IMT
SCHOOL

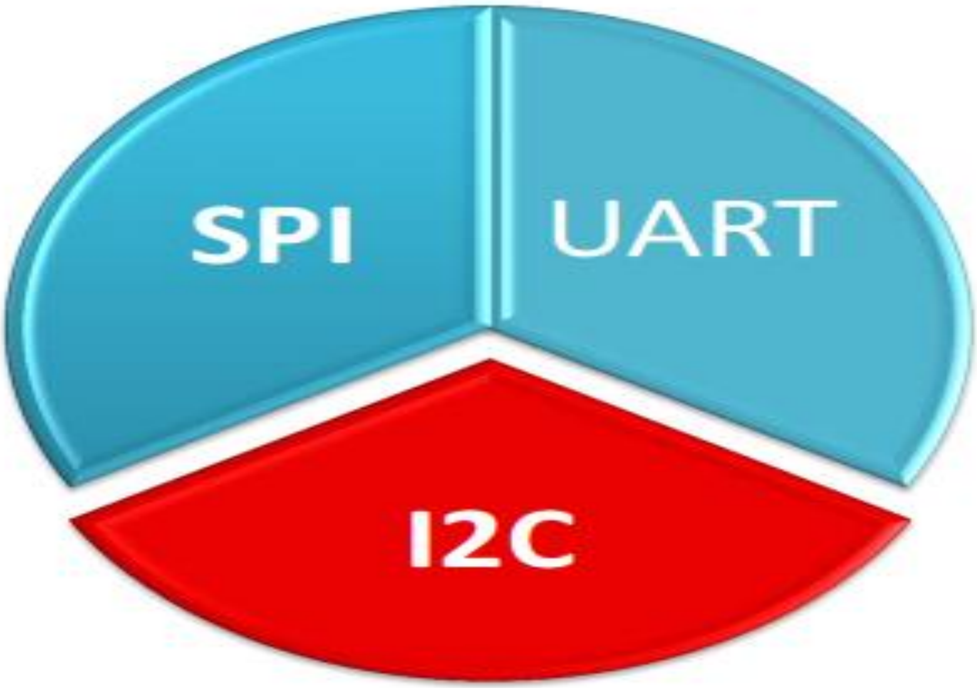
Embedded Systems Interfacing

Lecture Fifteen

I2C Serial Communication

This material is developed by IMTSchool for educational use only
All copyrights are reserved

Communication protocols

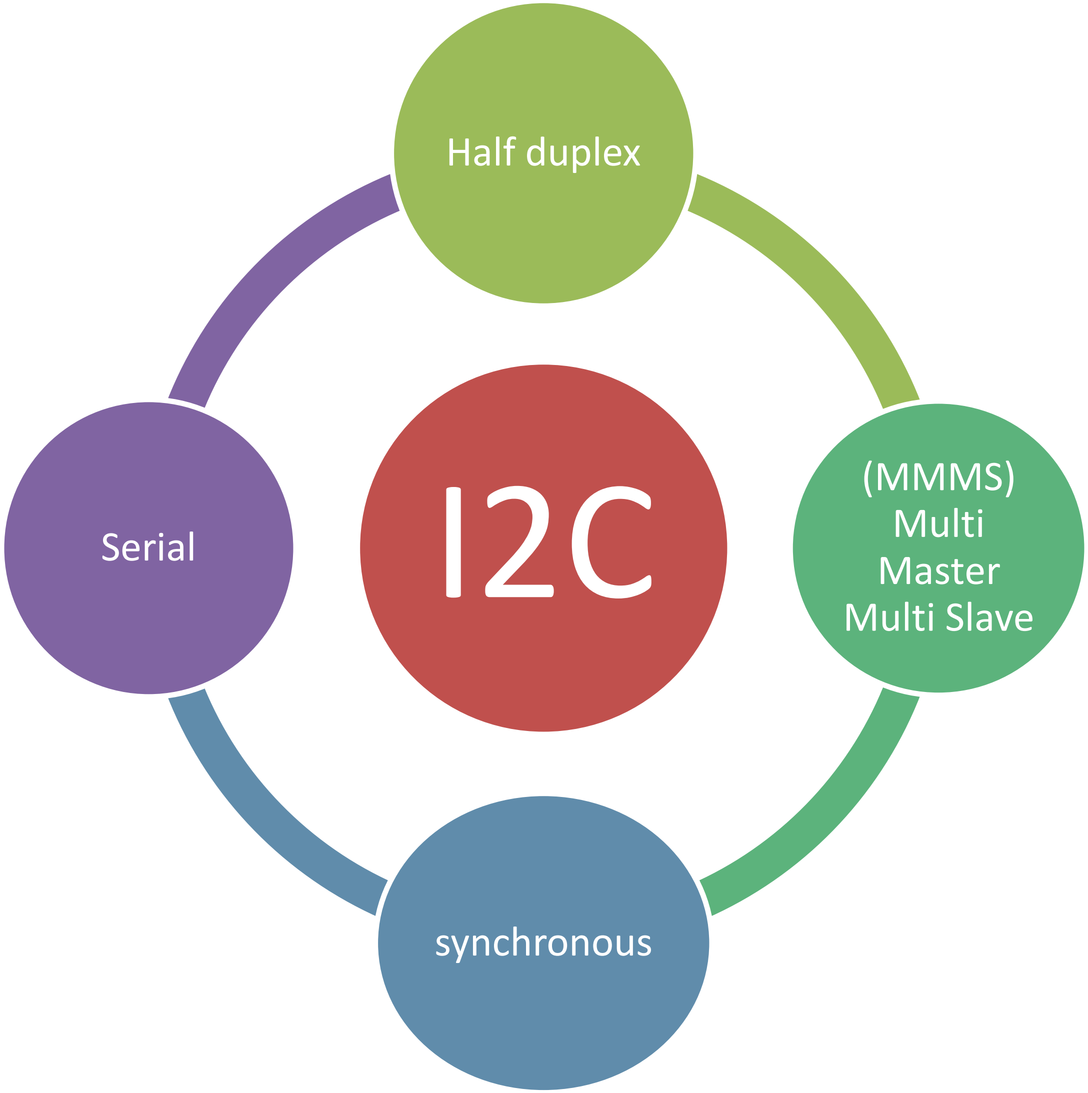


I2C concept

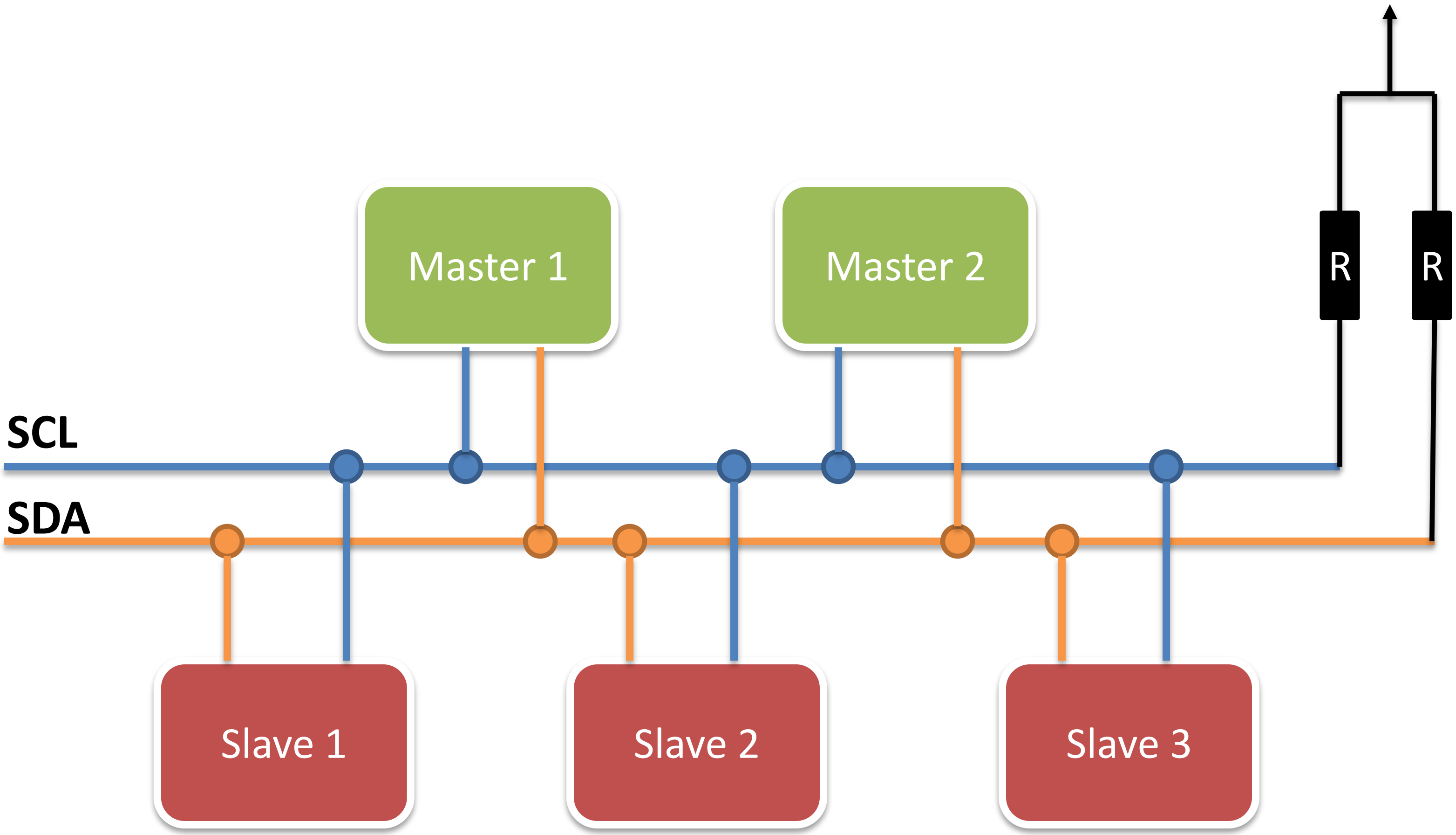
What is I2C?

- **Inter-Integrated Circuit** **I²C** (Pronounced I Two C or I Squared C), is a serial communication protocol at which the devices are hooked up to the I2C bus with just ***two wires***.
- It is sometimes referred to as ***Two Wire Interface***, or the ***TWI***
- Devices could be the CPU, IO Peripherals like ADC, or any other device which supports the I2C protocol.
- All the devices connected to the bus are classified as either being ***Master*** or ***Slave***.

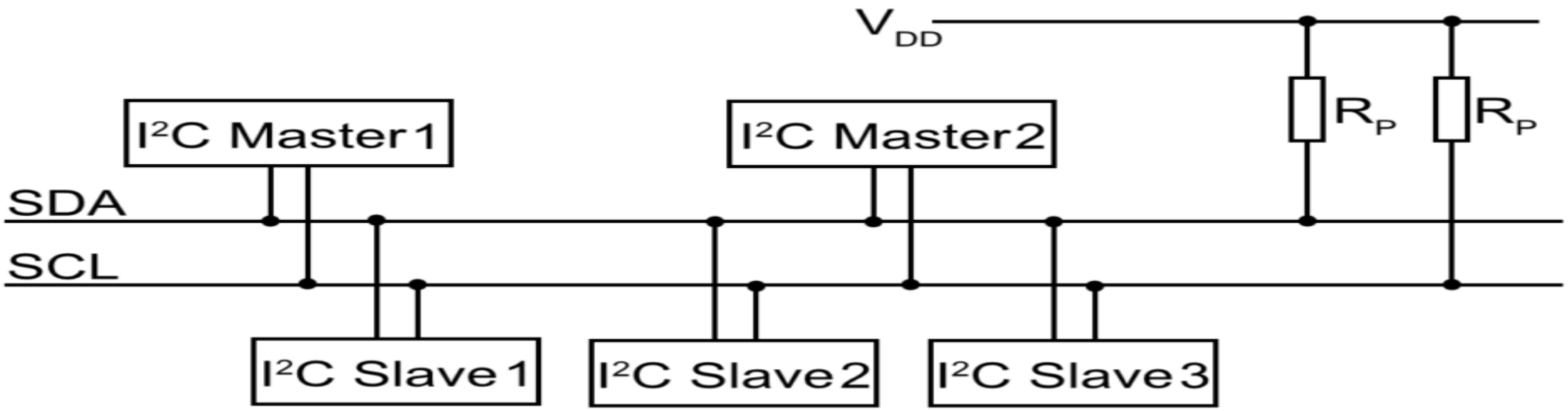
I2C features



I2C Bus Interface



I2C Bus Interface



Serial Data Line (SDA)

The **Serial Data Line (SDA)** is the data line. All the data transfer among the devices takes place through this line.

Serial Clock Line (SCL)

The **Serial Clock Line (SCL)** is the serial clock . I2C is a synchronous protocol, and hence, SCL is used to synchronize all the devices and the data transfer together. The active master is the responsible for driving this line.

Open-Drain Lines

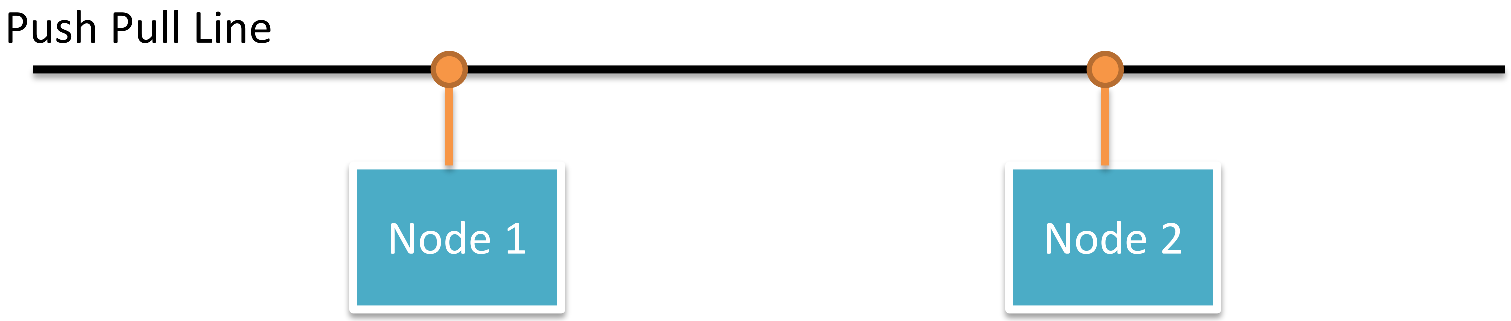
- ❑ **SDA** and **SCL** are *open-drain* (also called *open-collector*) lines.

- ❑ Normal IO Line (Called Push Pull Line) can be derived to GND by writing digital 0 (Pull) or can be derived to VCC by writing 1 (Push)

- ❑ Open Drain Line can be derived only to GND by writing digital 0. Writing digital 1 on open drain line makes it floating.

- ❑ Open Drain lines are used to implement open drain bus. Allowing many devices to be connected on one line, see the following example.

Open-Drain Lines

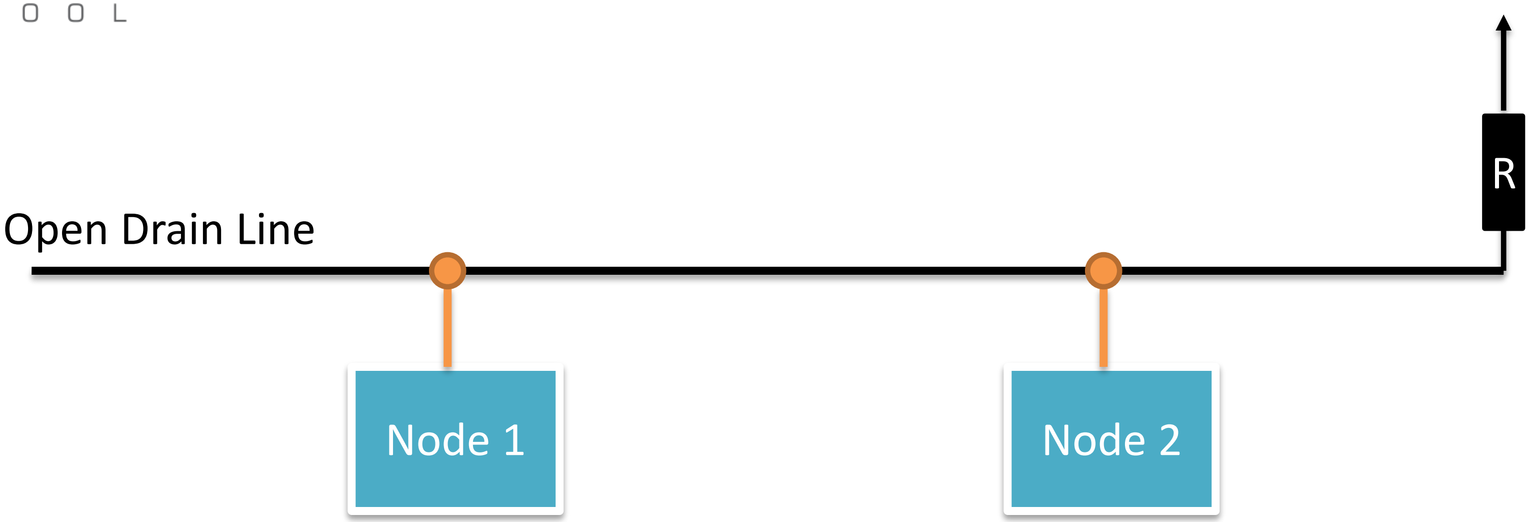


Looking to that bus and considering that Node 1, 2 and 3 can write on it. If the bus is Push Pull bus, then it can be derived to GND or to VCC, what would happen if Node 1 derived the bus to GND and Node 2 derived the bus to VCC ?!

The same line would have a connection to GND and a connection to VCC at the same time which makes a short circuit on the power source ! The network would be damaged

Node 1	Node 2	Result on Line
0	0	GND
1	1	VCC
0	1	Short Circuit
1	0	

Open-Drain Lines

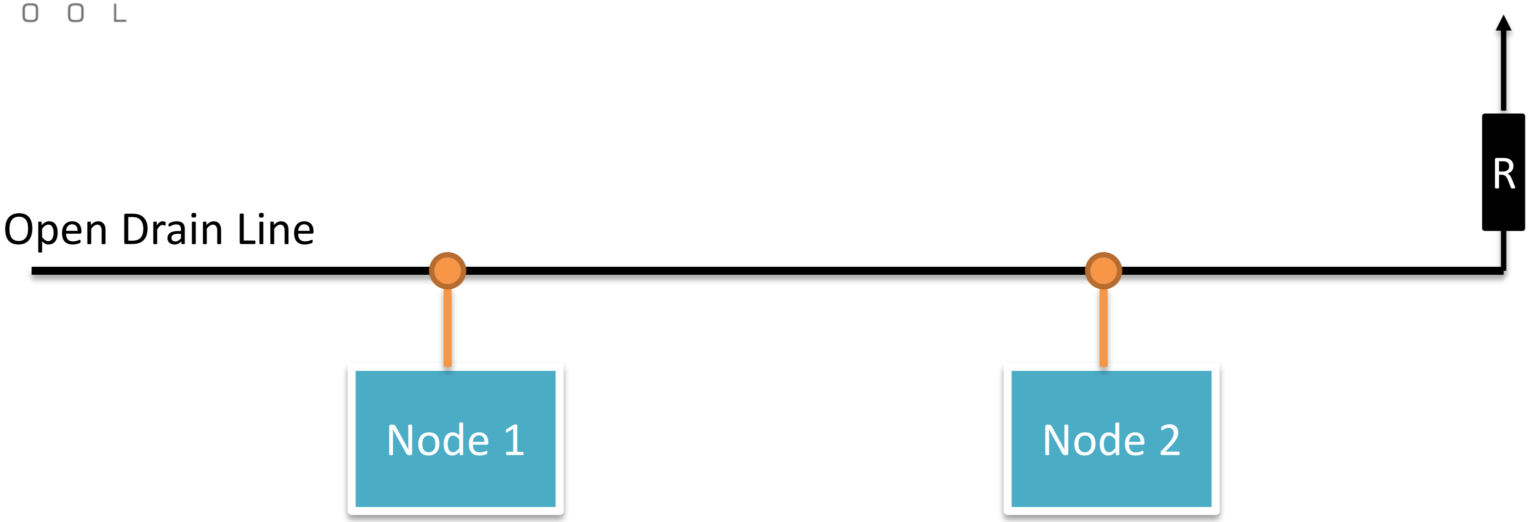


If the bus is open drain line, then any node that writes 0 the bus would be derived to GND, if all nodes writes 1 then the bus would be floating, that's why we use a pull up resistor to connect the line to VCC which will be effective only if all nodes writes 1.

It looks like AND gate, any nodes writes 0, the bus derived to GND, if all nodes write 1, the bus derived to Vcc by the pull up resistor.

Node 1	Node 2	Result on Line
0	0	GND
1	1	VCC
0	1	GND
1	0	

Open-Drain Lines



Dominant Bit

A bit that appears on the bus if any node writes it. In our system the 0 is the dominant

Recessive Bit

A bit that appears on the bus if all nodes write it. In our system the 1 is the recessive

Master and Slave

Master

❿Controls the SCL and generate the clock.

❿Starts and stops data transfer.

❿Controls addressing of other devices.

❿Determines data transfer direction.

Slave

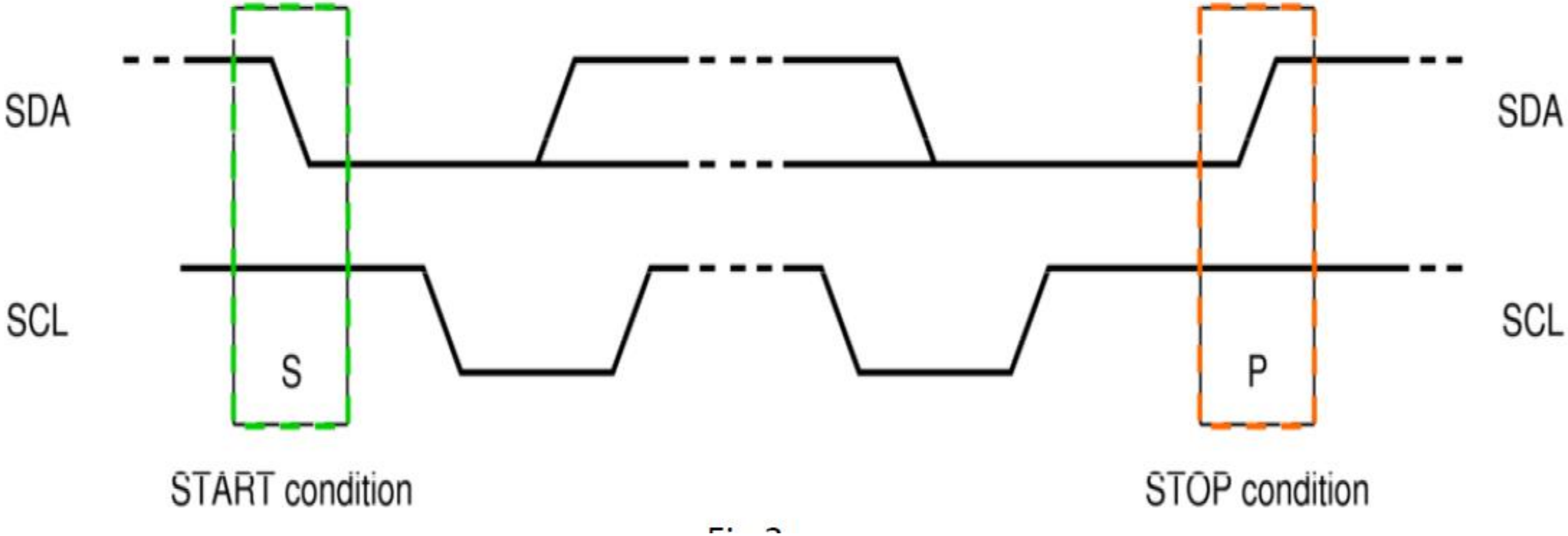
❿Device address by master.

❿Timing is controlled by the clock line.

Start and Stop Conditions

❑ **Start condition :** A high to low transition on **SDA** when **SCL** is high

❑ **Stop condition :** A low to high transition on **SDA** when **SCL** is high

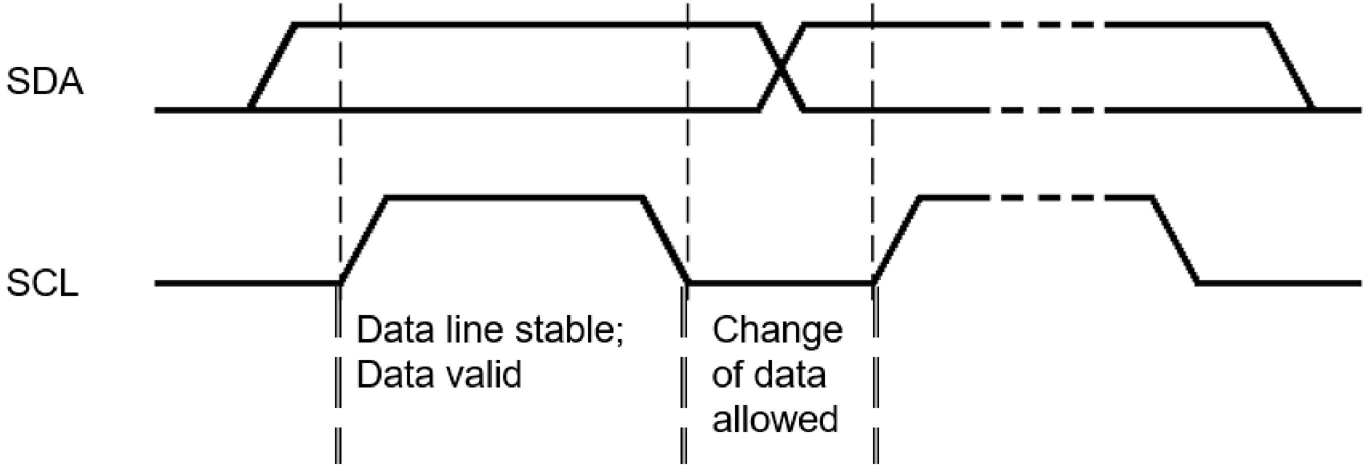


Data Transfer

- ✓ Every Byte put on SDA must be 8 bit long
- ✓ Each Byte followed by Acknowledge bit by the receiver
- ✓ Transfer- MSB to LSB

- ✓ **When SCL is low- Data can be transfer**

  “to avoid the start and stop conditions”

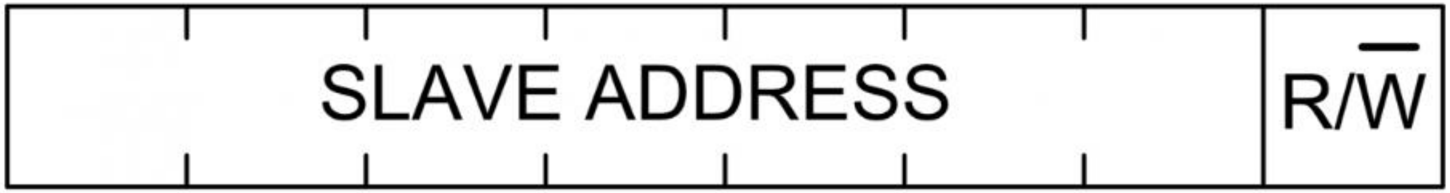


I2C Addressing

- Each node has a unique 7 (or 10) bit address
- Peripherals often have fixed and programmable address portions

✓ The first byte (immediately after the START condition) contains the I2C slave address.

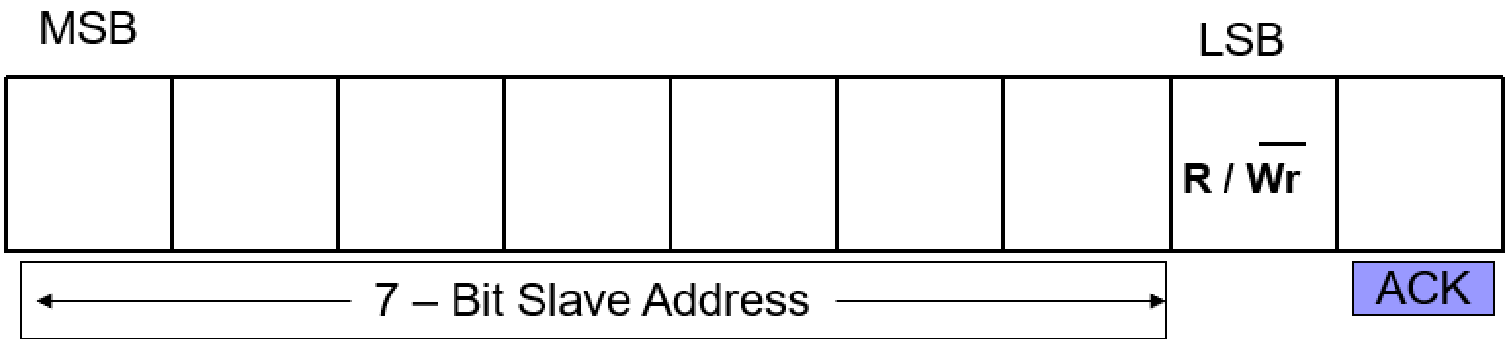
✓ The last bit of the I2C address is a data direction bit).



Addresses starting with 0000 or 1111 have special functions:-

0000000 Is a General Call Address
0000001 Is a Null (CBUS) Address
1111XXX Address Extension
1111111 Address Extension – Next Bytes are the Actual Address

First Byte in Data Transfer



MSB

LSB

R / $\overline{\text{Wr}}$

ACK

7 – Bit Slave Address

R/Wr

0 – Slave written to by Master

1 – Slave read by Master

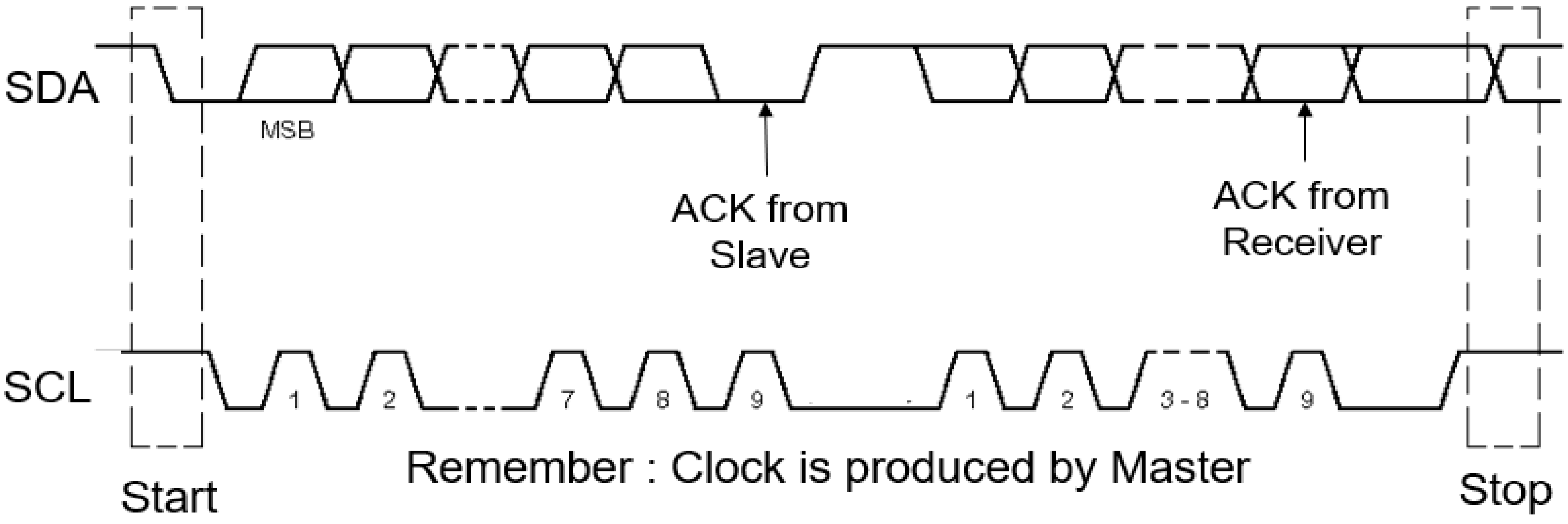
ACK – Generated by the slave whose address has been output.

Acknowledgements

- **From Slave to Master Transmitter:**
  - ➢ After address received correctly
  - ➢ After data byte received correctly
- **From Slave to Master Receiver:**
  - ➢ Never (Master Receiver generates ACK)
- **From Master Transmitter to Slave:**
  - ➢ Never (Slave generates ACK)
- **From Master Receiver to Slave:**
  - ➢ After data byte received correctly

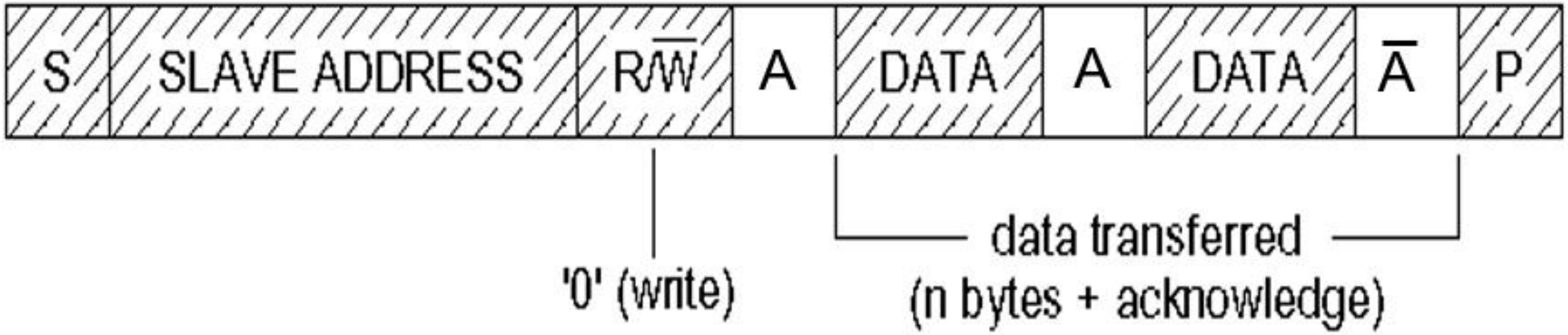
full Data frame

1. **Start Condition**
2. **Slave address + R/W**
   Slave acknowledges with ACK
3. **All data bytes**
   Each followed by ACK
4. **Stop Condition**



Remember : Clock is produced by Master

Master writing to a Slave



S	SLAVE ADDRESS	R/$\overline{W}$	A	DATA	A	DATA	$\overline{A}$	P

'0' (write)

data transferred
(n bytes + acknowledge)

▨ from master to slave

□ from slave to master

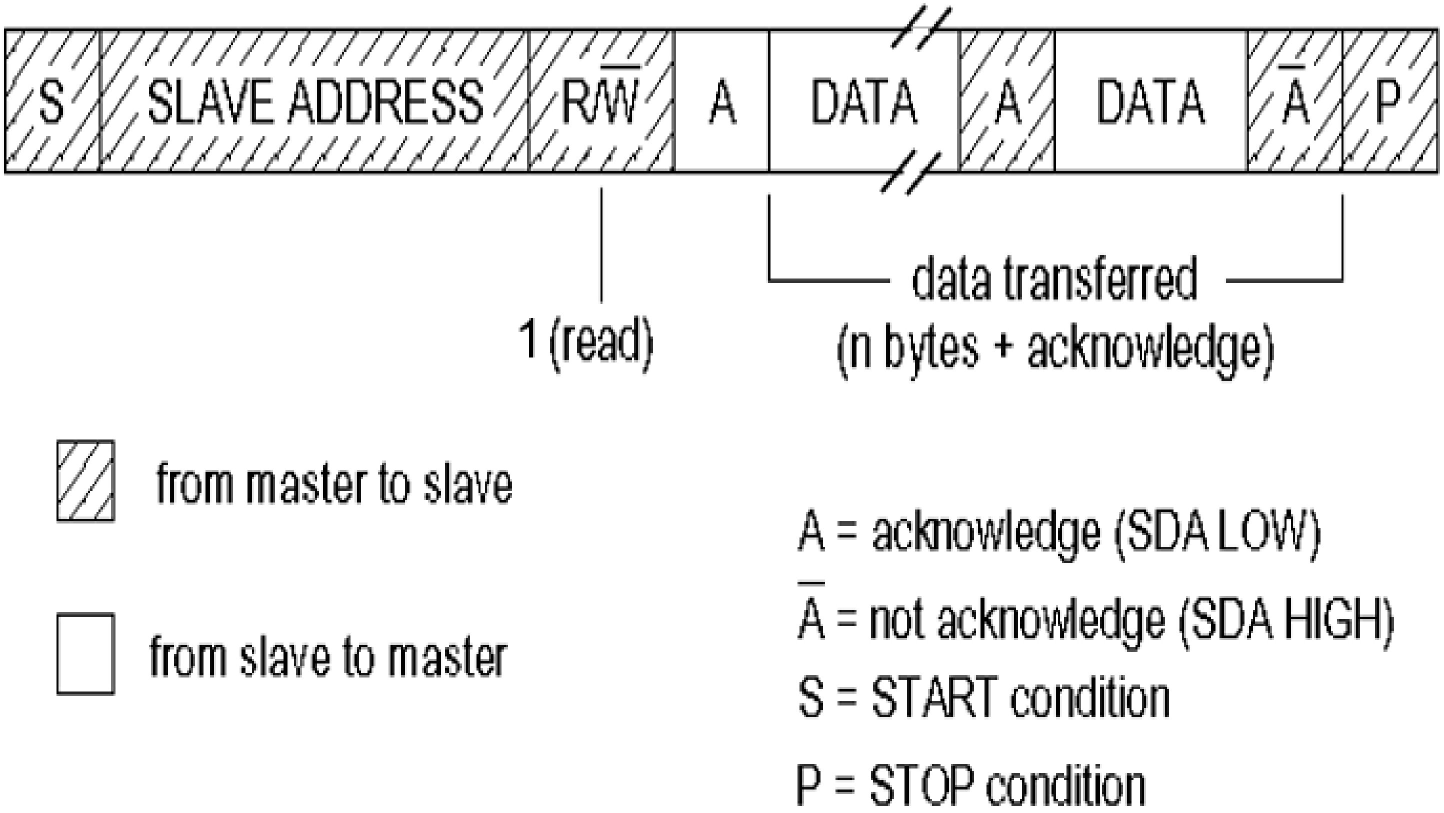
A = acknowledge (SDA LOW)

$\overline{A}$ = not acknowledge (SDA HIGH)

S = START condition

P = STOP condition

Master reading from a Slave



S | SLAVE ADDRESS | R/$\overline{W}$ | A | DATA | A | DATA | $\overline{A}$ | P

1 (read)

data transferred (n bytes + acknowledge)

▨ from master to slave

□ from slave to master

A = acknowledge (SDA LOW)

$\overline{A}$ = not acknowledge (SDA HIGH)

S = START condition

P = STOP condition

Multi Master Arbitration

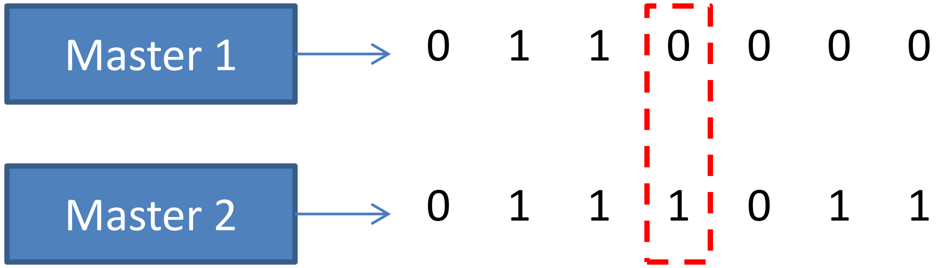
If two or more masters are initiating a frame at the same time, they goes into arbitration and only one of them would win. In multi master bus arbitration, all masters need to keep monitoring the bus for activity. They can send data only when the bus idle or following the stop condition.

If two masters are initiating a frame at the same time, they would send a start condition together. Then they would drive the SCL line with the clock and send data when the clock is low and listen to the data when the clock is high. If the master find that the data sent when the clock is low is not the same as the data sent received when the clock is high, then it understands that a higher priority message is being sent by another master and converts itself to a slave immediately till the stop condition.

This situation would happen if and only if the master is sending 1 and another master is sending 0 at the same time, because the 0 is dominant, then it would appear on the bus !

Multi Master Arbitration

The arbitration is done first on the slave address, think of the following scenario:



Master 1 → 0 1 1 0 0 0 0

Master 2 → 0 1 1 1 0 1 1

Master 1 is sending data to slave address (0110 000) and Master 2 is sending data to slave address (0111 011), remember that the MSB is sent first.

At the first clock, Master 1 sends 0 and Master 2 Sends 0, then when they listen on the high level of the same clock they will found a 0 bit, the same for the second and third clock. But at the forth clock (Bit Number 3) Master 1 sends 0 but Master 2 Sends 1, because 0 is dominant it would appear on the bus. At the high level of this clock cycle Master 1 find 0 as it expects, while Master 2 finds 0 although it sent 1 !
At this stage Master 1 wins the arbitration and Master 2 become a slave.

Multi Master Arbitration

Question, which master would win the arbitration assuming the following scenario:

Master 1 Sending data to Salve 0111   111
Master 2 sending data to Salve 1000   000

Multi Master Arbitration

Question, which master would win the arbitration assuming the following scenario:

Master 1 Sending data to Salve 0111 111
Master 2 sending data to Salve 1000 000

Answer, Master 1 wins the arbitration, because the MSB is sent first, then Master 1 wins the arbitration at first clock !

Summary:
It doesn’t matter how many 0’s in the slave address, it matters only when the first 0 comes !

The rule says:
The Highest priority slave is the slave has the lowest address value

Multi Master Arbitration

What if the 2 masters are communicating to the same slave … ?

Then the arbitration done on the control bit, the master who writes to the slave (Send 0) wins over the master who reads from the slave (Send 1).

What if the 2 masters are writing to the same slave … ?

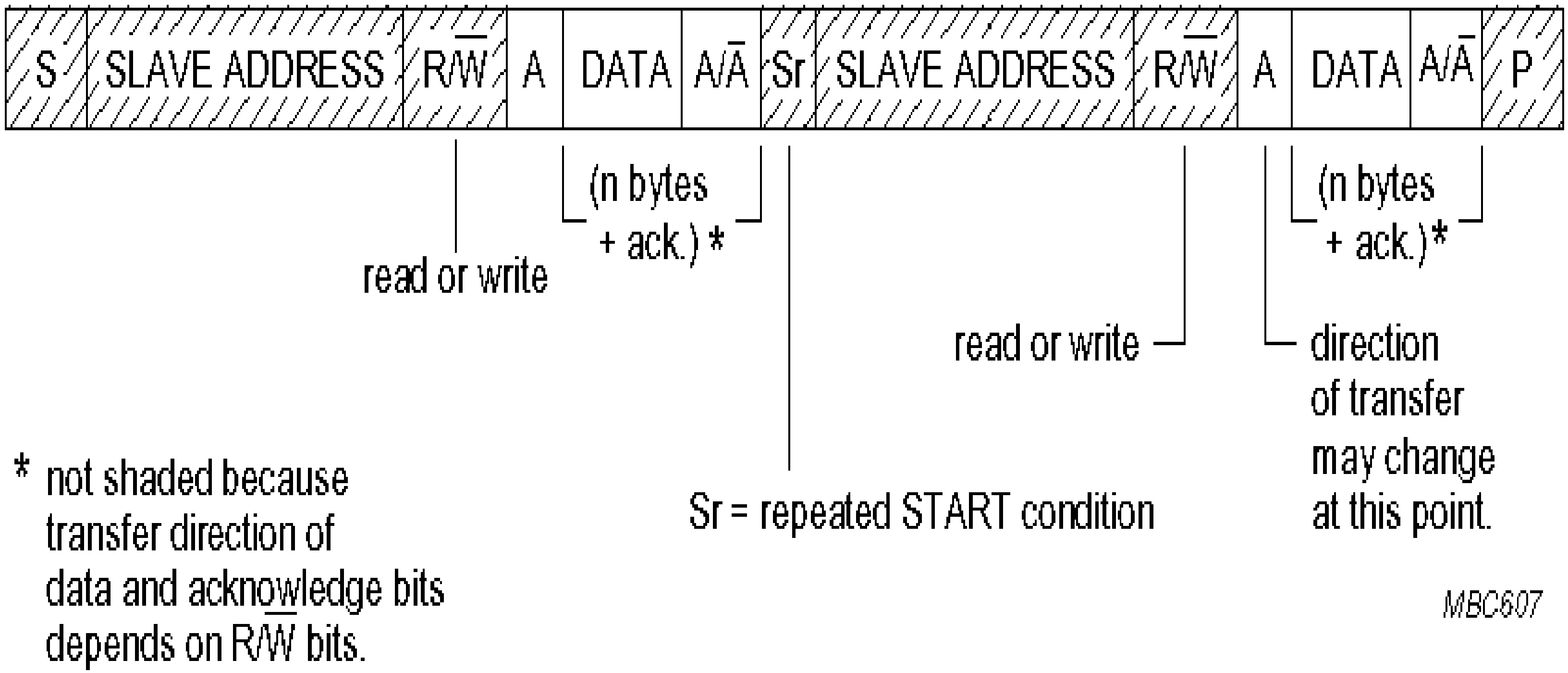
Then the arbitration done on data byte, the master who send low value data wins the arbitration.

What if the 2 masters are writing to the same slave with the same data … ?

Then no problem, all masters win 😁

Combined Format

- A **repeated start** avoids releasing the bus and therefore prevents another master from taking over the bus



S	SLAVE ADDRESS	R/$\overline{W}$	A	DATA	A/$\overline{A}$	Sr	SLAVE ADDRESS	R/$\overline{W}$	A	DATA	A/$\overline{A}$	P

read or write
(n bytes + ack.) *
read or write
(n bytes + ack.)*
direction of transfer may change at this point.
Sr = repeated START condition
MBC607

* not shaded because transfer direction of data and acknowledge bits depends on R/$\overline{W}$ bits.

Clock Stretching

when the receiver needs more time to process a write or read request before it gives an acknowledge, it can pull SCL low to keep it from going high a bit longer. i.e. the slave can write a low on the SCL line at clock cycle number 9 for every byte. This would prevent the master from writing 0 high on the SCL until the slave release the SCL.

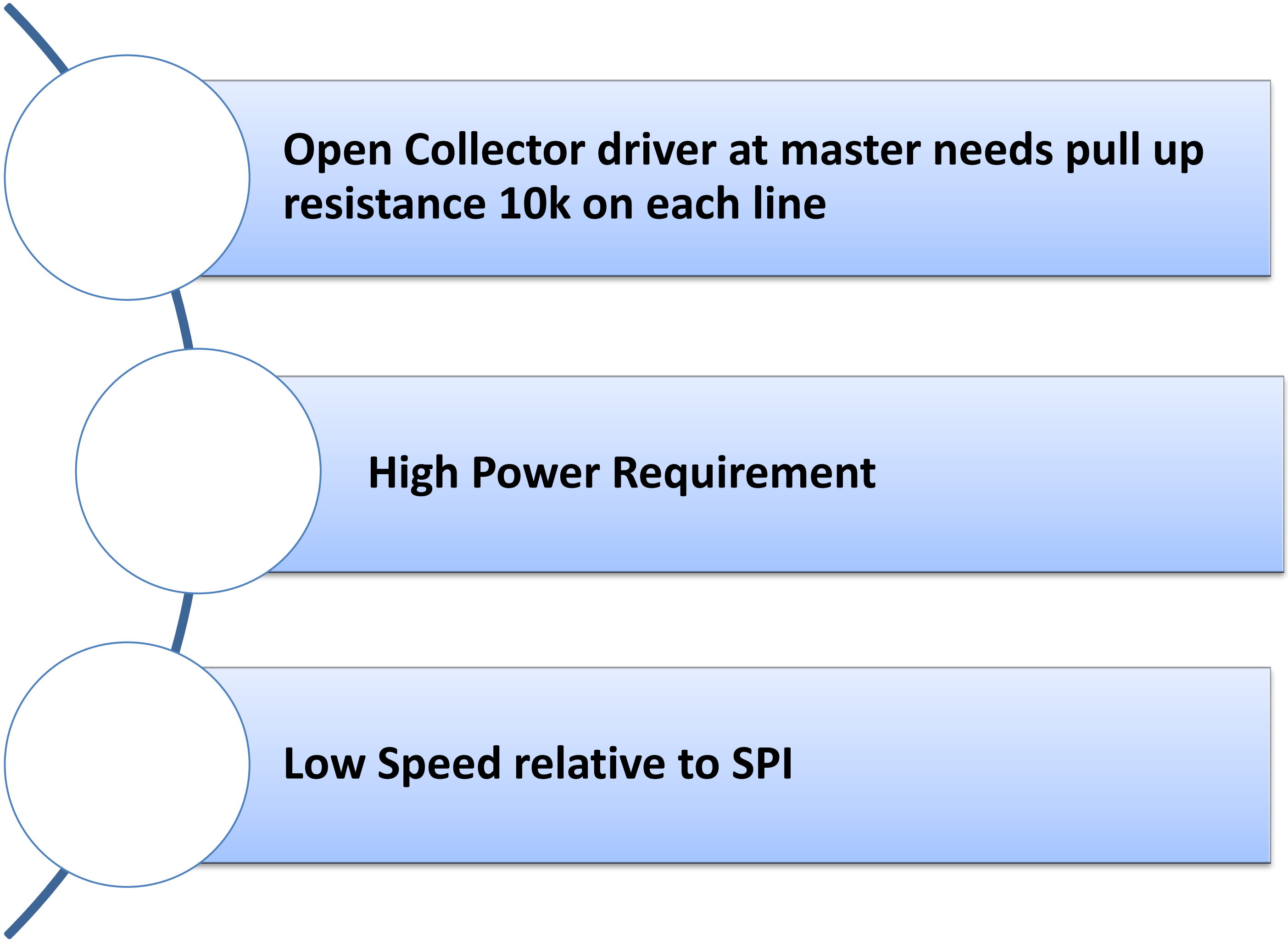
This is used to allow the slave to stretch the ninth clock to do some processing before it gives acknowledge to the master !

Slave can not never write on the SCL line except for this clock cycle.

I2C Advantages

- Only two signal lines requires
- Flexible data transmission rates
- Each device on the bus is independently addressable
- Devices have a simple Master/Slave relationship
- Capable of handling multiple master communications by providing arbitration

I2C Disadvantages



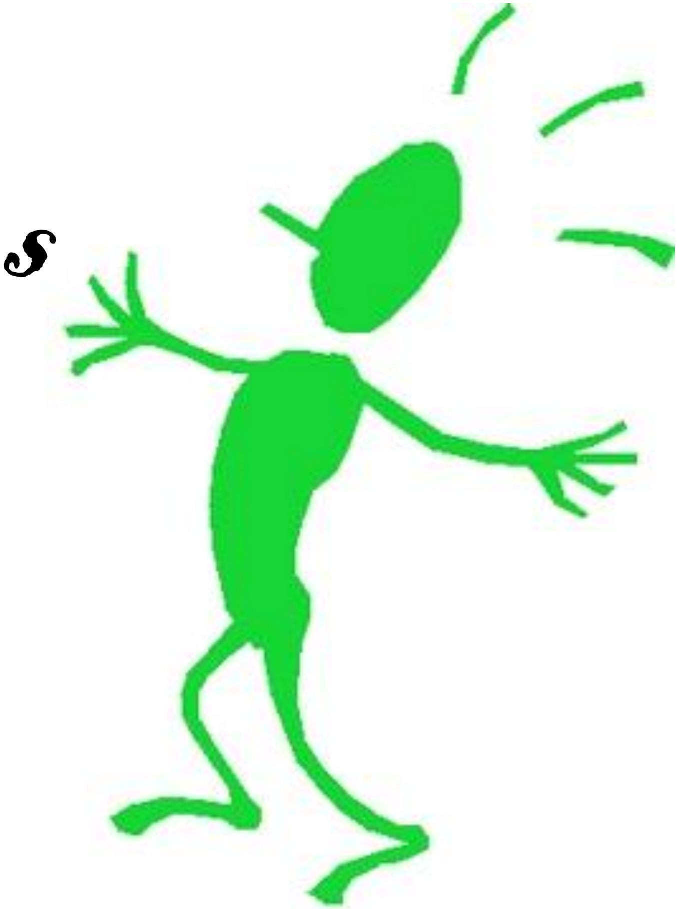
Open Collector driver at master needs pull up resistance 10k on each line

High Power Requirement

Low Speed relative to SPI

The End ...

Any questions ... ?



IMT SCHOOL

www.imtschool.com

ww.facebook.com/imaketechnologyschool/

This material is developed by IMTSchool for educational use only
All copyrights are reserved