Hand written notes on I2C

I2C

In I2C the data is transfer between two or
more than two ICs or Controllers.
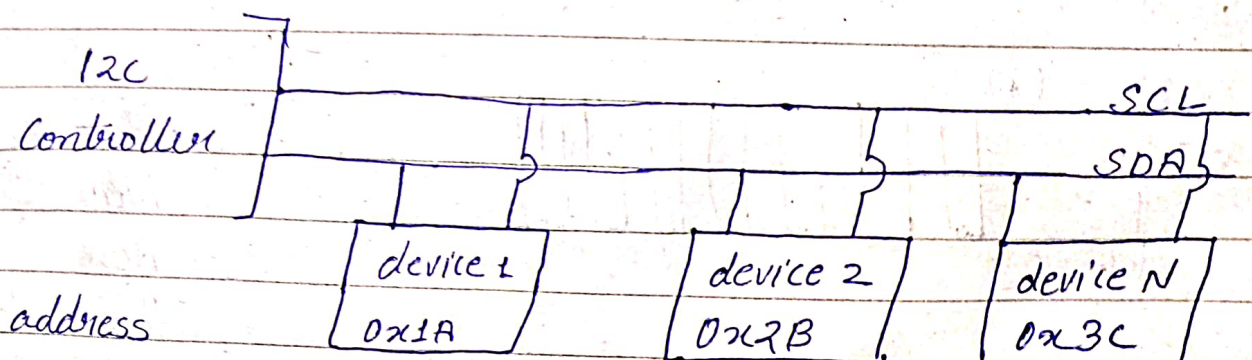In which one is master and others are slaves
means it is master - slave architecture

it also supports multi-master means multiple
masters can share the bus, but only one can
transmit data at a time to avoid collisions.

There are only two wire use for Communication
1. SDA : Serial data line
2. SCL : Serial clock line
                    All devices on the bus
are connected to these two lines, as show in the
figure below

SCL: Synchronise all devices on the bus to
        one clock while
SDA: line is bidirectional so data can travel
    either direction (from controller to target or
    target to Controller)

Each I2C target device has a unique address
that distinguishes it from the other I2C
target devices on the same bus.

unique addressing

Each slave device is assigned a unique 7-bit
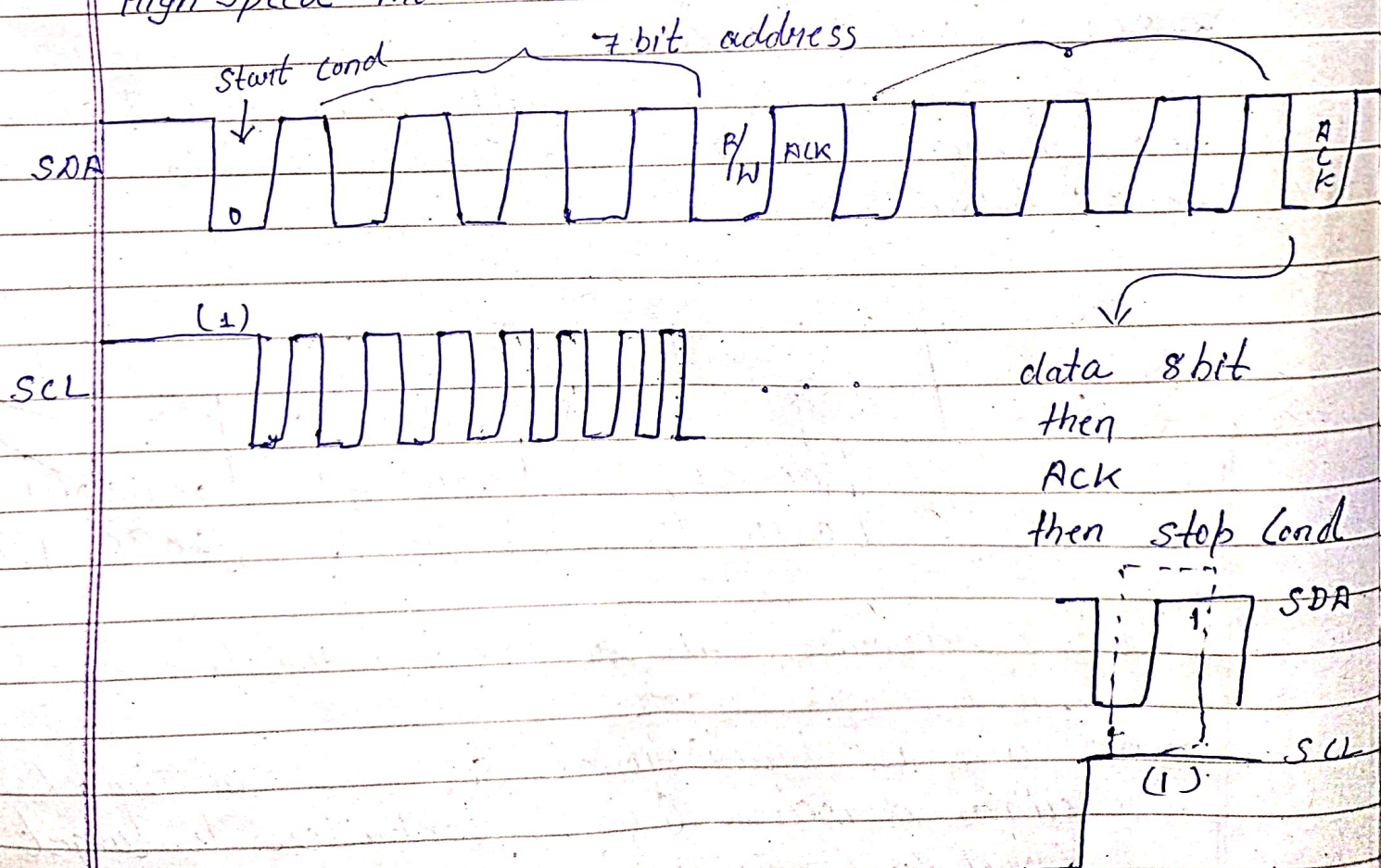(128 address) or 10-bit for identification

Standard clock speeds
Standard mode : 100 KHz
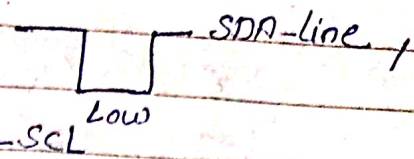Fast mode : 400 KHz
Fast mode Plus : 1 MHz
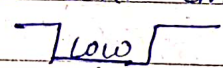High speed mode : 3.4 MHz

8 bit Register
address

Start cond
7 bit address
B/W ACK
SDA
0

(1)

SCL

data 8 bit
then
ACK
then stop cond

SDA

SCL
(1)

How I2C works?

Start Condition :

1 * The Master pulls the SDA line low while keeping the SCL line High ⎍ High ⎍ SCL Low ⌐ SDA-line

This Signals all devices on the bus that a transmission is starting.

2 * The Master sends the 7 bit or 10-bit address of the slave device it wants to communicate with, followed by a read / write R/W bit

0 : write , 1 : Read
The addressed slave responds with an ACK after it, each byte by pulling SDA low ⌐ Low ⌐

3 * Communication Can be master to slave (write) or slave to master (Read)

Data is sent byte by byte (8bits), with the receiver sending an Ack after each byte.

4 * Stop Condition :
the master releases the SDA-line (pull it high) ⌐ while SCL is high ⌐ ⌐ ,
this Signaling end of the Communication.

5 * Bus Arbitration (multi-Master systems)
when multiple masters attempts to use the bus Simultaneously

6 * Master monitors the SDA line while transmitting

if master detects a mismatch between what it sent and what is on the bus, it stop transmitting, letting the other master take control.

Common challenges in I2C

→ Bus line stuck low :
A faulty device or wiring issue can pull the SDA or SCL line low, disrupt the entire bus, stop working

→ Address conflicts :
Multiple devices with same address on the bus require hardware or software work arounds, like address remapping or using an I2C multiplexer.

→ clock stretching :
Some slave may holds the SCL line to low to delay the master, allowing more time for processing

→ Signal Integrity :
Noise or improper pull-up resistors can cause data corruption