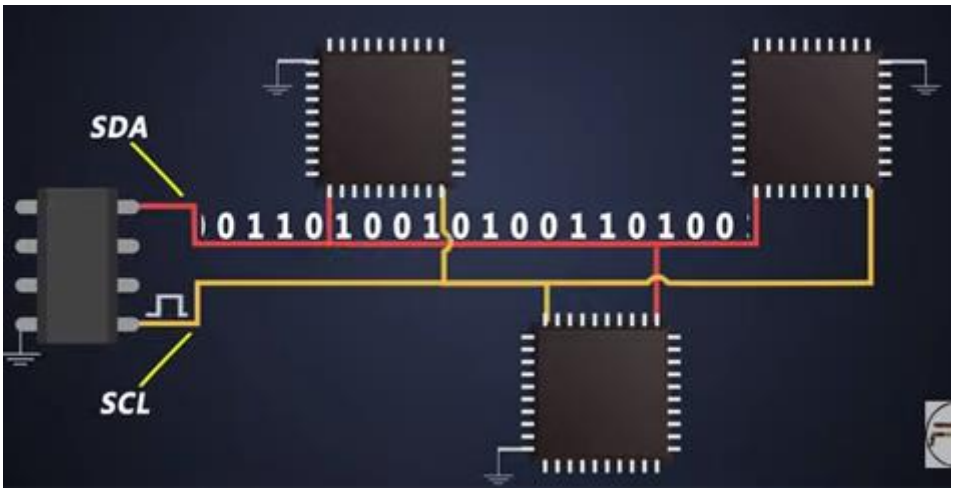
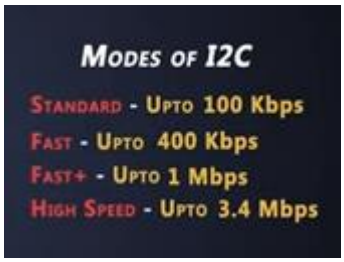
I2C protocol: (inter-integrated circuit)

- It is a synchronous type protocol, also called as two wire communication
- It is invented by Philips
- It is a bus topology-based protocol mainly used for communication between various ICs connected on the same PCB



- It is a multi-master communication protocol
- It can have more than two master can exist and clock is controlled by a master
- Only one master can access the bus at a time
- It is a half-duplex communication protocol since there is only one line for communication to transfer and receive the data
- In this protocol every slave has a unique slave address…when master want to talk to any of the slave 1st it will address the receiver that means it will call the receiver
- Based upon these modes whichever is suitable for all slaves the programmer will configure the I2C bus speed
- Every slave and master device have an open drain /open collector configuration pin these are connected to both SDA and SCL lines which are further connected to 5V/3.3V supply line depending upon VDD of the devices through an external resistor
- When the devices need to talk, they pull these I2C bus lines to ground which is known as logic level low by turning on this internal FET.
- when bus is pulled low by a certain device then every other device which is present on the bus understands the bus is pulled to ground, this is 0 for the devices and when internal mosfets are turned off, the I2C bus again goes to 5V or we can say logic level high and this is considered as 1

- the value of these resistors changes as per the I2C modes around 2Kohm resistors used for 400kbps speed



SDA	SCL	Condition of I2C bus
5V	5V	Idle (no data)

- The two wires that are connecting the devices are

1.SDA (data line): it sends the data to other devices

2.SCL (clock line): it is responsible for synchronization

Data frame:

It consists of:

1.start bit

2.slave address (to which master need to send data 7/10-bit address)

3.read/write signal (write=0, read=1)

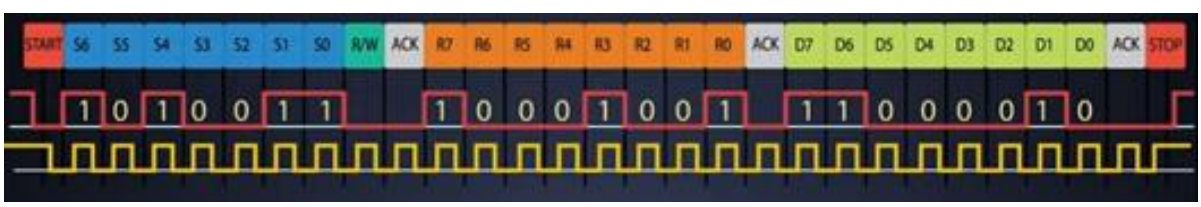
4.acknowledgement signal (sends by slave to master)

5.address of the internal register of slave (from which master wants to write or read)

6.acknowledgement signal (master to slave)

7.8-bit data followed by final ack signal

8.stop bit (to notify slave that data transmission has ended)



NACK:

If slave will not give an ack to master (negative ack).so when master receives NACK it can generate either stop bit or start from the top

BUS ARBITRATION:

- Arbitration is a process of solving a conflict between 2 or more masters who are trying access the bus simultaneously
- If two are more masters are communicating on a single bus at a time then then slaves will not be able to understand any of the information. So, to avoid that bus arbitration is used.
- This arbitration is achieved based on the slave address, can be achieved by monitoring SDA lines after sending data on the bus.
- Whichever slave has a lower address will get early access to talk over the bus
- If a master sends a logic high signal but detects low on SDA due to priority of the low slave address then naturally it loses the control over the bus and switches back into the slave mode so the other master that sends low on bus will wins and continues the transmission

CLOCK STRETCHING

- When a master wants to transmit data but the slave is not ready to receive it.
- **Example**: when some data is written on the EEPROM IC by a master it will save this data and if the master issues a read command immediately after write, then EEPROM IC can tell the master to wait while it is saving this data, here the clock stretching helps
- The SCL line is stretched by the slave which indicates the master that the slave is busy. this status of the clock is monitored by transmitter.
- The I2C slave has every right to hold down the bus speed, the master on the other hand has to read the clock signal after releasing the SCL to high state

ADVANTAGES OF I2C:

1.requires two connecting wires only

2.multi master compatibility

3.better error handling – ACK/NACK

4.Different mode flexibility for slave

DISADVANTAGES OF I2C

1.**Less flexibility for slave address** (because the slave address is fixed for particular device which is given by the manufacturer)

Example: if I want to use 2 accelerometers of same manufacture for motion sensing but the slave addressing doesn't allow me to use same device

2.**complexibility of firmware increases** because the master device has to support features like bus arbitration and clock stretching

3.**increased data overhead** – for 8bits of data need to send 1 start bit, I stop bit, ACK bits, R/W bit and 7-bit slave address, this increases data overhead for 8-bits to send the data, which reduces data throughput (in UART, SPI we don't see such addresses or ACK/NACK bits)