

BASICS OF

I2C

PROTOCOL

(PART -1 & PART-2)



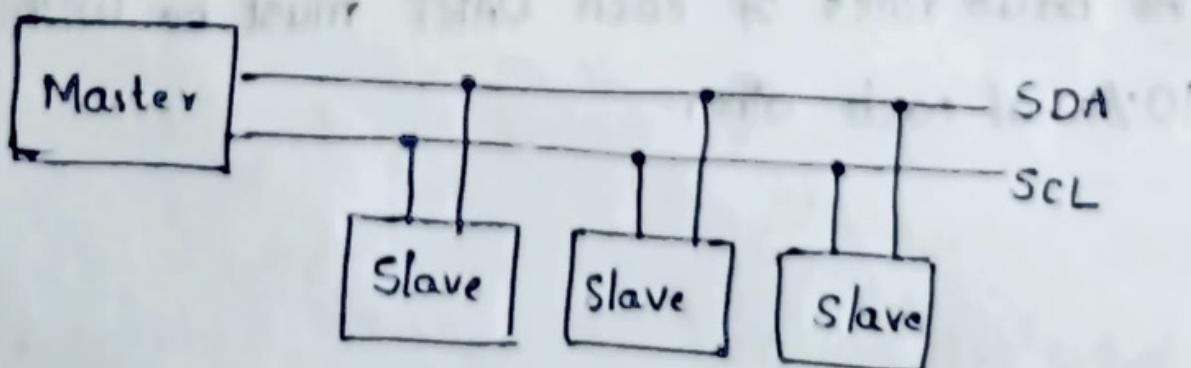
UMMIDI CHANDRIKA

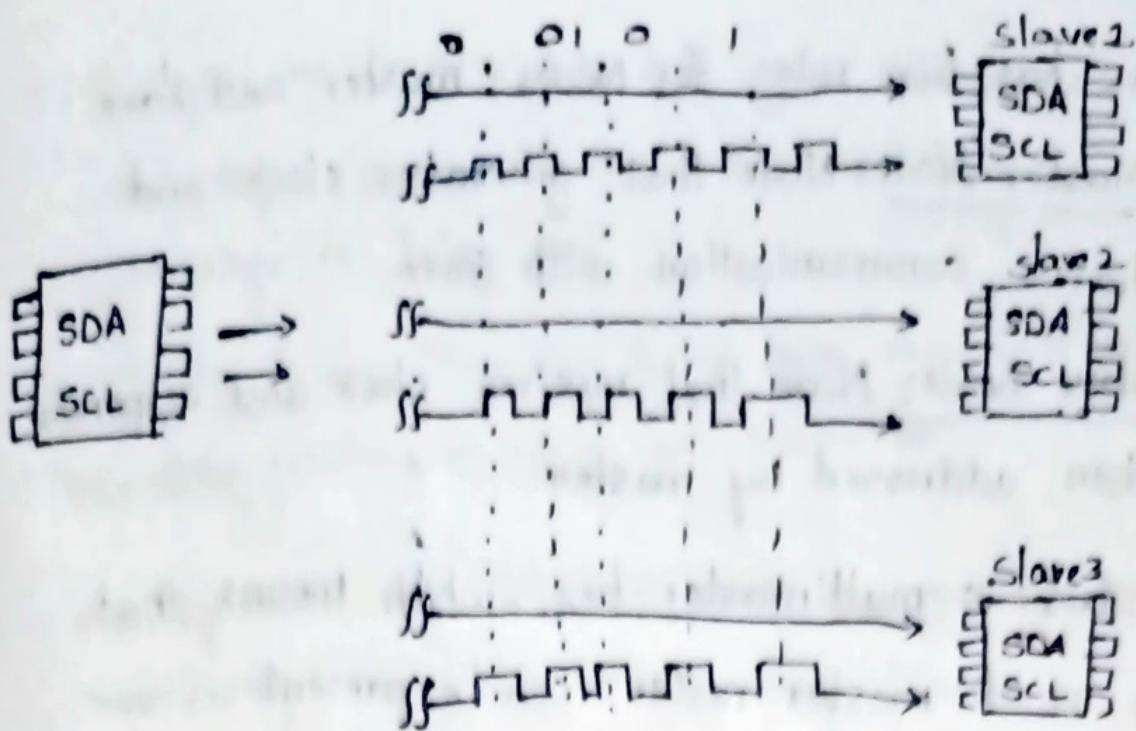
I²C (inter integrated circuit).

- introduced by philips company & NXP Semiconductor in 1972.
- Synchronous protocol [we use clock]
- it is multiple master & multi Slave.
- it is packet switched.
- Single ended.
- I²C is widely used for attaching lower-speed peripheral IC's to processors and microcontrollers in short distance, intraboard communication.
- A particular strength of I²C is the capability of microcontroller to control a network of device chips with just two general purpose I/O pins & software.
- Many other bus technologies used in similar application, such as SPI bus, require more pins & S/I's to connect multiple devices.

Design:

- I²C uses only two bidirectional open-drain lines. Serial data line (SDA) and serial clock line (SCL) pulled up with Resistors.





- The I²C bus drivers are "Open drain", means that they can pull corresponding s/l line low, but can't drive it high.
- Thus, there can be no bus contention where one device is trying to drive the line high while another tries to pull it low, eliminating the potential for damage to the drivers or excessive power dissipation in system.
- Each s/l line has a pull-up resistor on it, to restore s/l to high when no device asserting it low.
- The I²C reference design has 7 bit address space.
- The speed grades (Standard mode : 100kb/s, full speed : 400kb/s, fast mode : 1mb/s, high speed : 3.2mb/s) are maximum ratings.
- The maximal no.of. nodes limited by address space & also by total bus capacitance of 400pF, which restricts practical communication distance to few meters.

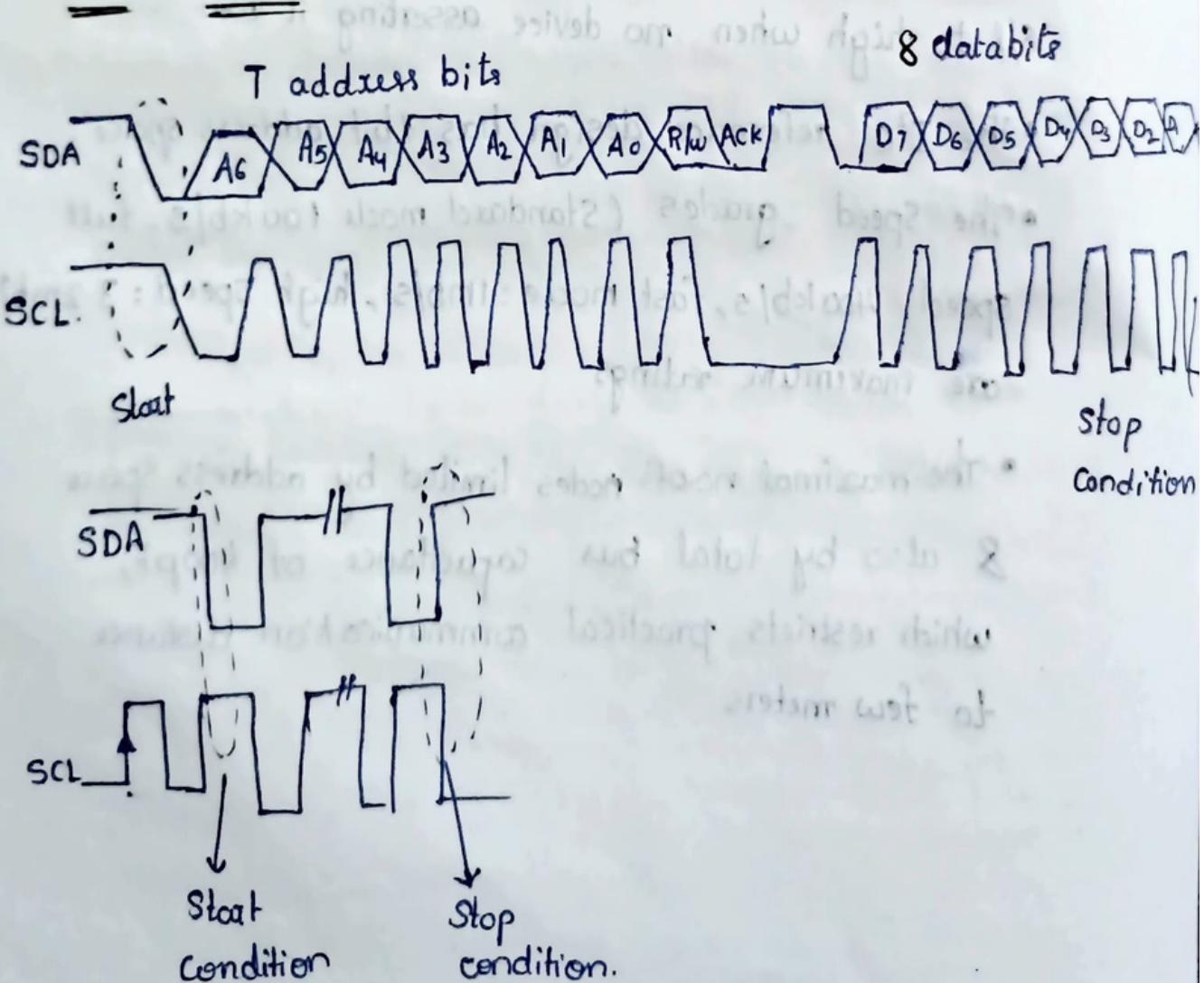
- The bus has two roles, for Nodes: master and slave.

→ Master Node: Node that generates clock and initiates communication with slave.

→ Slave Node: Node that receives clock and responds when addressed by master.

- The bus is multimaster bus, which means that any no. of master nodes can be present.
- Additionally, master & slave roles may be changed b/w messages (after STOP sent)
- There are 4 modes:
 - Master Transmit: Master Node sending data to slave
 - Master Receive: Master node receiving data from slave
 - Slave Transmit: slave node sending data to master
 - Slave Receive: Slave node receiving data from master.

Frame Format:



- Messages are broken into two types of frame:

→ Address frame: Master indicates slave to which message is being sent

→ Data frame: There are 8 bit data messages passed from master to slave or viceversa.

- Data placed on SDA line after SCL goes low, & is sampled after SCL line goes high.

- Start condition:

→ To initiate address frame, master device leaves SCL high and pulls SDA low.

→ This puts all slave devices on notice that a transmission is about to start.

→ If two master devices wish to take ownership of bus at one time, whichever pulls SDA low 1st wins race & gains control of bus.

- Address frame:

→ Address frame is always first in any new communication sequence.

→ For 7 bit address, address clocked out most significant bit first, followed by R/W bit indicating whether it is read (1) or write operation.

- NACK/Ack bit:

→ 9th bit of frame.

→ This is case for all frames (data or address)

→ Once, 1st 8 bits of frame are sent, receiving device is given control over SDA.

→ If receiving device does not pull SDA line low before 9th clock pulse, it is inferred that receiving device

- In that case, the exchange halts, and it's up to the master of the system to decide how to proceed.

- Data frame:

- After Address frame has been sent, data can begin being transmitted.
- The master will simply continue generating clock pulses at a regular interval, and data will be placed on SDA by either master or slave, depending on whether R/W bit indicated read/write operation.

- Stop Condition:

- Once all data frames have been sent, master will generate stop.
- Stop condition defined by $0 \rightarrow 1$ (low to high) transition on SDA after $0 \rightarrow 1$ transition on SCL, with SCL remaining high.

I₂C protocol:

1. The master initially in master transmit mode by sending START followed by 7-bit address of slave, it wishes to communicate with.
2. This is finally followed by single bit representing whether it wishes to write (0) to or read (1) from slave.
3. If slave exists on bus then it will respond with an Ack bit (active low for acknowledged) for that address.

4. the master then continues in either transmit or receive mode (according to read/write bit it sent), slave continues in complementary mode (receive or transmit respectively).
5. the address and data bytes are sent most significant bit 1st.
6. The master terminates a message with STOP condition if this is end of transaction or it may send another START condition to retain control of bus for another message.

Applications:

- Describing connectable devices via small ROM configuration tables to enable "plug and play" operation such as
- Accessing real-time clocks and NVRAM chips that keep user settings.
- Accessing low-speed DACs and ADCs.
- Controlling LCD displays.

Master: who will initiate transfer & gen. clk s/l to transfer
Slave: it will respond to master.

Two wired Communication: SDA, SCL
SDA, SCL

I2C Protocol - Part-2.

notes for A

Modes of Communication:

- The I2C peripheral has 4 basic operating modes to support data transfers as master and as slave.
 - if I2C peripheral is a master, it begins as master-transmitter and transmits an address for a particular slave. When giving data to slave, I2C peripheral must remain master-transmitter. In order to receive data from slave, I2C peripheral must be changed to master-receiver mode.
 - if I2C peripheral is a slave, it begins as slave receiver & sends acknowledgement when it recognizes its slave address from master. If master will be sending data to I2C, the peripheral must remain slave receiver. If master has requested data from I2C peripheral, the peripheral must be changed to slave-transmitter mode.

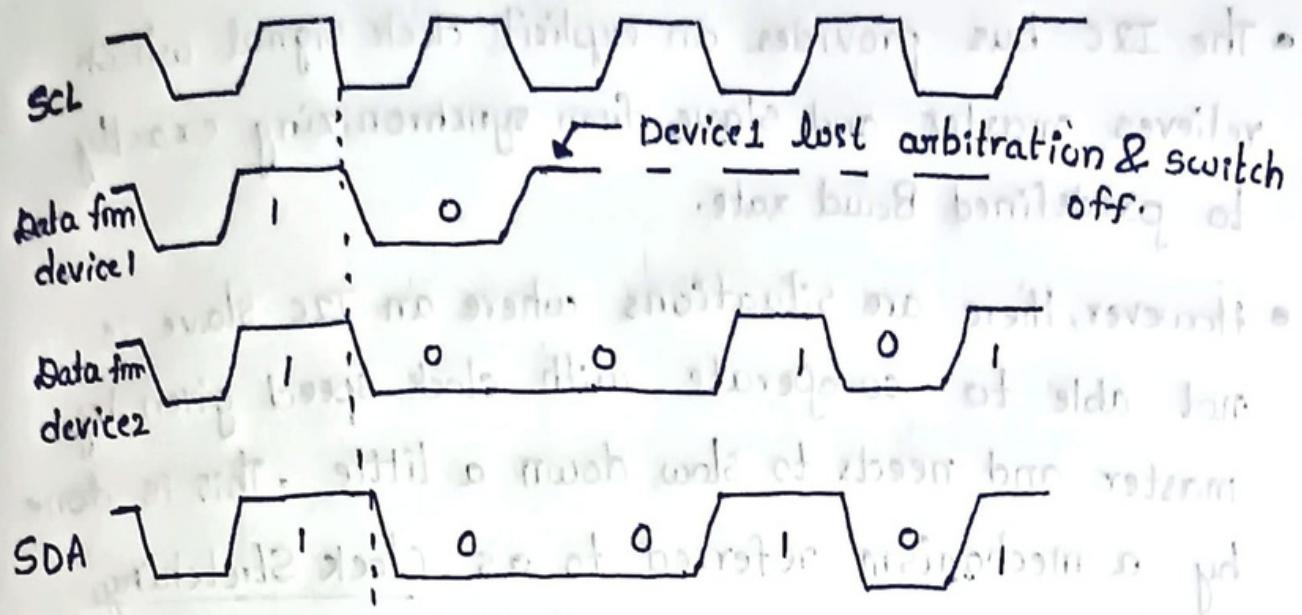
There are 4 modes:

- 1) Slave-Receiver mode.
- 2) Slave-Transmitter mode.
- 3) Master-Receiver mode.
- 4) Master-Transmitter mode.

Arbitration:

I2C - arbitration part

- If two or more master-transmitters simultaneously start transmission on same bus, an arbitration procedure is invoked.
 - The SDA and SCL are monitored by masters.
 - If SDA found high when it was supposed to be low it will be inferred that another master is active & hence it stops transfer of data.
 - The arbitration procedure gives priority to the device that transmits serial data stream with lowest binary value. Should two or more devices send identical bytes, arbitration continues on subsequent bytes.
 - if I2C peripheral losing master, it switches to slave-receiver mode, set arbitration lost (AL) flag and generates Arbitration-Lost interrupt.
 - if during serial transfer, the arbitration process is still in progress when repeated start condition or stop condition is transmitted to I2Cx-SDA, the master-transmitters involved must send repeated START condition or STOP condition at same position in format frame.
- Arbitration is not allowed b/w M & S
- Repeated START condition & data bit
 - STOP condition & data bit.
 - Repeated START condition & STOP condition.

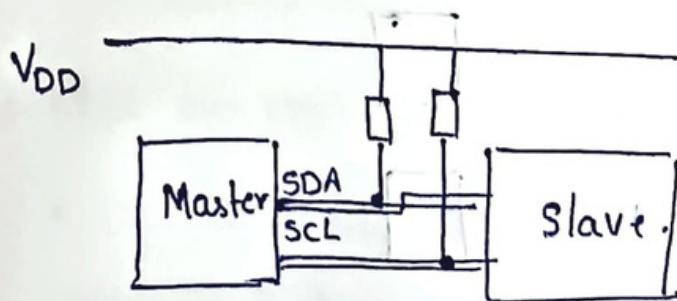


Clock Synchronization:

- The I2C does not synchronize devices to the predefined baud rate. It is master device which controls clock speed. So, master device determine clock speed.

However, Slave on I₂C may not co-operate with clock speed given by master & might needs to slow down.

- This is done by using clock synchronization.
- clock synchronization is performed using wired-AND connection of devices on SCL line.



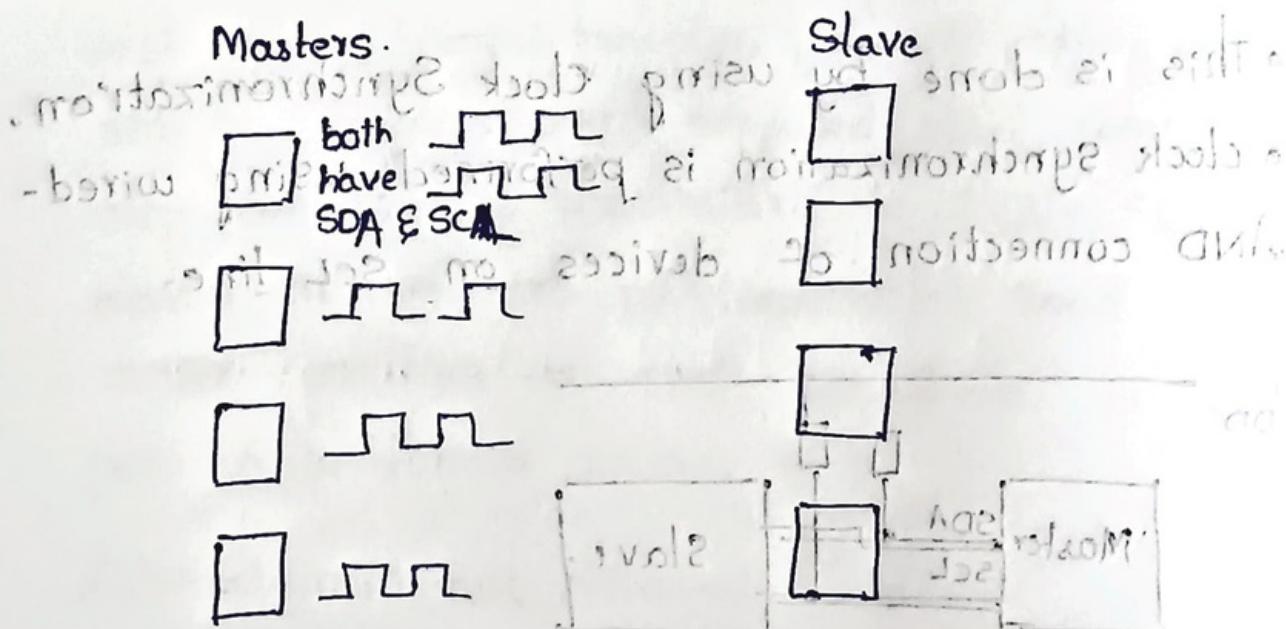
Clock Stretching:

- In an I₂C communication, the master device determines clock speed.

- The I₂C bus provides an explicit clock signal which relieves master and slave from synchronizing exactly to predefined Baud rate.
- However, there are situations where an I₂C slave is not able to co-operate with clock speed given by master and needs to slow down a little. This is done by a mechanism referred to as Clock Stretching.
- An I₂C slave is allowed to hold down clock if needs to reduce bus speed. The master, on other hand, is required to read back clock signal after releasing it to high state & wait until line has actually gone high.

Interview Questions:

Q) What if more than one master & slaves are there? and how data is transferred?



→ By using Arbitration Concept, we generate SDA and SCL lines in random, then any two can't be same, so based on SDA and SCL data transferred & specific master is selected.

2) By using AND gate concept also.

Arbitrate: if there are multiple masters.

Master 1: 1 1 0 1 0 }

Master 2: 1 0 1 0 1 }

Master 3: 1 0 1 1 0 }

AND/NAND

operations.

1 0 (M₁ lost bcoz sequence not matching)

1 0 (M₃ will lost)

M₂ will win bus further transfer

done by Master 2.

2. What are standard bus speeds in I₂C?

Mode	Speed
Standard Speed mode	100 kbps.
Fast mode	400 kbps.
Fast mode plus	1 mbps.
High Speed mode	3.4 mbps.
Ultra-high Speed mode	5 mbps.

3. What are main features in I₂C Communication?

- I₂C is multi-slave and multi-master communication protocol.

- I₂C supports Multi-master Arbitration Management.

- I₂C devices are open drain.

- I₂C bus is wired AND bus.

4) How many slaves can be connected to I₂C?

- Theoretically, we can connect 127 devices for 7-bit address & 1024 address devices will be connected for 10-bit address bus.
- But practically maximum no. of slaves on I₂C bus is limited by address space & also by bus capacitance.

Advantages of I₂C:

- 1) Flexibility.
- 2) Better error handling capacity.
- 3) important features like Arbitration, Clock Synchronization, Clock Stretching.
- 4) Adoptable.

Disadvantages of I₂C:

1. Half duplex communication, so data transmitted only in one direction at time.
2. Debugging an I₂C bus is difficult.
3. Pull-up resistors limit bus speed.