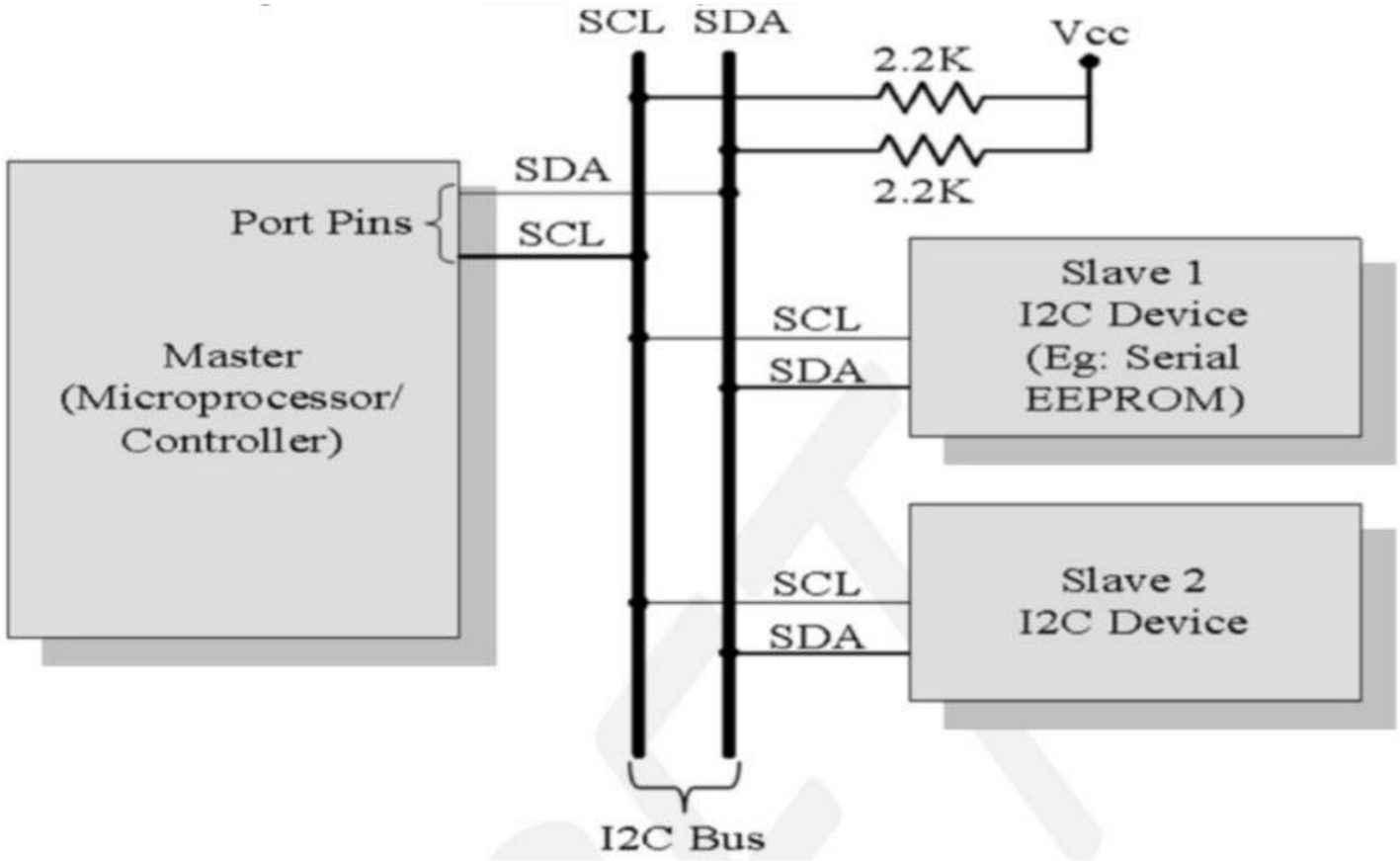
UNIT 4-Embedded System

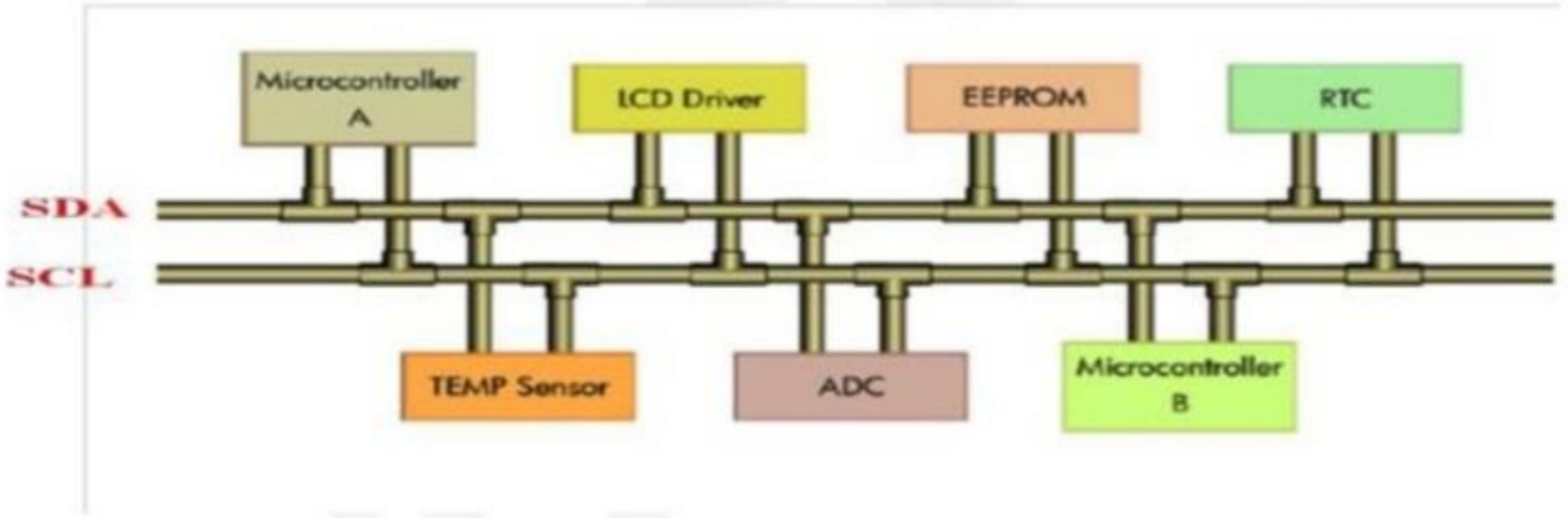
I2C(Inter Integrated Circuit)

I2C (Inter Integrated Circuit) Bus:

- Inter Integrated Circuit Bus (I2C - Pronounced „I square C‟) is a synchronous bi-directional half duplex (one-directional communication at a given point of time) two wire serial interface bus.
- The concept of I2C bus was developed by „Philips Semiconductors‟ in the early 1980‟s.
- The original intention of I2C was to provide an easy way of connection between a microprocessor/microcontroller system and the peripheral chips in Television sets
- The I2C bus is comprised of two bus lines, namely; Serial Clock – SCL and Serial Data – SDA.


SCL
SDA
Vcc
2.2K
2.2K
SDA
Port Pins
SCL
Master
(Microprocessor/
Controller)
SCL
SDA
Slave 1
I2C Device
(Eg: Serial
EEPROM)
SCL
SDA
Slave 2
I2C Device
I2C Bus

I2C Bus Interfacing

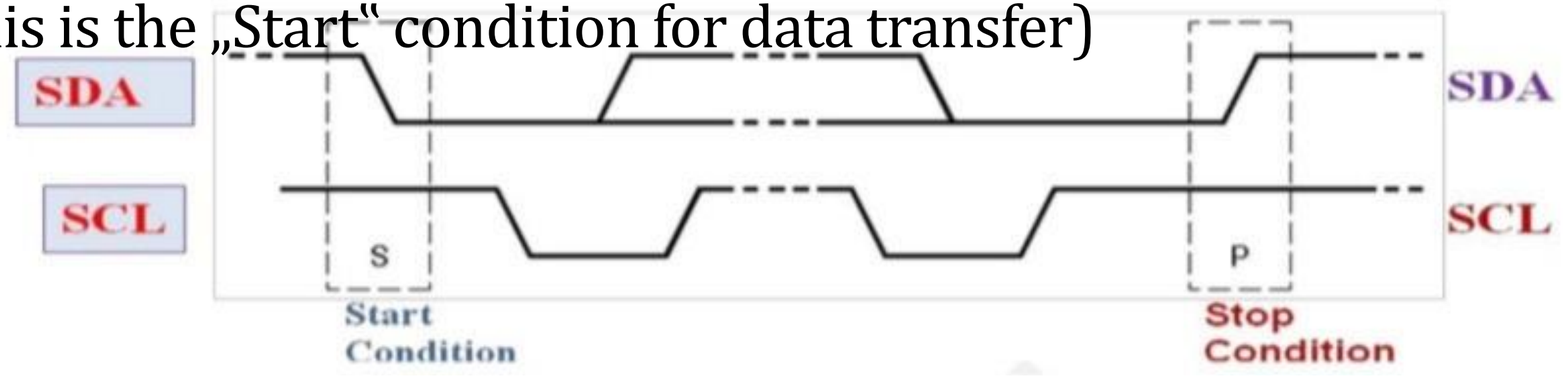


- SCL line is responsible for generating synchronization clock pulses and SDA is responsible for transmitting the serial data across devices.I2C bus is a shared bus system to which many number of I2C devices can be connected. Devices connected to the I2C bus can act as either „Master" device or „Slave" device.
- The „Master" device is responsible for controlling the communication by
- initiating/terminating data transfer, sending data and generating necessary synchronization clock
- pulses.
- Slave devices wait for the commands from the master and respond upon receiving the commands.Master and „Slave" devices can act as either transmitter or receiver. Regardless
- whether a master is acting as transmitter or receiver, the synchronization clock signal is generated by the „Master" device only.I2C supports multi masters on the same bus.

The sequence of operation for communicating with an I2C slave device is:

1. Master device pulls the clock line (SCL) of the bus to „HIGH“

2. Master device pulls the data line (SDA) „LOW“, when the SCL line is at logic „HIGH“ (This is the „Start“ condition for data transfer)

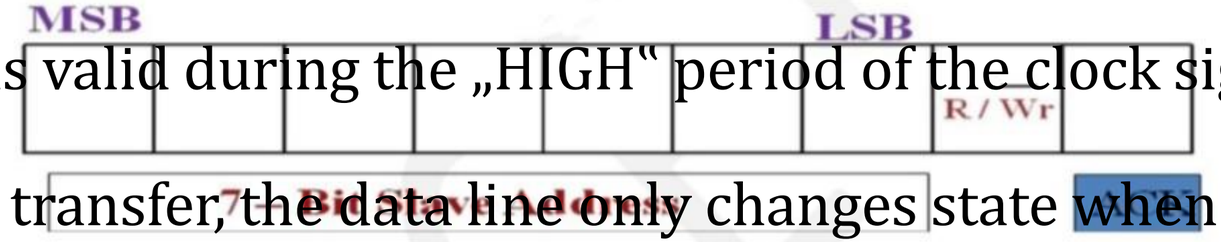


3. Master sends the address (7 bit or 10 bit wide) of the „Slave" device to which it wants to communicate, over the SDA line.

4. Clock pulses are generated at the SCL line for synchronizing the bit reception by the slave device.

5. The MSB of the data is always transmitted first.

6. The data in the bus is valid during the „HIGH" period of the clock signal



7. In normal data transfer, the data line only changes state when the clock is low

R/Wr

0 – Master writes to the slave

1 – Master read from slave

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

8. Master waits for the acknowledgement bit from the slave device whose address is sent on

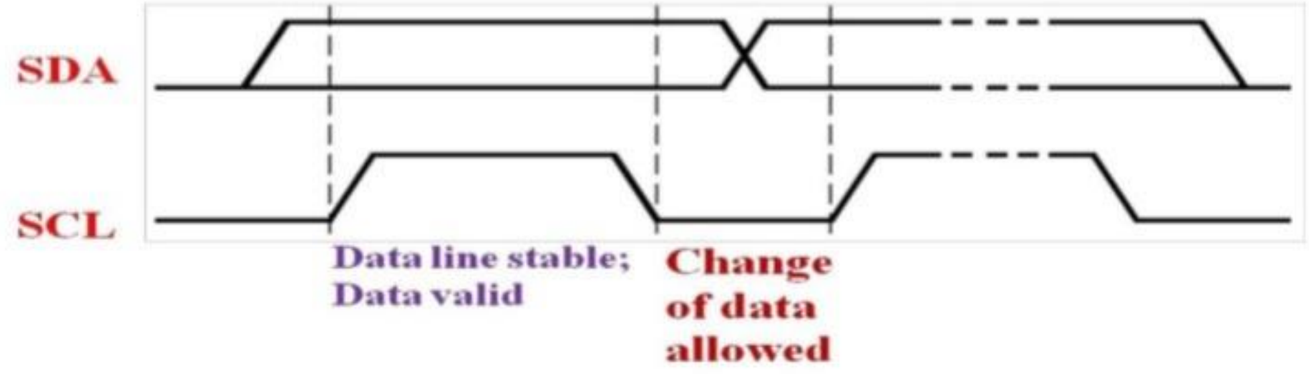
the bus along with the Read/Write operation command.
9. Slave devices connected to the bus compares the address received with the address

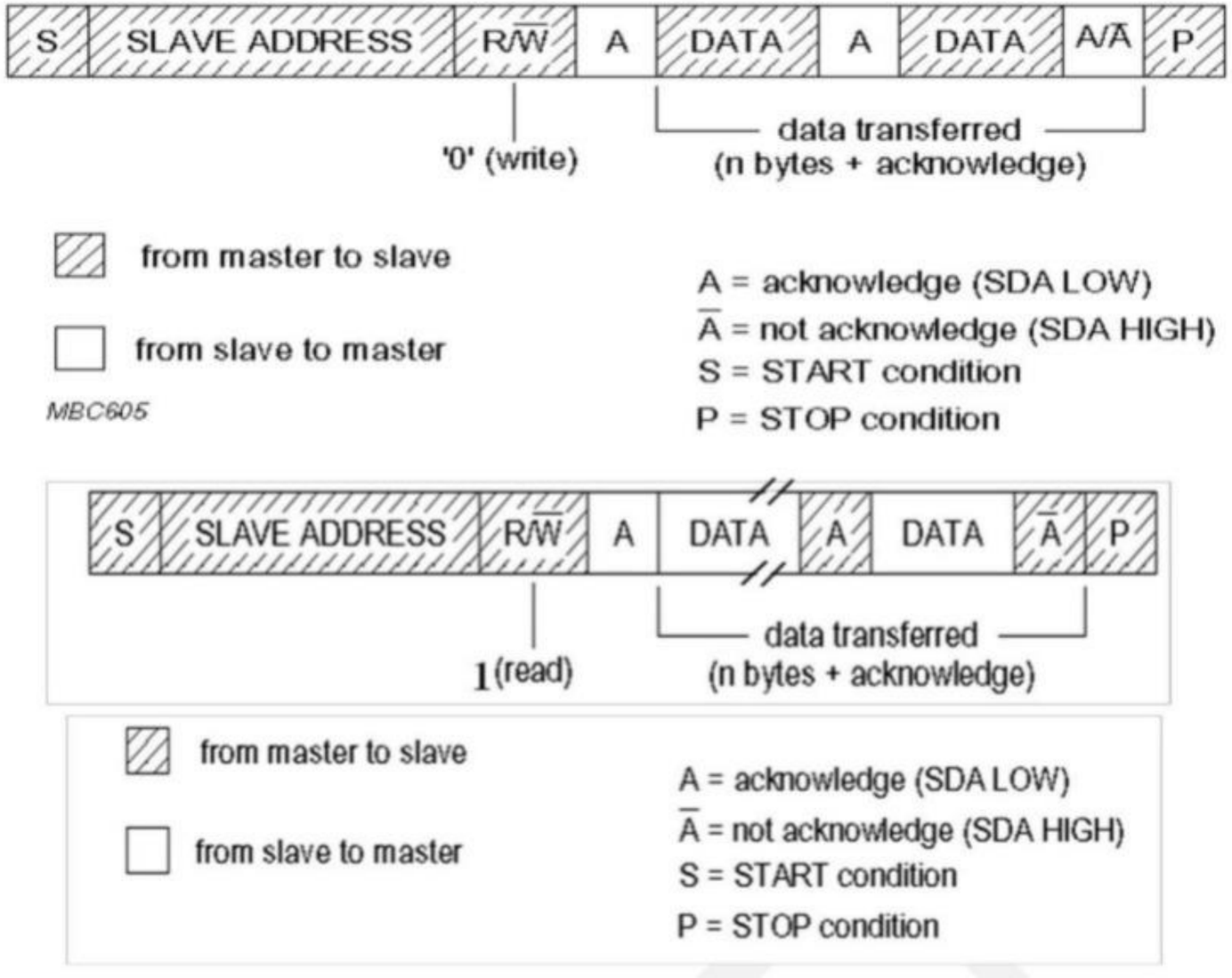
assigned to them

10. The Slave device with the address requested by the master device responds by sending an acknowledge bit (Bit value =1) over the SDA line
11. Upon receiving the acknowledge bit, master sends the 8bit data to the slave device over SDA line, if the requested operation is „Write to device".
12. If the requested operation is „Read from device", the slave device sends data to the master over the SDA line.

13. Master waits for the acknowledgement bit from the device upon byte transfer complete for a write operation and sends an acknowledge bit to the slave device for a read operation

14. Master terminates the transfer by pulling the SDA line „HIGH‟ when the clock line SCL is at logic „HIGH‟ (Indicating the „STOP‟ condition).




S
SLAVE ADDRESS
R/W
A
DATA
A
DATA
A/Ā
P
'0' (write)
data transferred
(n bytes + acknowledge)
from master to slave
from slave to master
MBC605
A = acknowledge (SDA LOW)
Ā = not acknowledge (SDA HIGH)
S = START condition
P = STOP condition
S
SLAVE ADDRESS
R/W
A
DATA
A
DATA
Ā
P
1 (read)
data transferred
(n bytes + acknowledge)
from master to slave
from slave to master
A = acknowledge (SDA LOW)
Ā = not acknowledge (SDA HIGH)
S = START condition
P = STOP condition

Thank you