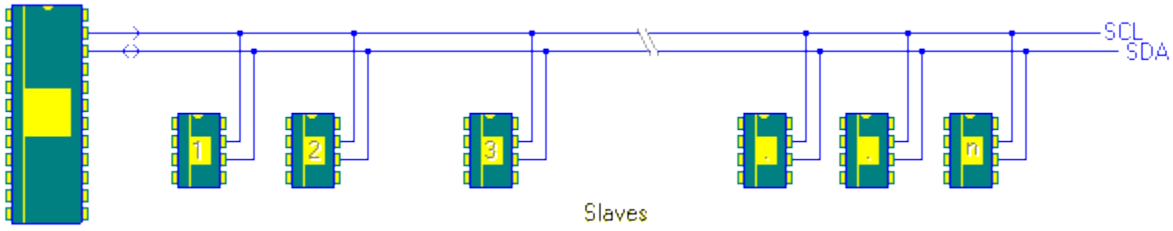
I2C – Inter-IC Communications

Lectures 28, October 26-29, 2012

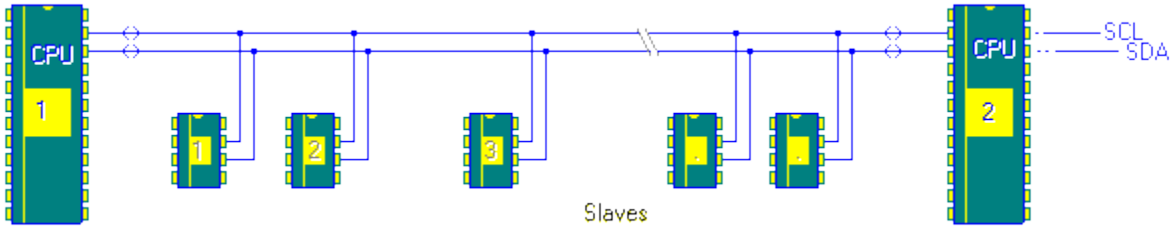
I. Background - The I2C bus was developed in the early 1980's by Philips Semiconductors. Its original purpose was to provide an easy way to connect a CPU to peripheral chips in a TV-set. The original specification supported data rates up to 100 Kbits/s. Conventional hardware now supports up to 400 Kbits/s. The slowest device on the I2C network dictates the maximum data transfer rate.

II. Connection:

a. Single Master configuration



b. Multi Master Configuration



III. Protocol

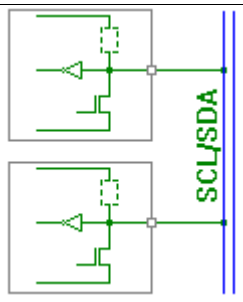
a. Physical Layer

i. Two wire (plus reference) A clock signal (SCL) and the bidirectional data signal (SDA). SDA and SCL buses are usually operated using an open drain (collector) configuration.

ii. Signal levels – as needed by the hardware. Typical levels are 0 to 3.3V / 5.0V

iii. High logic levels are designated as the recessive level, the low logic level is the dominant level. If two devices are simultaneously attempting to control the signal line, the dominate level is always asserted over the recessive level.

iv. Device Termination:

Simultaneous input and output on both SCL and SDA. Recessive bit – processor sets pin for input or open drain output Dominate bit – processor sets pin as output with the output logic level set for 0.	

v. Clock signal is always controlled by the Master Device.

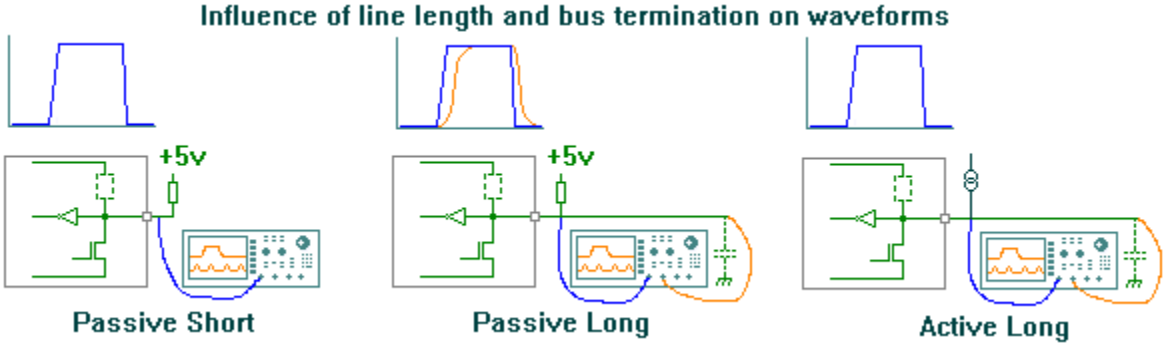
1. Synchronization: All masters generate their own clock on the SCL line to transfer messages on the I2C-bus. Data is only valid during the HIGH period of the clock. A defined clock is therefore needed for the bit-by-bit arbitration procedure to take place.

   Clock synchronization is performed using the wired-AND connection of I2C interfaces to the SCL line. This means that a HIGH to LOW transition on the SCL line will cause the devices concerned to start counting off their LOW period and, once a device clock has gone LOW, it will hold the SCL line in that state until the clock HIGH state is reached. However, the LOW to HIGH transition of this clock may not change the state of the SCL line if another clock is still within its LOW period. The SCL line will therefore be held LOW by the device with the longest LOW period. Devices with shorter LOW periods enter a HIGH wait-state during this time.

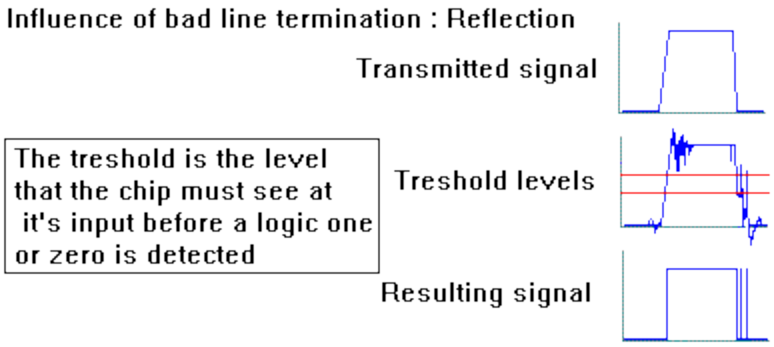
   When all devices concerned have counted off their LOW period, the clock line will be released and go HIGH. There will then be no difference between the device clocks and the state of the SCL line, and all the devices will start counting their HIGH periods. The first device to complete its HIGH period will again pull the SCL line LOW. In this way, a synchronized SCL clock is generated with its LOW period determined by the device with the longest clock LOW period, and its HIGH period determined by the one with the shortest clock HIGH period.

vi. Advantages:
   1. Error detection during transmission of recession bits.
   2. SLAVE device can slow clock down by asserting the SDA line in the dominate state until the SLAVE is ready to continue

vii. Disadvantages:
   1. Long bus lines limit the data speed – Low pass filtering



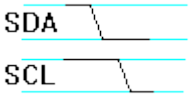
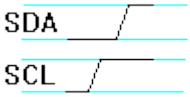
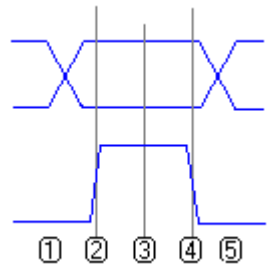
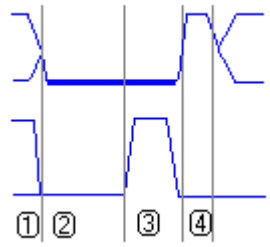
   2. Poor line terminations



b. Data Framing.

Write Data

F	S	Slave Address	$\overline{W}$	A	Data	A	Data	$\overline{A}$	P	F
					< n data bytes >		last data bytes			

Start Signal		The device issuing the Start condition first pulls the SDA (data) line low, and next pulls the SCL (clock) line low.
Stop Signal		The Bus Master first releases the SCL and then the SDA line.
Data Signal		All the master has to do is generate a rising edge on the SCL line (2), read the level on SDA (3) and generate a falling edge on the SCL line (4). The slave will not change the data during the time that SCL is high. (Otherwise a Start or Stop condition might inadvertently be generated.)
Acknowledge Signal		This means that as soon as the master pulls SCL low to complete the transmission of the bit (1), SDA will be pulled low by the slave (2). The master now issues a clock pulse on the SCL line (3). the slave will release the SDA line upon completion of this clock pulse (4). The bus is now available again for the master to continue sending data or to generate a stop condition. (See note)

c. Uses of ACKNOWLEDGE bit:

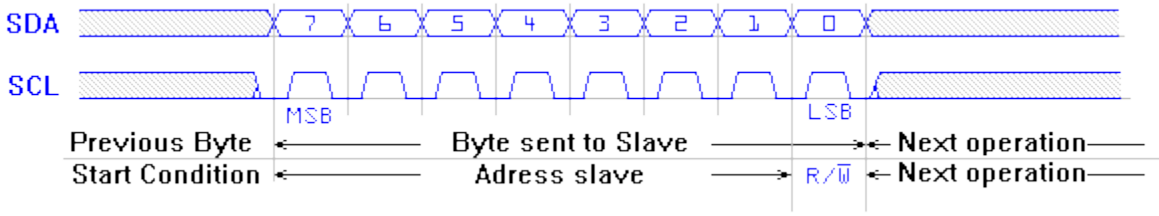
i. The ACKNOWLEDGE always follows 8 bits of data transfer The slave device transmitted must ACKNOWLEDGE address and data sent by the MASTER device. The slave sends an ACK signal to the MASTER device. The MASTER device must assert the ACKNOWLEDGE signal when receiving data from the SLAVE device.

ii. No ACKNOWLEDGE Condition:

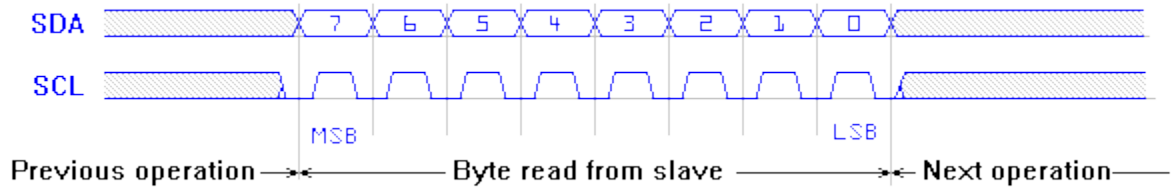
1. NACK – Sent from Master to Slave to indicate the end of a read data sequence. This is NOT an error condition.
2. No ACK – Received by the master to indicate that a Slave device failed to acknowledge the reception of data or address byte. This IS an error condition. Possible reasons for a no ACK condition:
   a. The slave is not there (in case of an address) – possible incorrect device address.
   b. Slave device is not functional.
   c. The slave missed a pulse and got out of sync with the SCL line of the master.
   d. The bus is "stuck". One of the lines could be held low permanently.

d. Data segment

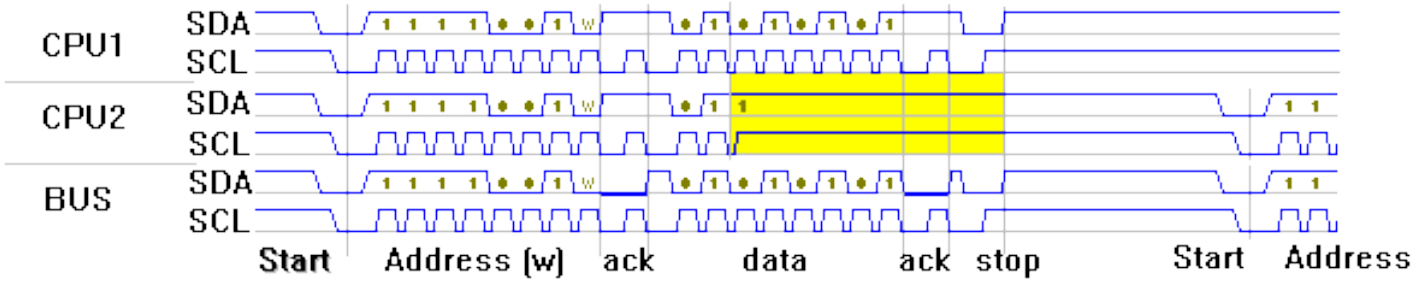
i. Master to Slave transfer



ii. Save to Master transfer



e. Multi Master Bus Arbitration



IV. Advanced operations

a. Extended Addressing – 10 bit identifiers
b. High Speed 3.4 MBit/s
c. Enhanced I2C (FAST Mode) – 400 Kbits/s

V. References:

a. Course Text – Chapter 8
b. http://www.esacademy.com/en/library/technical-articles-and-documents/miscellaneous/i2c-bus/general-introduction/bus-arbitration.html
c. Microchip Application Note AN735
http://ww1.microchip.com/downloads/en/appnotes/00735a.pdf
d. I2C-bus specification and user manual
http://www.nxp.com/documents/user_manual/UM10204.pdf
e. IIC Tutorial http://www.i2c-bus.org/i2c-primer/
f. 24LC256 EEPROM Data sheet
http://ww1.microchip.com/downloads/en/devicedoc/21203m.pdf
g. PIC32 Family Reference Manual Section 24. Inter-Integrated Circuit
http://ww1.microchip.com/downloads/en/DeviceDoc/61116E.pdf
h. 32-bit Peripheral Library Guide Chapter 15 I2C Functions
http://ww1.microchip.com/downloads/en/DeviceDoc/32bitPeripheralLibraryGuide.pdf