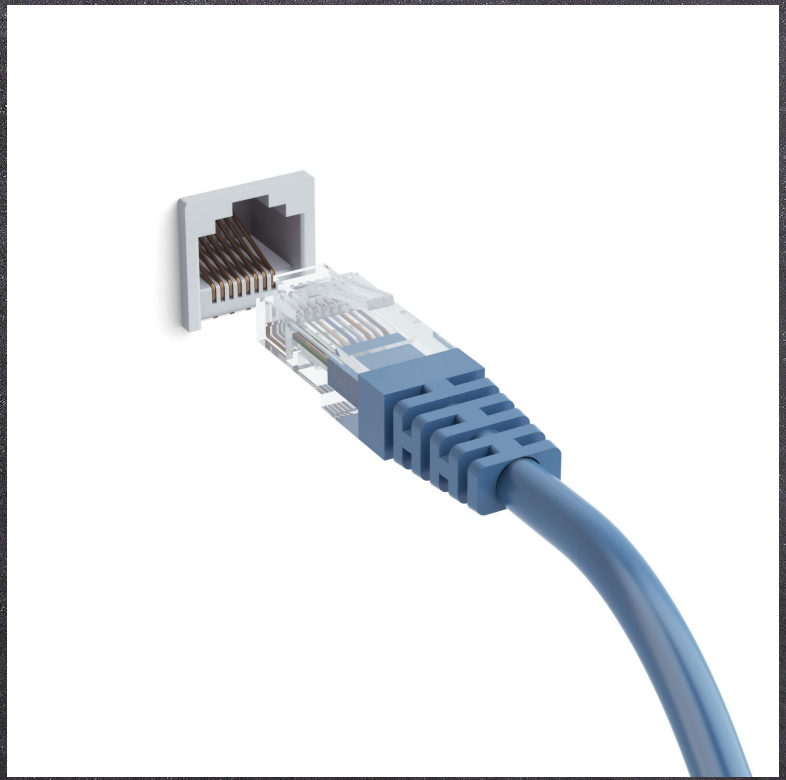
ORIGINAL



FOR REFERENCE ONLY



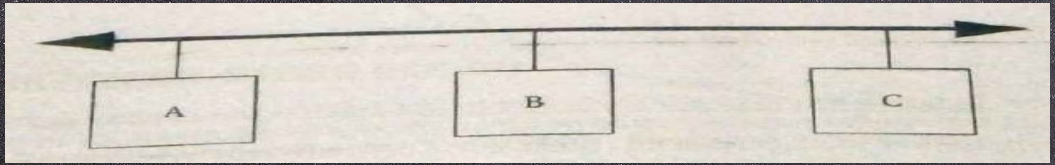
Transmitting a byte on the I_2C Bus



Ethernet

Ethernet (Short notes)

- Ethernet is very widely used as a local area network for general-purpose Computing. Because of Its Ubiquity and the low cost of Ethernet interfaces.
- Ethernet is particularly useful when PCs are used as platforms
- Physical organisation of an Ethernet is given below

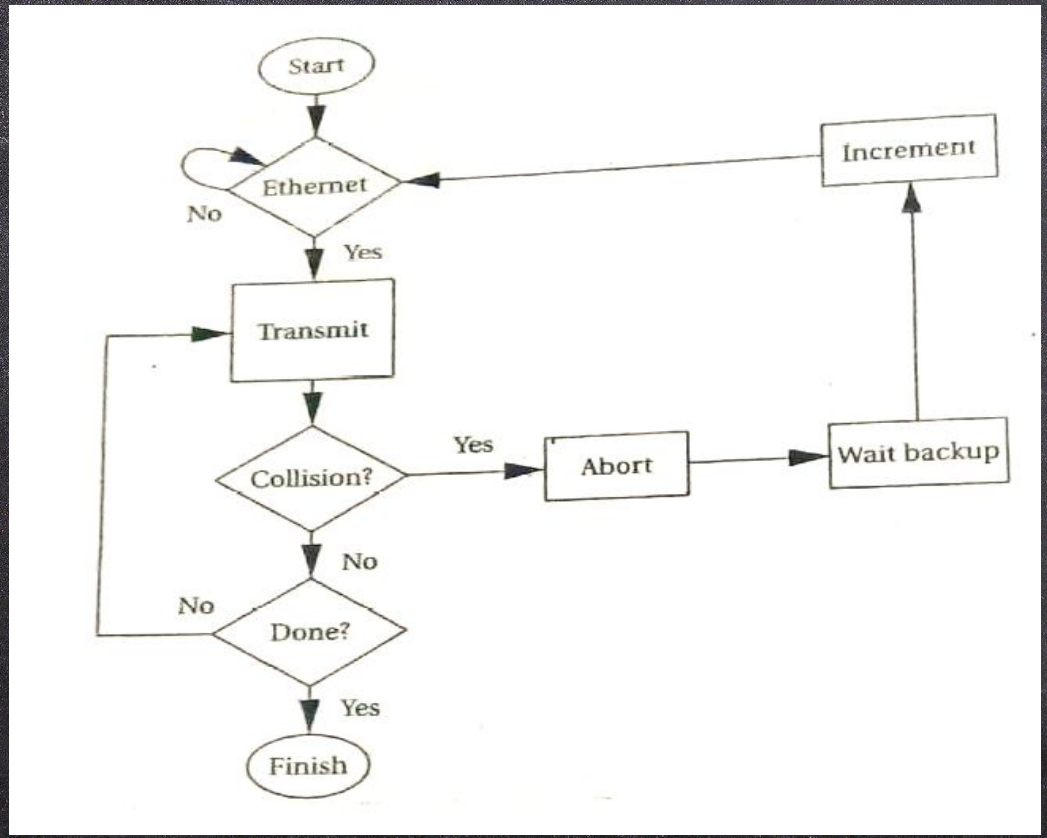


- Unlike the I^2C bus, nodes on the Ethernet are not syfichronized they can send their bits at any time.
- I^2C relies on the fact that a collision can be detected and quashed within a single bit time thanks to synchronization. But since Ethernet nodes are not synchronized if two nodes decide to transmit at the same time, the message will be ruined.

Ethernet (Short notes)

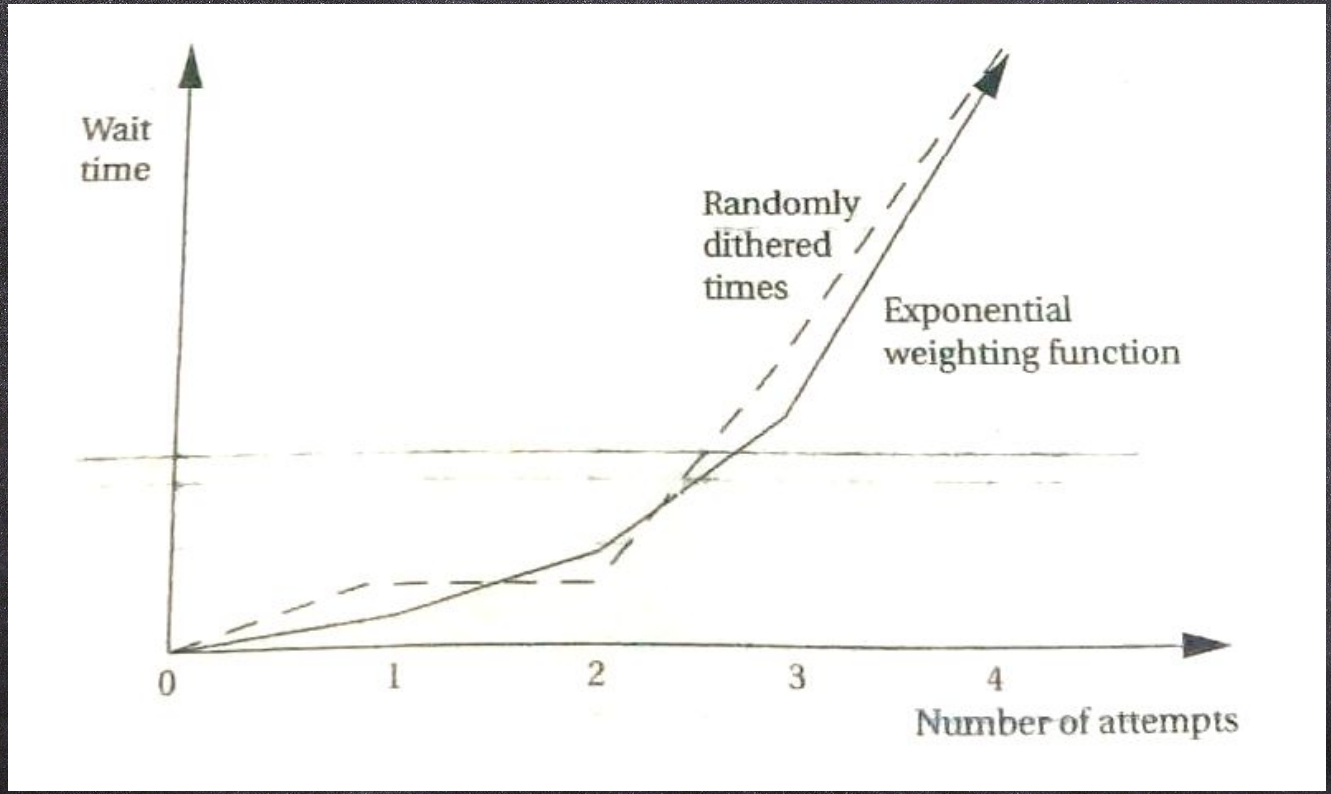
- The Ethernet arbitration scheme is known as Carrier Sense Multiple Access with Collision Detection (CSMA/CD)
- Ethernet CSMA/CD Algorithm : A node that has a message waits for the bus to become silent and then starts transmitting. It simultaneously listens, and if it hears another transmission that interferes with its transmission, it stops transmitting and waits to retransmit. The waiting time is random, but weighted by an exponential function of the number of times the message has been aborted.
- Exponential Backoff Technique : Since a message may be interfered with several times before it is successfully transmitted, the exponential backoff technique helps to ensure that does not become overloaded at high demand factors. The random factor in the wait time minimizes the chance that two messages will repeatedly interfere with each other.
- The maximum length of an Ethernet is determined by the nodes ability to detect collisions. The worst case occurs when two nodes at opposite ends of the bus are transmitting simultaneously. For the collision to be detected by both nodes, each node's signal must be able to travel to the opposite end of the bus so that it can be heard by the other node. In practice, Ethernets can run up to several hundred meters.

Ethernet (Short notes)



The Ethernet CSMA/CD algorithm

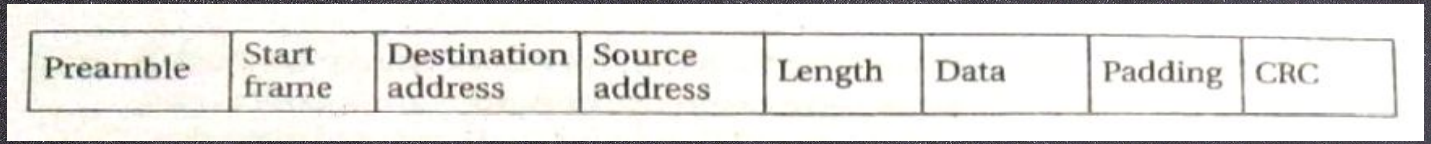
Ethernet (Short notes)



Exponential backoff times

Ethernet (Short notes)

- the basic format of an Ethernet packet is given below.



Preamble	Start frame	Destination address	Source address	Length	Data	Padding	CRC

- It provides addresses of both the destination and the source. It also provides for a variable-length data payload.

- The fact that it may take several attempts to successfully transmit a message and that the waiting time includes a random factor makes Ethernet performance difficult to analyze.

- It is possible to perform data streaming and other real-time activities on Ethernets, particularly when the total network load is kept to a reasonable level, but care must be taken in designing such systems.

Ethernet (Short notes)

- Ethernet was not designed to support real-time operations. the exponential backoff scheme cannot guarantee delivery time of any data.
- Because so much Ethernet hardware and software is available, many different approaches have been developed to extend Ethernet to real-time operation;
- some of these are compatible with the standard while others are not.
- There are 3 ways to reduce the variance in Ethernet packet delivery time:
  - suppress collisions on the network,
  - reduce the number of collisions,
  - or resolve collisions deterministically.
- Fieldbus (http://www.fieldbus.org) is a set of standards for industrial control and instrumentation systems.

Ethernet (Short notes)

CHAPTER 8.3

NETWORK-BASED DESIGN

Network based Design (Short notes)

- Designing a distributed embedded system around a network involves some of the same design tasks we faced in accelerated systems.
- We must schedule computations in time and allocate them to PEs.
- Scheduling and allocation of communication are important additional design tasks required for many distributed networks.
- Many embedded networks are designed for low cost and therefore do not provide excessively high communication speed.
- In this section we concentrate on design tasks unique to network-based distributed embedded systems.

Network based Design (Short notes)

- We know how to analyze the execution time of programs and systems of processes on single CPUs, but to analyze the performance of networks we must know how to determine the delay incurred by transmitting messages.

- Let us assume for the moment that messages are sent reliably—we do not have to retransmit a mes sage. The message delay for a single message with no contention (as would he the case in a point-to-point connection) can be modeled as

$$t_m = t_x + t_n + t_r$$

t_m - Message delay

t_x - Transmitter side Overhead

t_n - Network transmission time

t_r - Reciever side overhead

Network based Design (Short notes)

- In I^2C, t_x and t_r are negligible relative to t_n , as illustrated by Example 8.2 below

Example 8.2 : Simple message delay for an 1^2C message

Let's assume that our 1^2C bus runs at the rate of 100 KB/s and that we need to send one 8-bit byte. Based on the message format shown before, we can compute the number of bits in the complete packet:

$$n_{packet} = \text{startbit} + \text{address} + \text{data} + \text{stopbit}$$
$$= 1 + 8 + 8 + 1 = 18 \text{ bits}$$

The time required, then, to transmit the packet is

$$t_n = n_{packet} \times t_{bit} = 1.8 \times 10^{-4} s.$$

Network based Design (Short notes)

Some of the instructions in the transmitter and receiver drivers—namely, the loops that send bytes to and receive bytes from the network interface—will run concurrently with the message transmission. If we assume that 20 instructions outside of these loops are executed by the transmitter and receiver, overheads on an 8 MHz microcontroller would be as follows:

$$t_x = t_r = 20 \times 0.125 \times 10^{-6} = 2.5 \times 10^{-6}$$

The total message delay is:

$$tm = 2.5 \times 10^{-6} + 1.8 \times 10^{-4} + 2.5 \times 10^{-6} = 1.85 \times 10^{-4}$$

// $t_m = t_x + t_n + t_r$

Overhead is <3% of the total message time in this case.

Network based Design (Short notes)

- If messages can interfere with each others in the network, analyzing communication delay becomes difficult. in general, because we-must-wait-for-the network to become available and then transmit the message, we can write the message delay as

$$t_y = t_d + t_m$$

t_m - Message delay

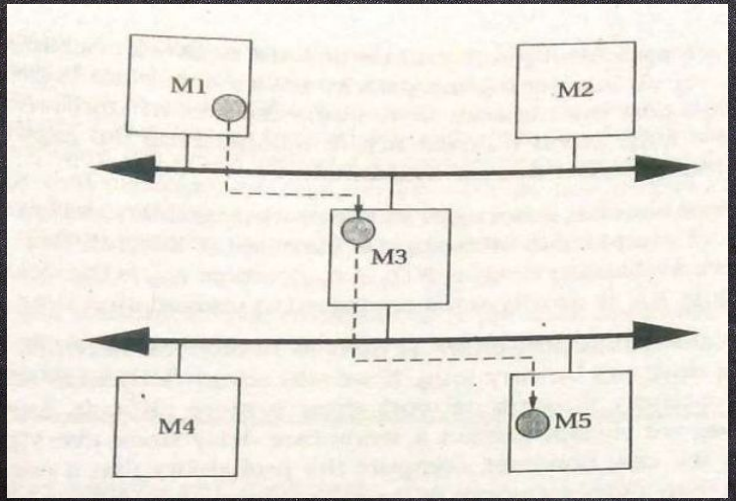
t_d - Network availability delay

- The main problem, therefore, is calculating t_d. That value depends on the type of arbitration used in the network.
    - If the network uses fixed-priority arbitration , the network availability delay is unbounded for all but the highest-priority device (since gets the network first)
    - If the network uses fair arbitration , the network availability delay is bounded.

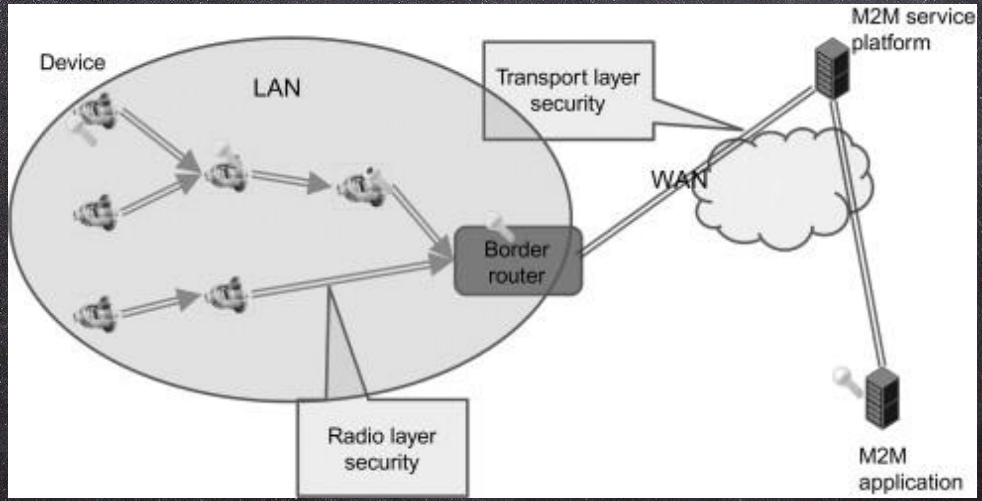
Network based Design (Short notes)

- Multihop Network
  - Multihop LANs involve transmitting data from a device through adjacent nodes in order to reach a final destination node

ORIGINAL



FOR REFERENCE ONLY



A multihop Communication

THANKS FOR WATCHING