

Parameter	Analog	Digital
Definition	Analog comm. is the technology which uses analog signal for the transmission of info.	Digital comm. is technology which uses digital signal for transmission of info.
Noise & distortion	Get affected by noise	Immune from noise & distortion
Error probability	Error probability is high due to Parallax	Error probability is low
Hardware	Hardware is complicated & less flexible than digital system	Hardware is flexible & less complicated than analog system
Cost	Low cost	High cost
Bandwidth requirement	Low bandwidth requirement	High bandwidth requirement
Power requirement	High power is required	Low power is required
Portability	Less portable as the components are heavy	More portable due to compact equipments
Modulation used	Amplitude & angle modulation	Pulse coded due to compact equipments
Representation of signal	Analog signal can be represented by sine wave	Digital signal is represented by square wave
Signal values	Consists of continuous values	Consists of discrete values
Eg of wave	Voice, sound etc	Digital signals are used in computers
<pre> graph LR A[Analog info source] --> B[Encoder] B --> C[modulator] C --> D[Channel] D --> E[Demodulator] E --> F[Decoder] F --> G[msg transducer] </pre>		

	Serial Comm.	Parallel Comm.
2. Data transmission speed	Slow	Comparatively fast
No. of comm. link used	Single	Multiple
No. of transmitted bit/clock cycle	only one bit	n no. of link will carry n bits
Cost	Low	High
Ground talk	Not present	Present
System up-gradation	Easy	Quite difficult
Mode of transmission	Full duplex	Half duplex
Suitable for	long distance	short distance
High freq. operation	more efficient	less efficient

3. ① Bit rate is defined as transmitting no. of bits per second.
- ② Bit rate is defined as per second travel no. of bits.
- ③ Bit rate emphasized on computer efficiency.
- ④ Bit = Baud rate \times rate no. of bit per band
- ⑤ Bit rate is used to decide the requirement of BW for transmission of signal.
- ⑥ Baud rate is defined as no. of signal units per second.
- ⑦ Baud rate is also defined as per second no. of changes in signal.
- ⑧ Baud rate emphasized on data transmission
- ⑨ Baud rate = $\frac{\text{bit rate}}{\text{no. of bit per band}}$
- ⑩ Baud rate is used to decide the requirement of BW for transmission of signal.

5. In RS232, RS stands for Recommended Standard. It defines the serial comm. using DTE & DCE signals. Handshaking involves 4 steps:

- ① Data terminal equipment (DTE) puts the RTS line into "On" state.
- ② The data comm. equipment (DCE) puts the CTS line into the "On" state.
- ③ The DTE puts the DTR line into the "On" state.
- ④ The DTR line remains in the "On" state while data is being transmitted.

After transmission of data is completed, the DTE puts the DTR & RTS lines into the "Off" state & the DCE puts the CTS line into the "Off" state.

With RS 232 comm. only will take place when both ends of the RS232 link are ready. Thus, the RS232 handshake process enables the DTE to request control of the communications link from a related modem & allows the modem to inform the terminal equipment that the control has been acquired.

6. DTE element

Data ^{trans}terminating ~~equipment~~ equipment is a device that sits between data terminal equipment & a data transmission circuit. It is also called data comm. equipment & data carrier equipment. Usually, the DTE device is the terminal & the DCE is a modem. A modem is the most common kind of DCE. Other common examples are ISDN adaptors, satellites, microwave stations, base stations & network interface cards.

DTE Element

Data terminal element is an end instrument that converts user information into signals or reconverts received signals. These can also be called tail circuits. A DTE ~~element~~ device communicate with ~~data~~ DCE.

7. Bit time = no. of bit / baud rate
Baud rate = 9600
 $\therefore 9600 \rightarrow 1.04 \text{ micro second}$

8. Serial peripheral interface (SPI) is an interface bus commonly used to send data b/w ~~etc~~ a small peripherals such as shift register, sensors, & SD cards. It uses separate clock & data lines, along with a select line to choose the device you wish to talk to.

Applications

1. SD card
2. RFID card reader module
3. Wireless transmitter/receiver

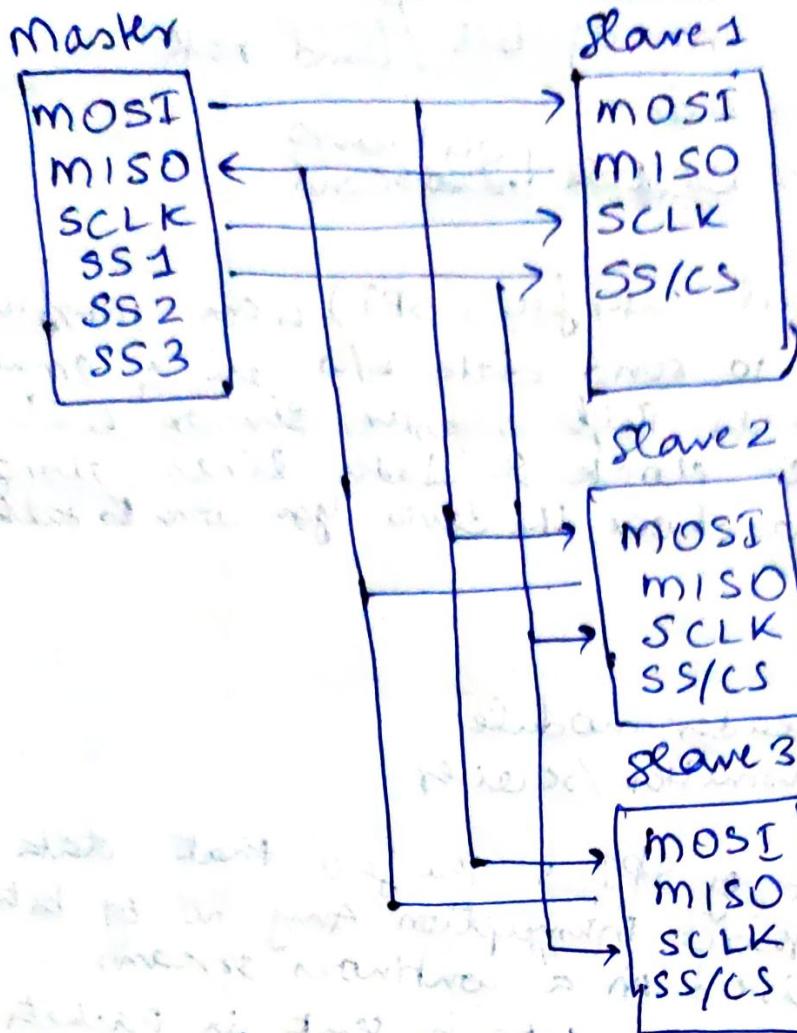
Features:

Unique benefit of SPI is the fact that data can be transferred without interruption. Any no. of bits can be sent or received in a continuous stream.

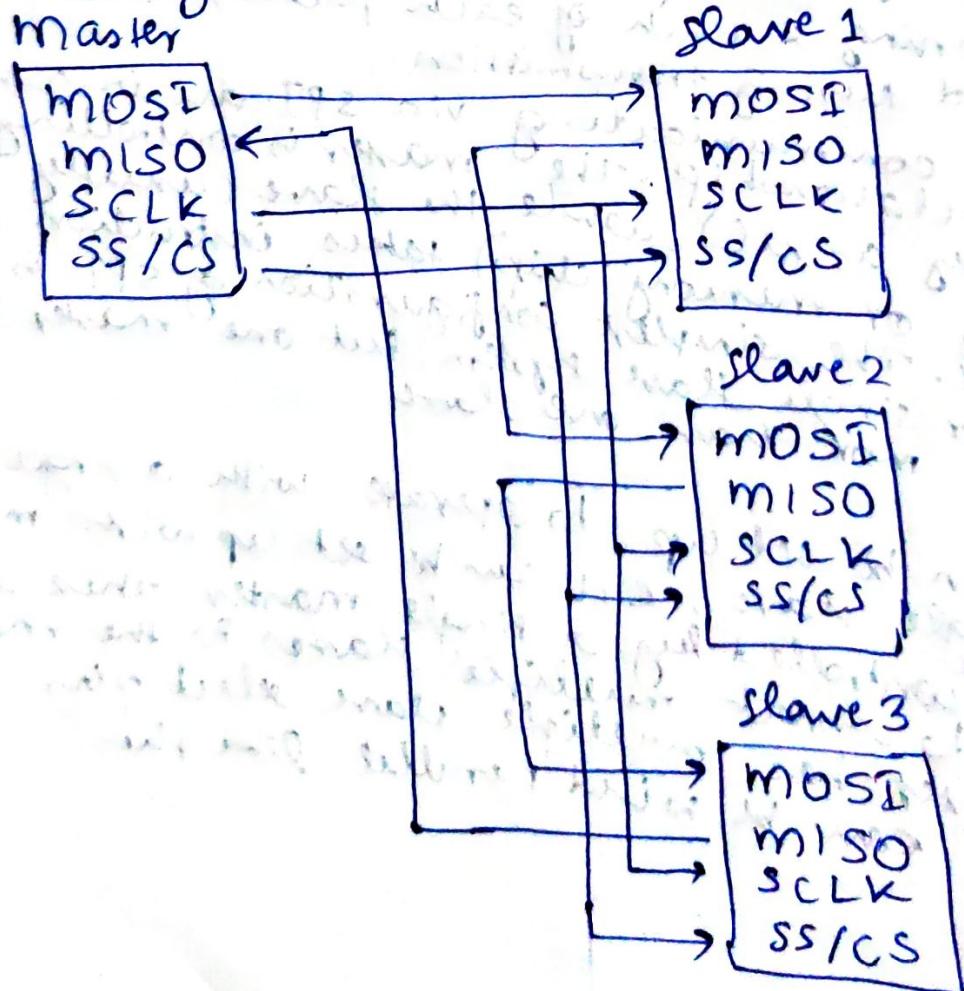
With I₂C & UART, data is sent in packets, limited to a specific no. of bits. Start & stop conditions define the beginning & end of each packet, so the data is interrupted during transmission.

Devices communicating via SPI are in a master-slave relationship. The master is controlling device (usually a microcontroller), while the slave (usually a sensor, display or memory chip) takes instruction from master. The simplest configuration of SPI is a single master single slave system, but one master can control more than one slave.

9. SPI can be set up to operate with a single master & a single slave, & it can be set up with multiple slaves controlled by a single master. There are 2 ways to connect multiple slaves to the master. If master has multiple slave select pins, the slaves can be wired parallel like this.



If only one slave select pin is available, the slaves can be daisy-chain like this

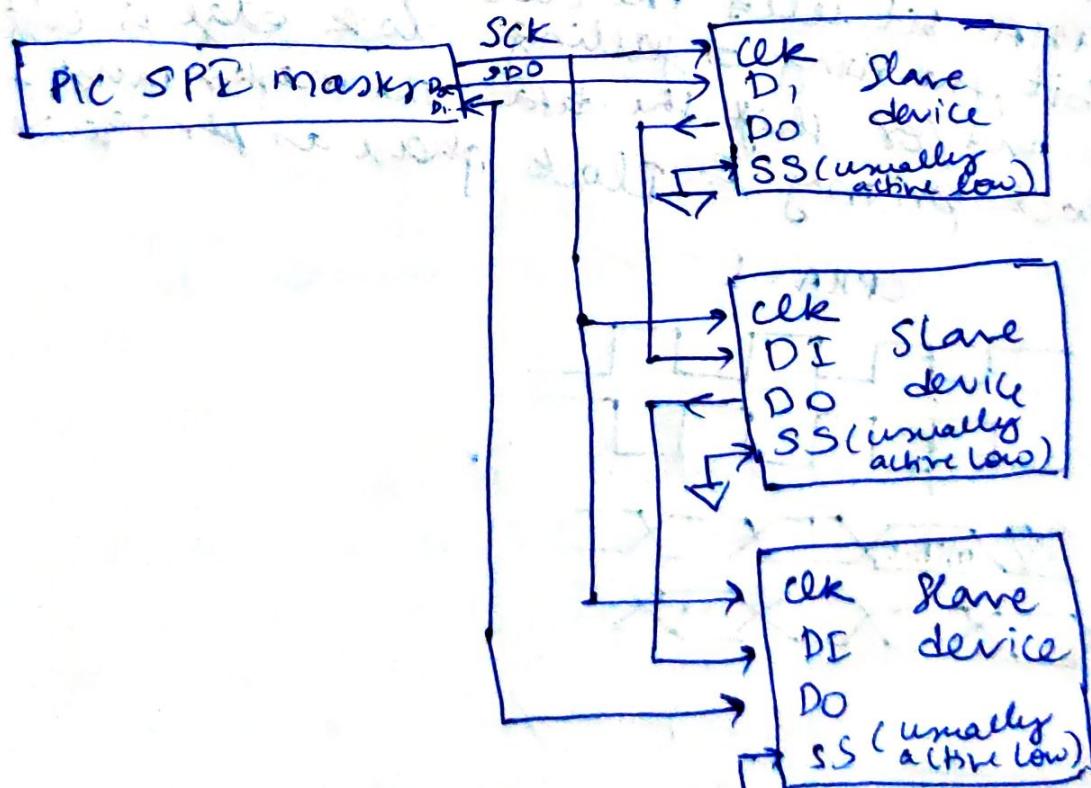


10. SPI interface bus is a high speed, 3 wire, serial comm. protocol (4 if u include SS_n) its primary purpose is to reduce on PCB wire routing by replacing the traditional parallel bus with a serial interface.

It works by transferring data one bit at a time b/w 2 devices with the master device sending the clock signal (SCK). The clock controls the timing of data transfer.

Data (MOSI [SDO]) is sent out of a shift register in the master SPI device along ~~the~~ with a clock signal (SCK) while at same time another shift register receives data from the slave (MISO, SDI).

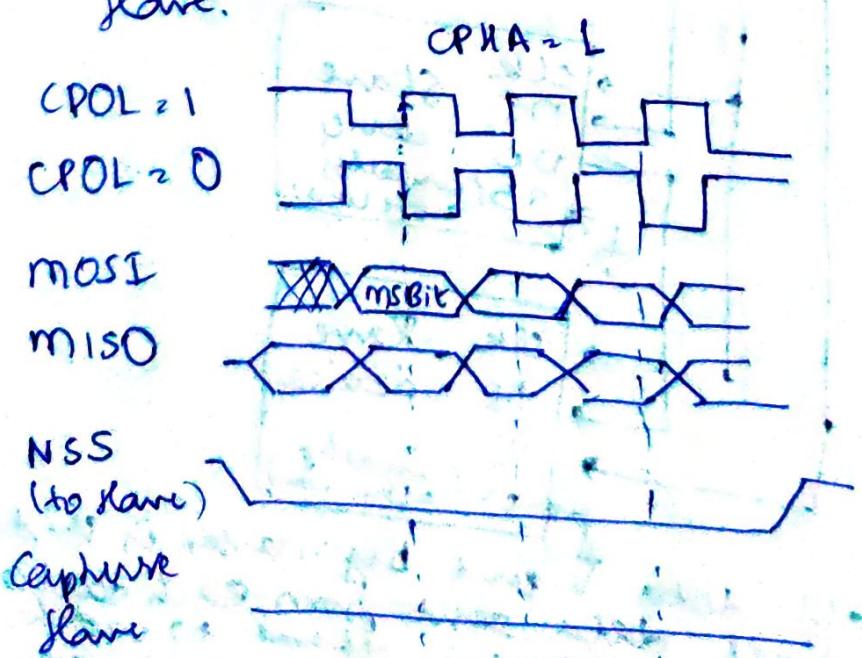
The master is always in control & initiates the data transfer using the clock signal. Slave devices are selected using a separate slave select signal that is software controlled ie those signals are separate from the SPI hardware module.



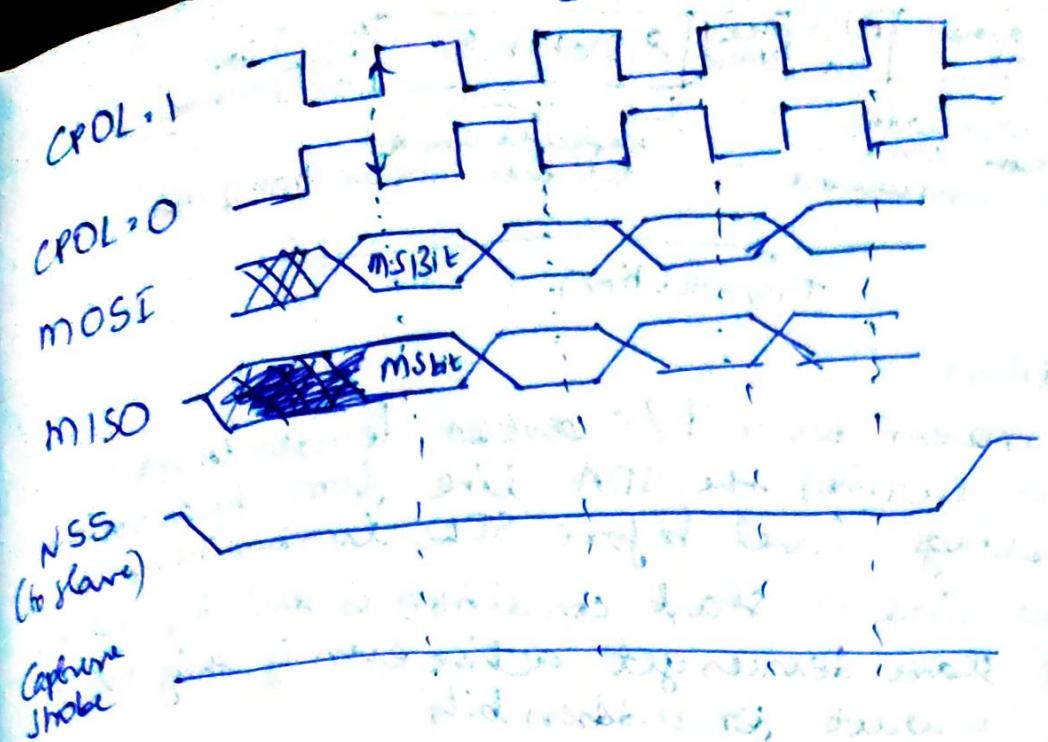
With this scheme all data sent by master is shifted into all devices and all data sent from each device is shifted out to the next. For this scheme to work you have to make sure that each slave uses the clock in the same way & you have to get the right no of bits, so there is more work to do in software.

11. SPI interface allows to transmit & receive data simultaneously on two lines (MOSI & MISO). Clock polarity (CPOL) & clock phase (CPHA) are the main parameters that define a clock format to be used by SPI bus. Depending on CPOL parameter, SPI clock may be inverted or non-inverted. CPHA parameter is used to shift the sampling phase. If $CPHA = 0$ the data are sampled on leading (first) clock edge. If $CPHA = 1$ the data are sampled on the trailing (second) clock edge, regardless of whether that clock edge is rising or falling.

In SPI, the master can select the clock polarity & clock phase. The CPOL bit sets the polarity of clock signal during the idle state. The idle state is defined as period when CS is high & transitioning to low at the start of transmission & when CS is low & transitioning to high at the end of transmission. The CPHA bit selects the clock phase. Depending on CPHA bit, the rising or falling clock edge is used to sample and/or shift the data. The master must select the clock polarity & clock phase as per requirement of slave.



CPHA = 1



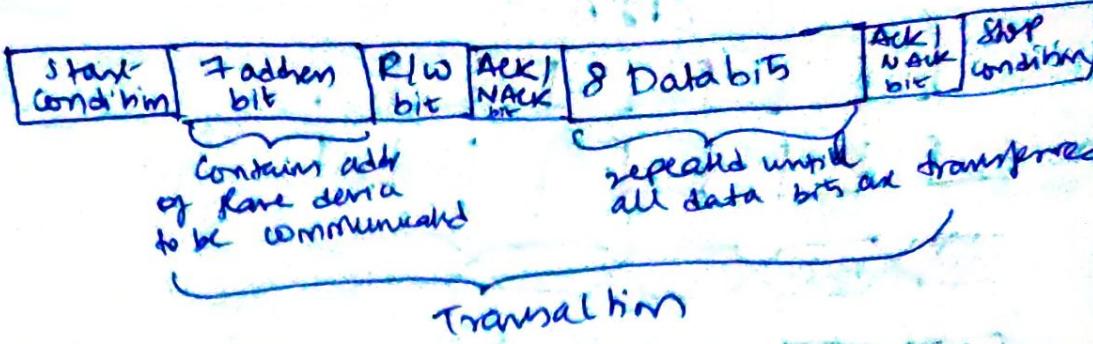
12. I₂C comm. is short form for inter-integrated circuits. It's a comm. protocol developed by Philips semiconductors for transfer of data b/w a central processor & multiple ICs on the same circuit board using just two common wires. Owing to its simplicity, it is widely adopted for comm. b/w microcontrollers & sensor arrays, displays, IoT devices, EEPROM etc.

This is a type of synchronous serial comm. protocol. It means that data bits are transferred one by one at regular intervals set by a reference clock line.

Features:

- Only 2 common bus lines (wires) are required to control any device/IC on the I₂C n/w.
- No need of prior agreement on data transfer rate like UART comm. so data transfer speed can be adjusted whenever required.
- Simple mechanism for validation of data transferred.
- Uses 7 bit addressing system to target a specific device/IC on I₂C bus.
- I₂C n/w's are easy to scale. New devices can simply be connected to 2 common I₂C bus lines.

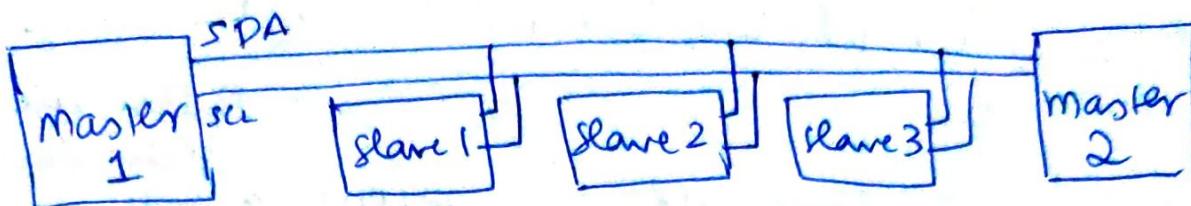
13.



- Start Condition
Whenever a master device P/C devices decides to start a transaction, it switches the SDA line from high voltage to low voltage level before SCL line switches from high to low. Once a start condition is sent by master device, all slave devices get active even if they are in sleep mode & wait for address bits.
- Address block
It comprises of 7 bits are filled with address of slave device to /from which master device needs send/receive data. All slave devices on I2C bus compare these address bits with their address.
- Read / write bit
This bit specifies the direction of data transfer. If master device / IC need to send data to slave device, this bit is set to 0. If master IC needs to receive data from slave device, it is set to 1.
- Ack / NACK bit
Stands for acknowledgement / not-acknowledgement bit. If physical address of any slave device coincides with address broadcasted by master device, the value of this bit is set to 0 by slave device. Otherwise it remains at logic 1.
- Data block
Comprises of 8 bits & they are set by sender, with data bits it needs to transfer to the receiver. This block is followed by an Ack / NACK bit & set to 0 by receiver if it successfully receives data. Otherwise it stays at logic 1.
This combination of data block followed by Ack / NACK bit is repeated until the data is completely transferred.

Stop condition
After required data blocks are transferred through SDA line, the master device switches the SDA line from low voltage level to high voltage level before SCL line switches from high to low.

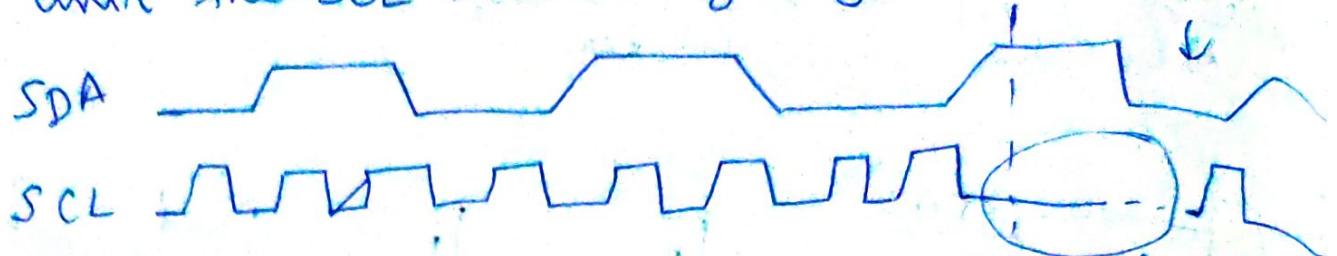
14.



I₂C is designed for multiplexed master purpose this means that more than one device can initiate transfers. During the transfer, the masters constantly monitor SDA & SCL. If one of them detects that SDA is low when it should actually be high, it assumes that another master is active & immediately stops the transfer. This process is called arbitration.

For example, Master 1 issues a start sequence & sends an address, all slaves will listen, including Master 2 which at the time, is considered a slave as well. If address does not match Master 2 withdraws its transactions until the bus becomes idle. The structure of bus is wired-AND, this means if device pulls the bus line low then the line stays low. When Master 2 changes the state of line to high then bus line does not go high when the bus is occupied already by some other device which has put the bus low such as Master 1. Thus master does not get its data on bus. For as long as there has been no STOP sequence present on bus, it won't touch the bus & leaves the SCL & SDA line alone. If master cannot make data line to go high then it loses the arbitration & needs to go back off & wait until the stop sequence is seen. Later it can check the line & make another attempt when line is free.

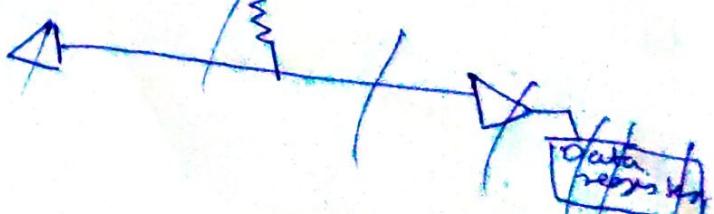
15. In I₂C comm. can be paused by clock stretching by holding the SCL line low & it cannot continue until the SCL released high again



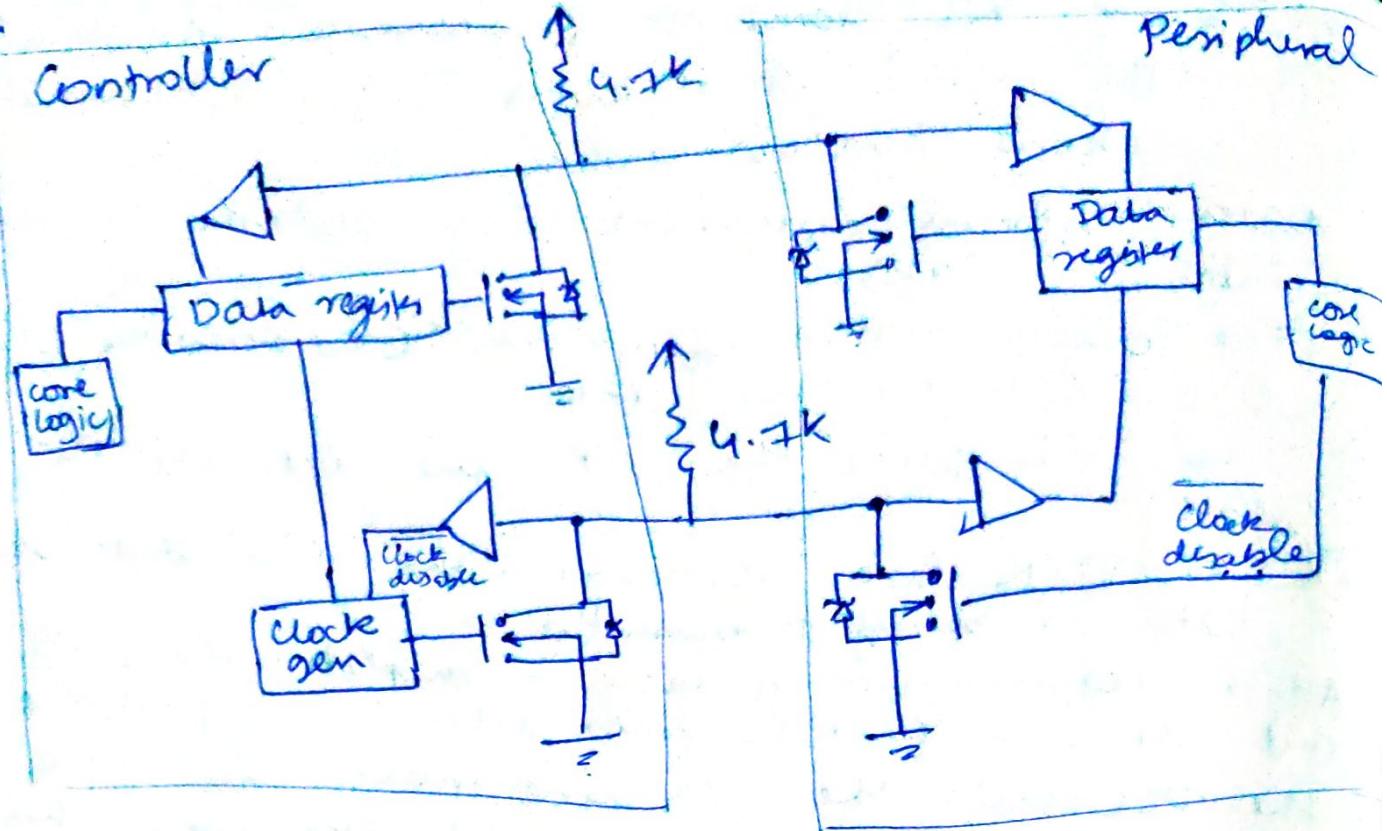
In I₂C slave able to receive a byte of data on the fast rate but sometimes slave takes max time in processing the received bytes in that situation slave pull pull the SCL line to pause the transaction & after the processing of the received ~~data~~ bytes, it again released the SCL line high again to resume the communication.

The clock stretching is the way in which slave drive the SCL line but it is the fact, most of the slave do not drive the SCL line

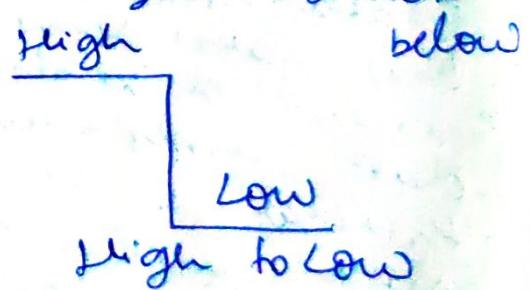
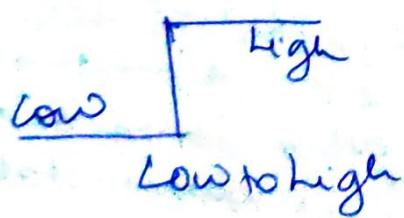
17. In I₂C bus drivers are "open drain" meaning that they can pull the corresponding signal line low, but cannot drive it high. Thus, there can be no bus contention where one device is trying to drive the line high while another tries to pull it low, eliminating the potential for damage or damage to drivers or excessive power dissipation in system. Each signal line has a pull-up resistor to restore the signal to high on it, device is asserting it low when no



117



18. In a digital circuit the pins are always either 0 or 1. In some cases, we need to change the state from 0 to 1 from 1 to 0. In either case, we need to hold the digital pin either 0 & then change the state to 1 or we need to hold it 1 & then change to 0. In both cases, we need to make the digital pin either 'High' or 'Low' but it can't be left floating.
- So, in each case, state gets changed as shown



A pull up resistor is used to make the default state of digital pin as high or to the logic level and a pull down resistor does exactly opposite, it makes the default state of digital pin as low (0V)

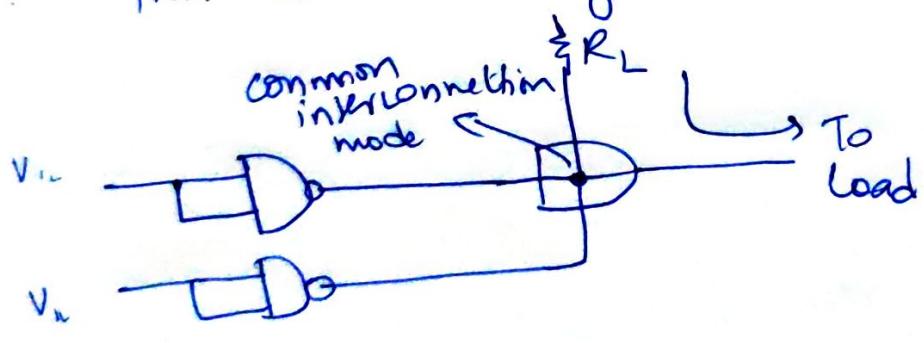
1b. I₂C is synchronous communication in which clock is always generated by master & this clock is shared by both master & slave.

In case of multi master, all generate their own SCL clock, hence it is necessary that clock should be synchronized by wired AND logic



Electrical interconnection beyond the wires is a single 4.7 kΩ pull up resistor for each of the I₂C bus lines. Consequently, a slave device writing to SDA only needs to pull the ~~resistor~~ line low for each '0' written, otherwise, it leaves the line alone to write a '1', which occurs due to lines being high externally by pull up resistor.

This is commonly known as wired AND config.



i/p i/p - writing		C (o/p)
A	B	
0	0	0
0	1	0
1	0	0
1	1	1