

(Inter-IC bus)

By
Tejaswini Gadicherla
(tgadiche@uncc.edu)

Topics Discussed....

- n What is I²C ???
- n Why has it endured?
- n Bus transfer terminology
- n How does it work?

Introduction to I²C

- n I²C is a control bus providing communications link between integrated circuits in a system.
- n Developed by Philips in the early 1980s.
- n Examples of I²C-compatible devices found in embedded systems include EEPROMs, thermal sensors and real-time clock.

Why I2C has endured?

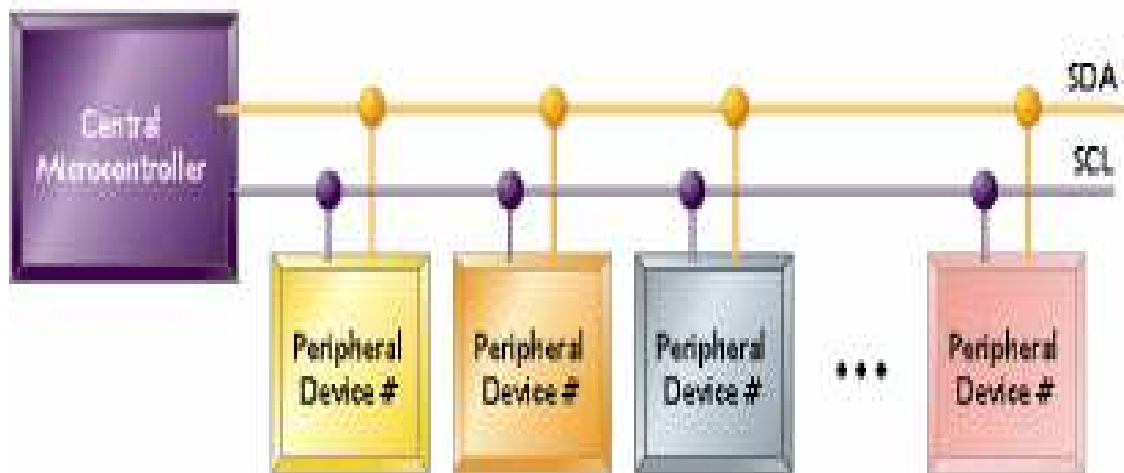
- n Bus has kept pace with performance and today offers 3 levels of data transfer rate
 - i 100kbps in standard mode
 - i 400kbps in Fast mode
 - i 3.4 Mbps in high-speed mode

Why I2C has endured?

- n Reliable performance using software-controlled collision detection and arbitration.
- n Ease of use. 2 lines connect all ICs in a system.
- n Software controlled addressing scheme eliminating need for address-decoding hardware.

What does I2C consist of?

Figure 1: I²C has two lines in total



I2C is a 2 wire serial bus as shown above. The 2 signals are


SDA à Serial Data

SCL à Serial Clock

Together these signals make it possible to support serial transmission.

I2C BUS

- n The device that initiates the transaction on the I2C bus is termed the master. The master normally controls the clock signal.
- n A device being addressed by the master is called the slave.
- n There needs to be at least one master(a microcontroller or a DSP) on the bus, but there can be more than one master. All the masters on a bus have equal priority.

- 
- n Every device on the I2C bus has a unique 7 bit (or 10 bit) I2C address.
 - n Typically the 4 most significant bits are fixed and assigned to specific categories of devices. Ex: 1010 is assigned to serial EEPROM.
 - n The lower 3 bits are programmable allowing 8 devices of one kind to be present on a single I2C bus.

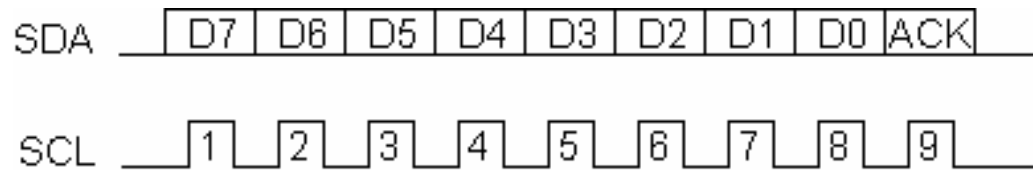
Terminology for bus transfer

SCL and SDA are bi-directional !

- ∅ F(free) – Bus is idle or free. Both SDA and SCL are in a high state
- ∅ S(Start) or R(Restart) – SDA changes from high to low with the SCL line remaining high. All data transfers begin with S(Start) condition.
- ∅ C(Change) – SCL line is low. Data bit to be transferred is applied to the SDA line.
- ∅ D(Data) – high or low bit of information on SDA line is valid during the high level of the SCL line.
- ∅ P(Stop) – SDA line changes from low to high with SCL line remaining high. All data transfers end with P(Stop) condition.

Data transfer

- Data is transferred in sequences of 8 bits.
- Bits are placed on SDA line with MSB first.
- SCL line is then pulsed high and then low.
- For every 8 bits transferred, the device receiving the data sends back an acknowledge bit.



Data transfer(contd)

- n If the receiving device sends back a low ACK bit, then it has received the data and is ready to accept another byte.
- n If it sends back a high ACK, then the device is indicating that it cannot accept any further data and the master should terminate the transfer by sending a stop sequence.

I2C device addressing

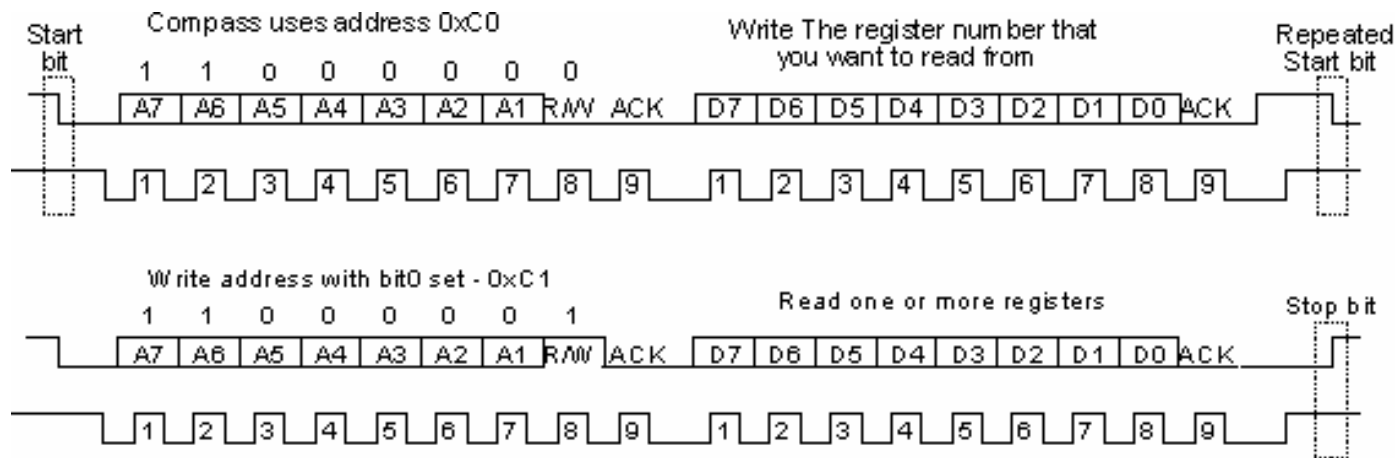
- n All I2C devices are either 7 (or 10) bits. Yet we send out 8 bits when trying to address a device.
- n This extra bit is used to inform the slave whether the master is writing to it or reading from it.
- n The 7 bit address is placed in the upper 7 bit and the R/W bit is placed in the LSB.
- n So if LSB is 1, master wants to read from a slave. Else if LSB is 0, master wants to write to a slave.

Writing to a slave device

- n Master sends a start sequence. This alerts all the slave devices to an impending transaction and they should listen , in case its for them.
- n Next the master sends out the device address with read/write bit low. The slave that matches this address will continue the transaction , while others ignore.
- n Master can now send data byte(s).
- n Master sends the stop sequence.

Reading from a slave

- n Master sends a start sequence.
- n Master sends the device address with read/write bit high.
- n Master reads data from the device.
- n Master sends the stop sequence



References

- n <http://www.semiconductors.philips.com/markets/mms/protocols/i2c/facts/>
- n <http://www.embedded.com/story/OEG20010718S0073>
- n <http://www.mcc-us.com/i2chowto.htm>