RV College of Engineering

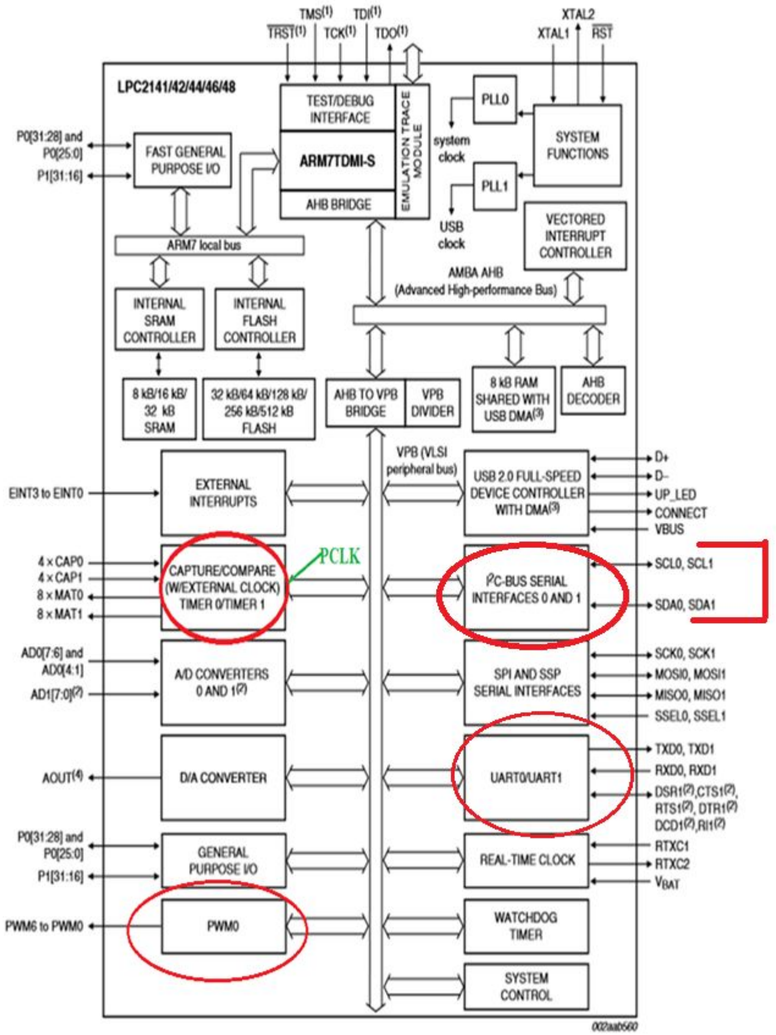
Go, change the world

Course Code:CS344AI

IOT & Embedded Computing

Programming LPC 2148 I2C and SPI Fundamentals

LPC2148 I2C bus …

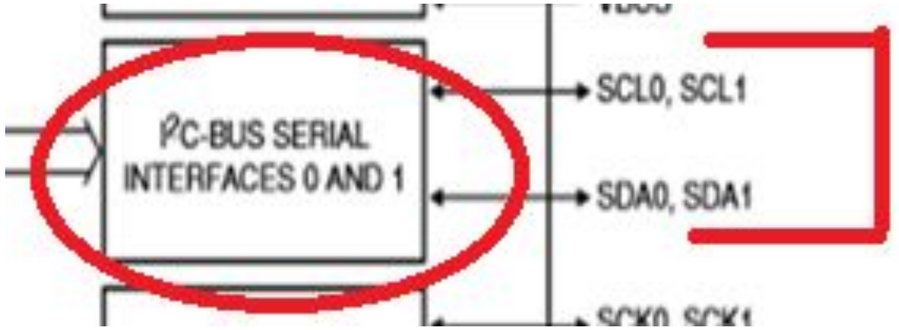


LPC 2148 provides two I2C buses

Pins relating to I2C Module of LPC2148

For I2C0 block the SCL(Clock) pin is P0.2 and SDA(Data) Pin is P0.3,

For I2C1 block the SCL pin in P0.11 and SDA pin is P0.14.



Introduction to I2C Communication…

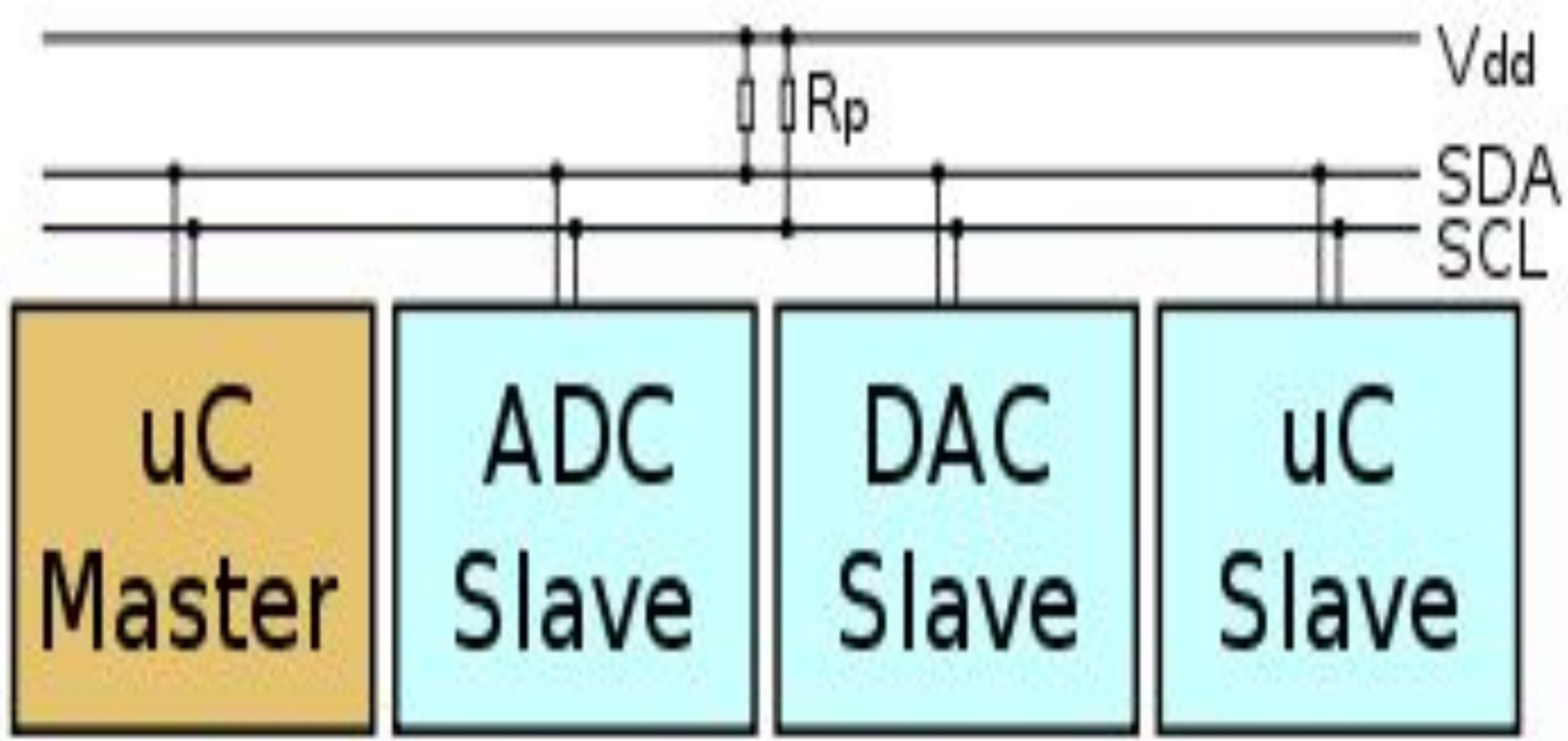
- I2C combines the best features of SPI and UARTs. With I2C, you can connect multiple slaves to a single master (like SPI) and you can have multiple masters controlling single, or multiple slaves. This is really useful when you want to have more than one microcontroller logging data to a single memory card or displaying text to a single LCD.

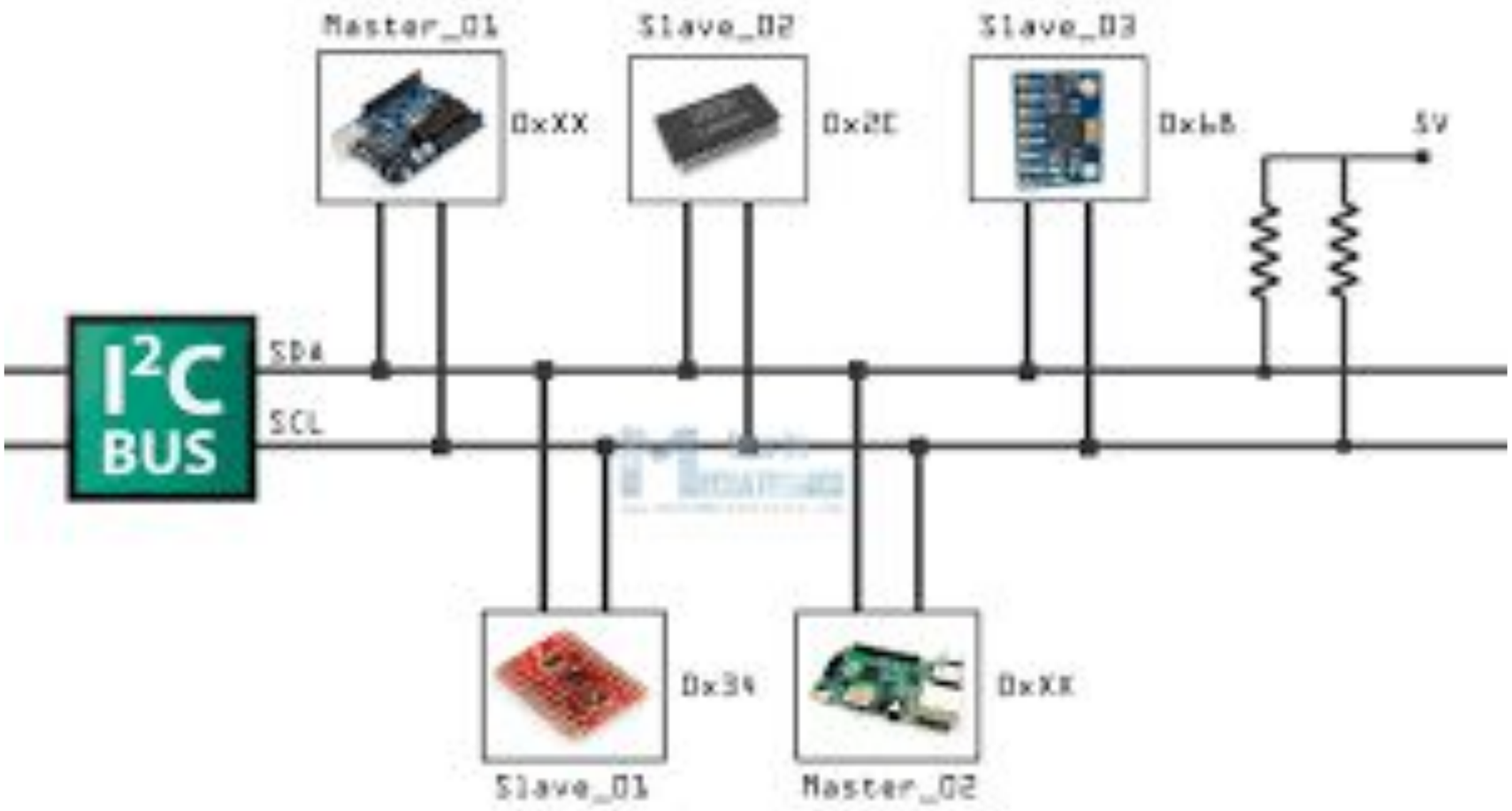
- It is multi-master, multi-slave, packet switched, single-ended, serial computer bus. It is widely used for attaching lower-speed peripheral IC’s to processors and microcontrollers in short-distance, intra board communication.

Advantages

- Only uses two wires
- Supports multiple masters and multiple slaves
- ACK/NACK bit gives confirmation that each frame is transferred successfully
- Hardware is less complicated than with UARTs
- Well known and widely used protocol

Connection Diagram..




Master_01
0xXX
Slave_02
0x2C
Slave_03
0x68
5V
I²C BUS
SDA
SCL
0x34
Slave_01
0xXX
Master_02

Master node

Node that generates the clock and initiates communication with slaves.

Slave node

Node that receives the clock and responds when addressed by the master

Modes of operation for I2C bus device ..

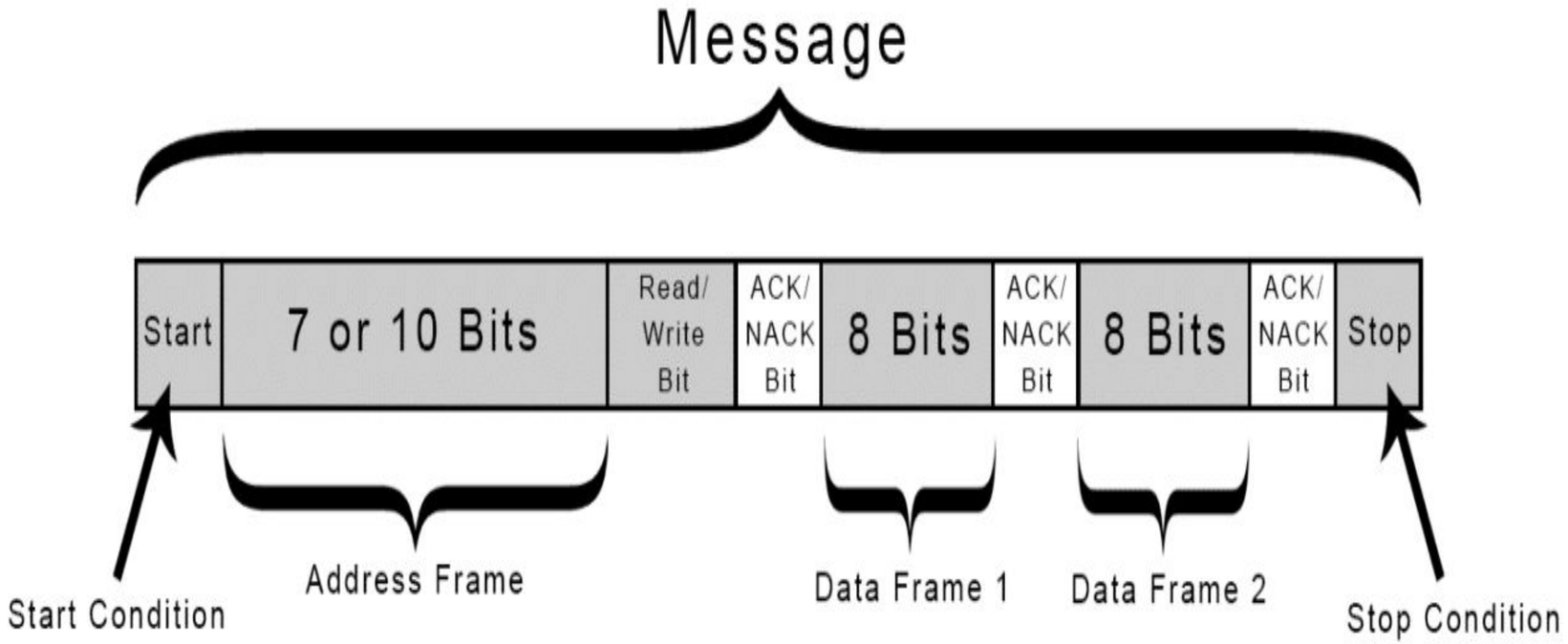
1. Master transmit – master node is sending data to a slave
2. Master receive – master node is receiving data from slave
3. Slave transmit – slave node is sending data to the master
4. Slave receive – slave node is receiving data from the master

(although most devices only use a single role and its two modes)

I2C Features…

- Developed by Philips in 1982, revised the specification in 1992 (intels SMBus developed on similar lines to I2C)
- Protocol is Serial, half-duplex
- Supports speeds of 100Khz,400Khz (fast mode), 1Mhz –fast mode plus,3.4Mhz for high speed mode, 5Mhz for ultra fast mode
- It is recommended up to 2-3 meters.
- Supports 7bit address (128 devices)/10 bit address (1008 devices)
- Signals: It requires 2 signals
  SDA – serial data
  SCL - serial clock
- The clock signal is always generated by the current bus master; at times some slave devices may force the clock low at times to delay the master sending more data, called as clock stretching.
- Support multi master system, more than one master can communicate with the devices, for every 8 bits of data sent, one extra bit of meta data (ACK/NACK bit) must be transmitted. The hardware required to implement I2C is more complex than SPI but simpler than UART

I2C Protocol



Steps of I2C Data Transmission..

- The master sends the start condition to every connected slave by switching the SDA line from a high voltage level to a low voltage level before switching the SCL line from high to low
- The master sends each slave the 7 or 10 bit address of the slave it wants to communicate with, along with the read/write bit
- Each slave compares the address sent from the master to its own address. If the address matches, the slave returns an ACK bit by pulling the SDA line low for one bit. If the address from the master does not match the slave's own address, the slave leaves the SDA line high.
- The master sends or receives the data frame:
- After each data frame has been transferred, the receiving device returns another ACK bit to the sender to acknowledge successful receipt of the frame:
- To stop the data transmission, the master sends a stop condition to the slave by switching SCL high before switching SDA high:

Registers used for programming LPC2148 I2C block

I2C0CONSET (8 bit) – I2C control set register: The bits in this register control the operation of the I2C interface. Writing a 1 to a bit of this register causes the corresponding bit in the I2C control register(inside I2C block) to be set. Writing a 0 has no effect. This is a Read-Write register.

I2C0CONCLR (8 bit) – I2C control clear register. This register is used to clear bits in I2C0CONSET register. Writing 0 no effect. The bit locations are same as that of I2C0CONSET register given above. Its a Write only register.

I2C0STAT (8 bit) – This gives the current state of I2C interface in form of state codes. This is a read

I2C0DAT (8 bit) – This register contains the data that is to be transmitted or the latest received data. Data in this register is always shifted from right to left i.e. the first bit to be transmitted is the MSB (bit 7), and after a byte has been received, the first bit of received data is located at the MSB of I2C0DAT.

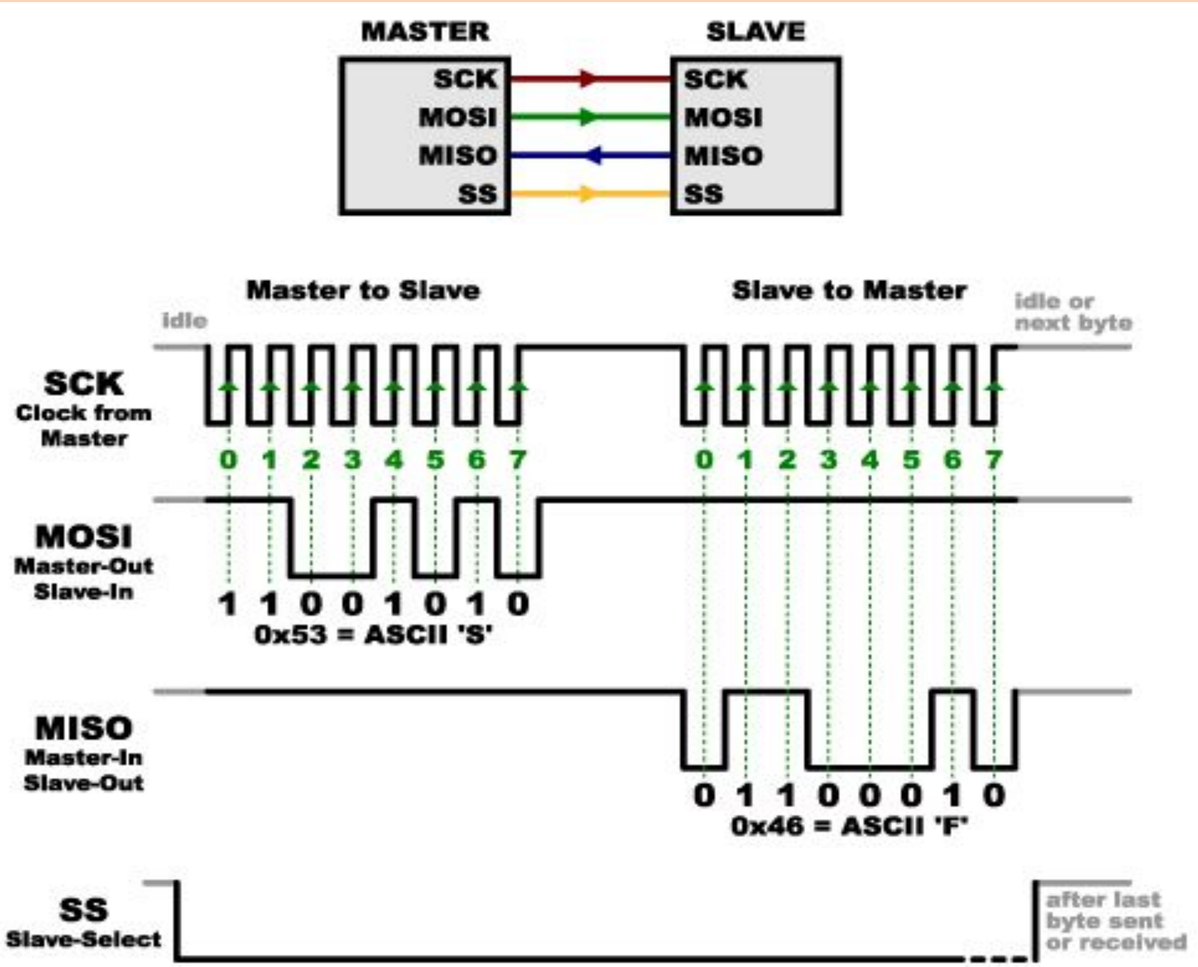
I20SCLH (16 bit) – This register is used to store the High time period of the SCL pulse.

I20SCLL (16 bit) – This register is used to store the Low time period of the SCL pulse.

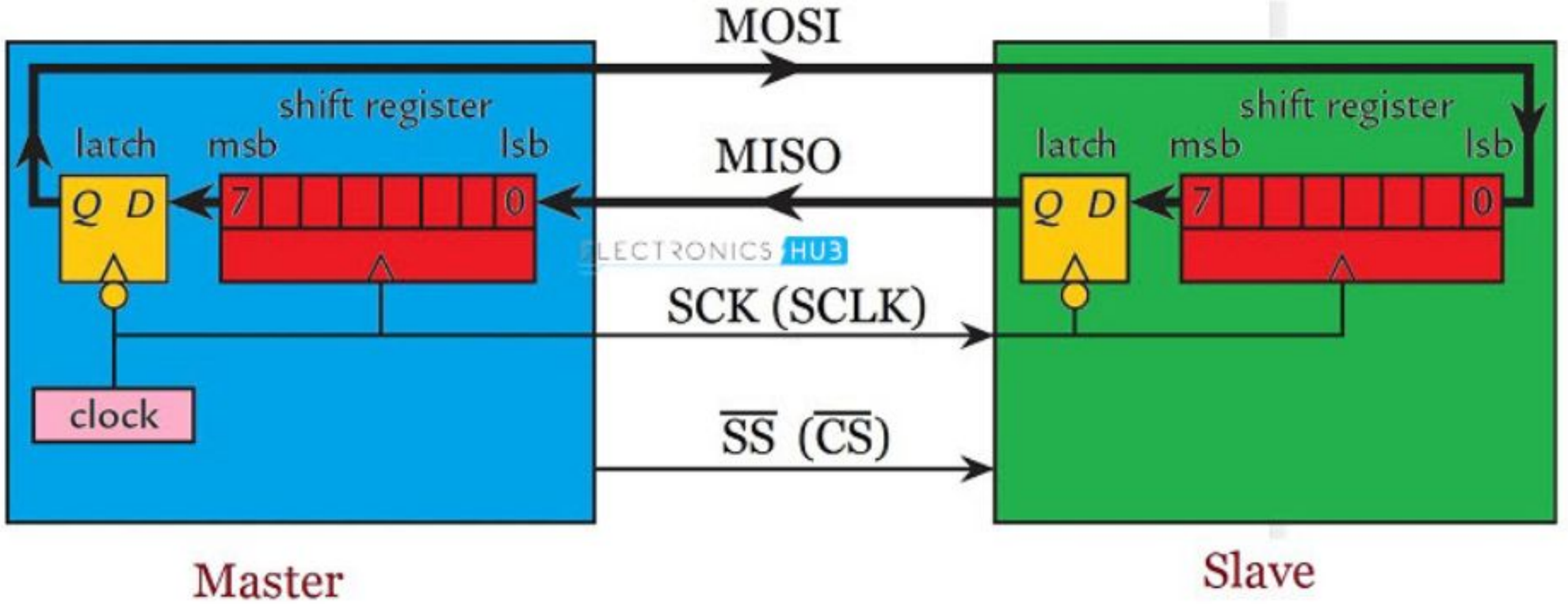
I2C0ADR (8 bit) – I2C Slave Address register : Not applicable for master mode. Used to store the address in slave mode.

SPI - Serial Peripheral Interface

The Serial Peripheral Interface (SPI) is a bus interface connection protocol originally started by Motorola Corp.



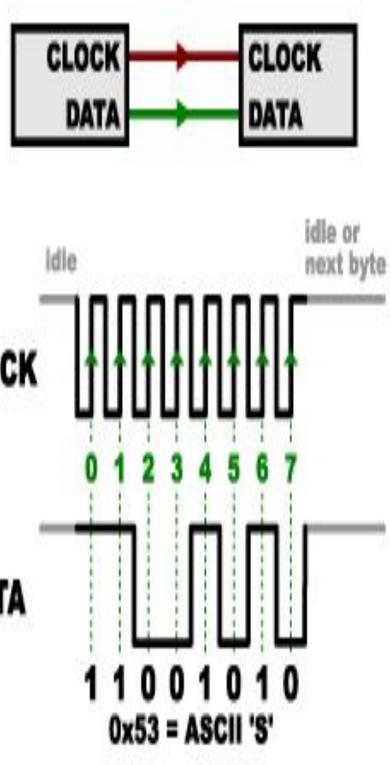
(


MOSI
MISO
SCK (SCLK)
SS (CS)
shift register
latch
msb
lsb
Q D
7
0
clock
ELECTRONICS HUB
Master
Slave

(

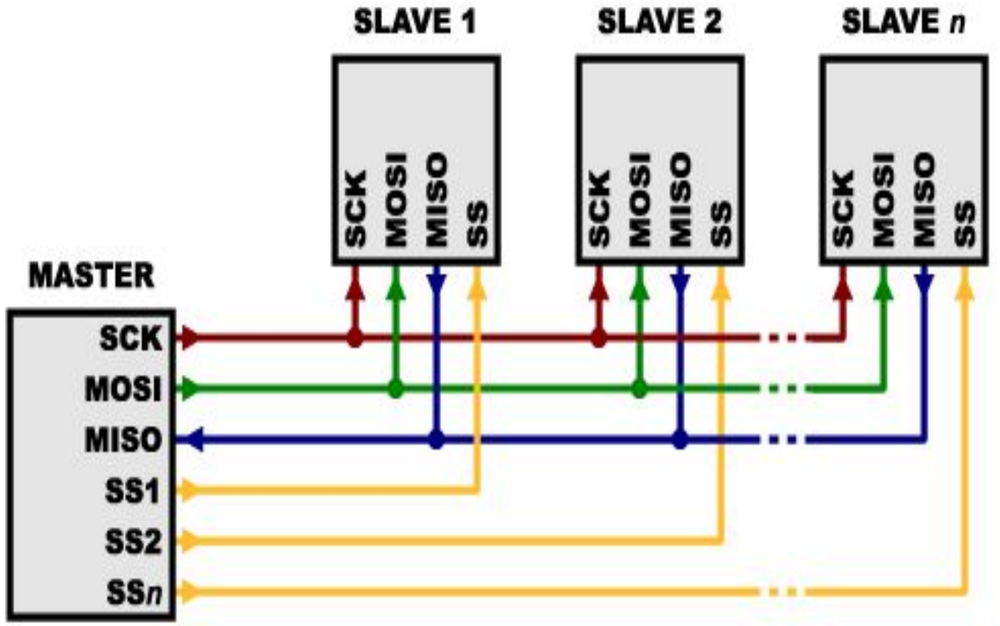
SPI - Features

- SPI Interface uses four wires for communication. Hence it is also known as four wire serial communication protocol.
- SPI is a full duplex master-slave communication protocol. This means that only a single master and a single slave can communicate on the interface bus at the same time.
- SPI enabled devices work in two basic modes of SPI operation i.e. SPI Master Mode and SPI Slave Mode.
- Master Device is responsible for initiation of communication.
- Master Device generates Serial Clock for synchronous data transfer.
- Master Device can handle multiple slave devices on the bus by selecting them one by one.



One Master – Many Slaves

- Used to send /receive data between microcontrollers and small peripherals such as shift registers, sensors, Graphics LCD displays, flash memory chips and SD cards
- Hardware requirement for SPI is very simple compare to UART & I2C
- SPI is full duplex (has separate send & receive lines unlike I2C)
- Faster than asynchronous serial (UART)
- It supports multiple slaves



SPI Pins…

MOSI – Master Out Slave In – Used to send serial data from Master to slave

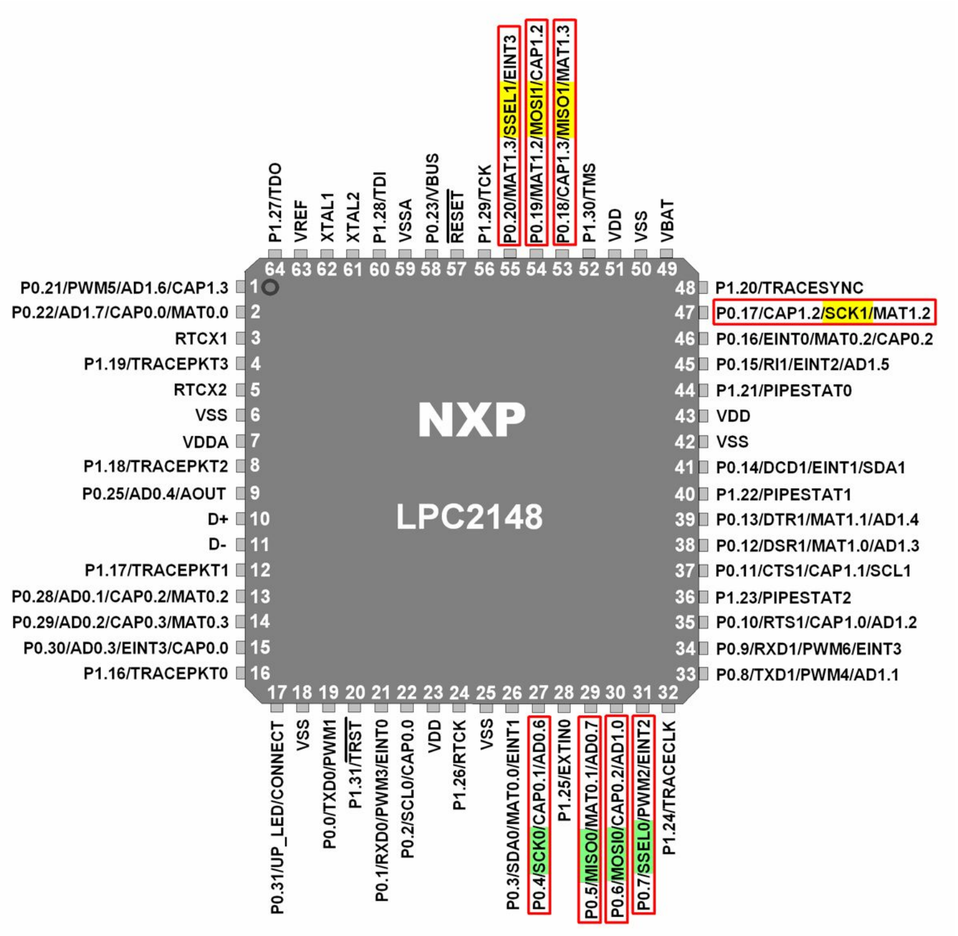
MISO - Master In Slave Out – Used to send data from Slave to Master , If the slave needs to send a response back to the master, the master will continue to generate a prearranged number of clock cycles

SCK – Serial Clock - Serial clock generated by the Master, for use by both master & clock, The clock is an oscillating signal that tells the receiver exactly when to sample the bits on the data line. This could be the rising (low to high) or falling (high to low) edge of the clock signal; the datasheet will specify which one to use. When the receiver detects that edge, it will immediately look at the data line to read the next bit

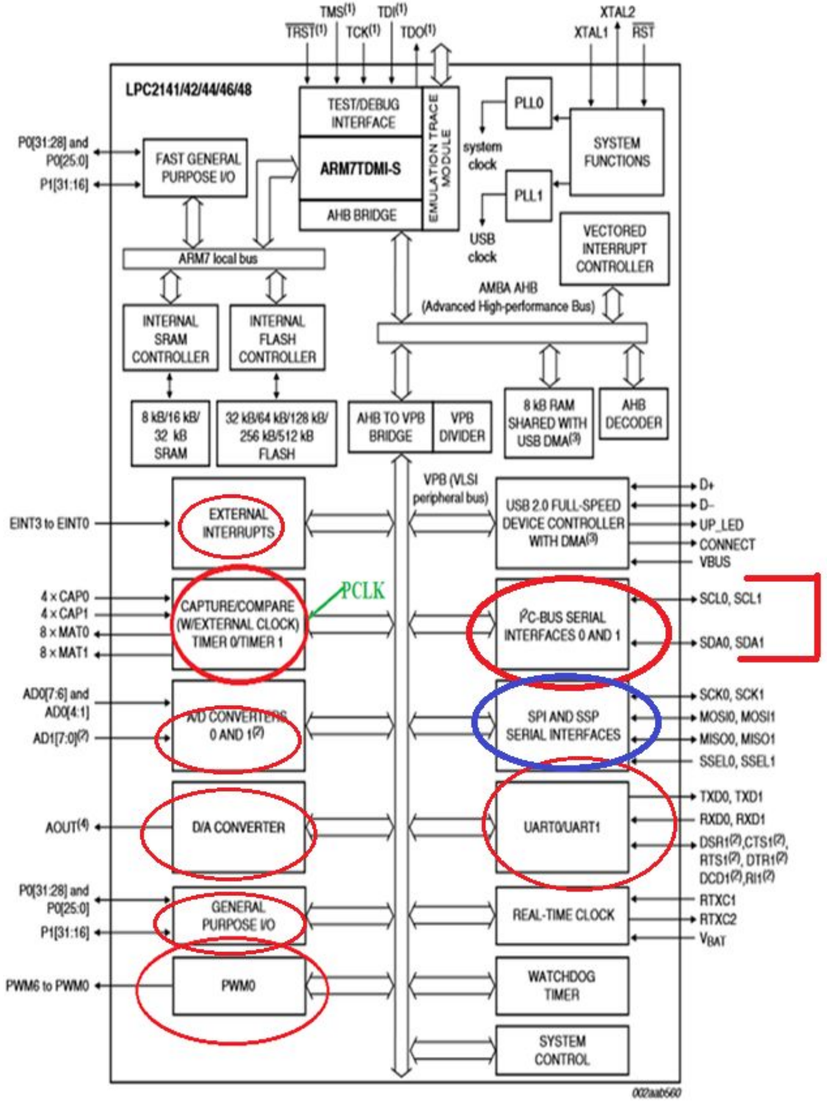
SS – Slave Select - This tells the slave that it should wake up and receives / send data and is also used when multiple slaves are present to select the one you'd like to talk to. The SS line is normally held high, which disconnects the slave from the SPI bus. Just before data is sent to the slave, the line is brought low, which activates the slave. When you're done using the slave, the line is made high again.

LPC2148 SPI

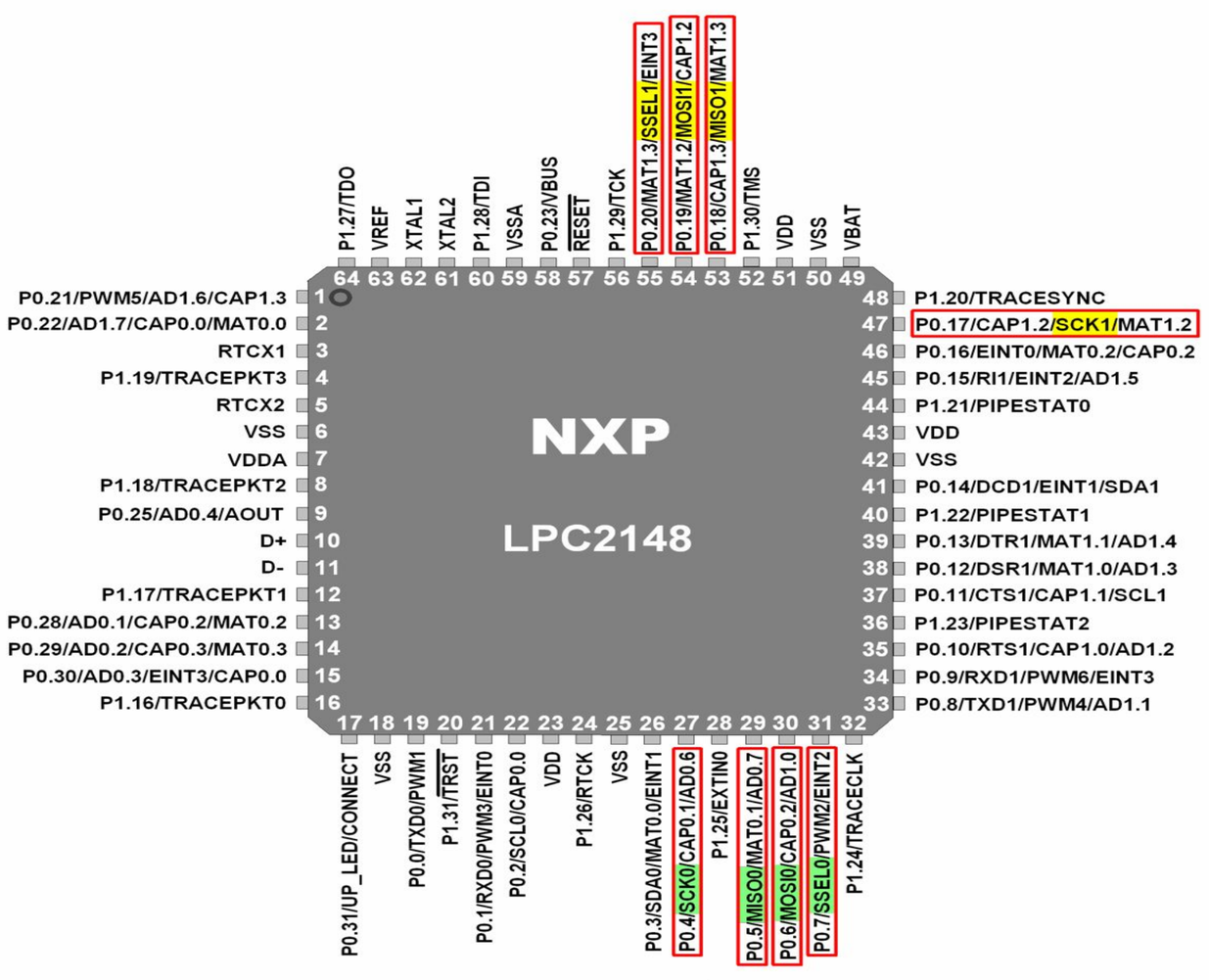
- LPC2148 has two inbuilt SPI modules **SPI0** and **SPI1/SSP (Synchronous Serial Port).**
- SPI0 supports variable (8 to 16) bits of data per transfer. SPI0 module is compatible with Motorola SPI (SPI0).
- SSP/SPI1 supports variable (4 to 16) bits of frame.SSP/SPI1module is compatible with Motorola SPI (SPI1), Texas Instruments SSI format, National Semiconductor Microwire format bus interface.



LPC2148 SPI block



LPC2148 SPI PINS



Registers used for programming LPC2148 SPI block

SPI0 Registers

1.S0SPCR (SPI0 Control Register) : It is used to control the function of SPI block, data width(8 to 16bits), Clock Edge and polarity selection for copying the data, configure to send msb/lsb bit first etc.

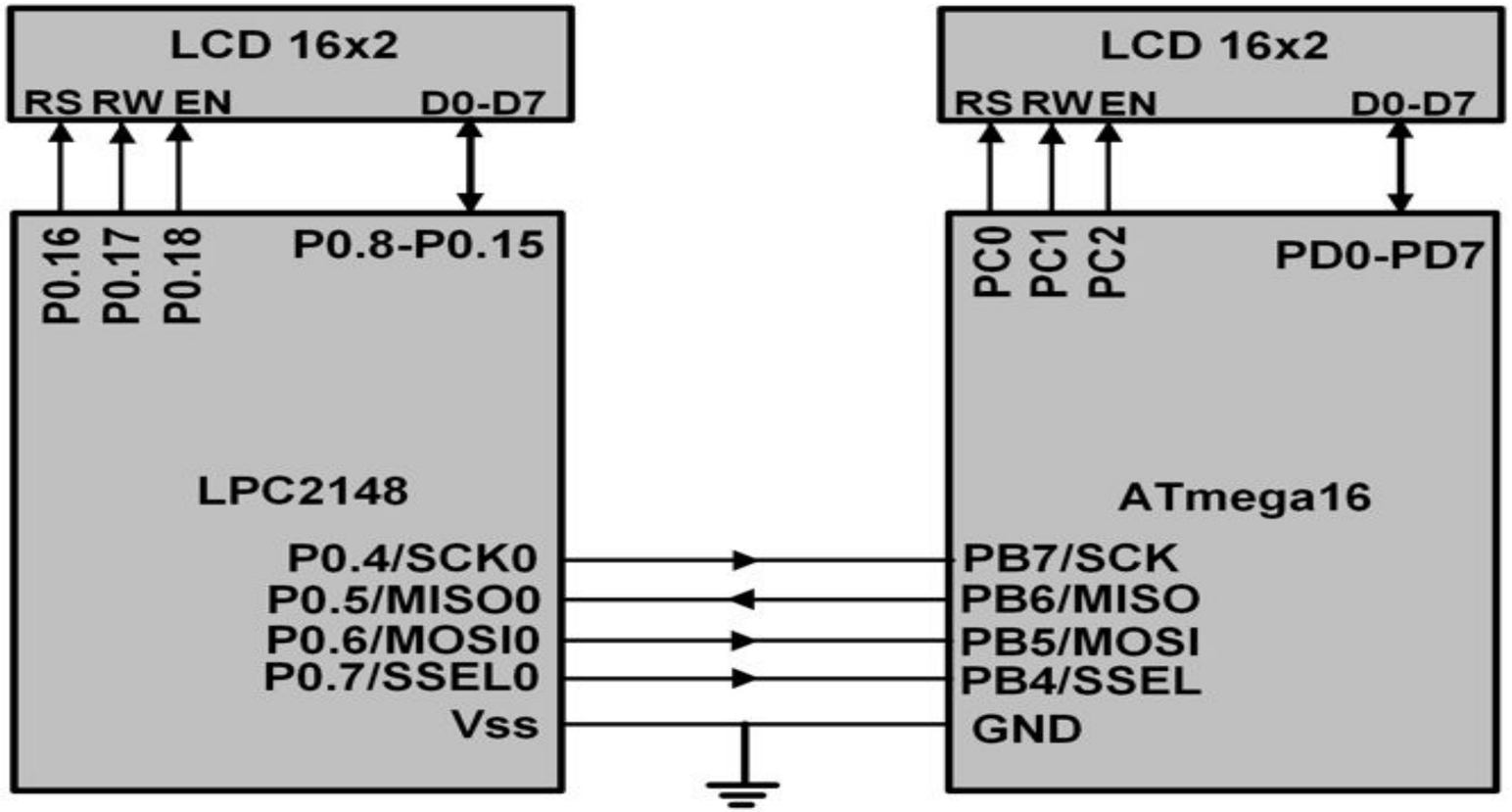
2.S0SPSR (SPI0 Status Register) : Indicate transfer complete and/or any errors during transmission

3.S0SPDR (SPI Data Register). It is 16 bit register, holds data to be sent/received (8 to 16 bits can be sent)

4.S0SPCCR (SPI0 Clock Counter Register): This register controls the frequency of master SCK (Serial Clock) in terms of PCLK

SPI Pins	Pin Direction in Master Mode	Pin Direction in Slave Mode
MOSI	Output	Input
MISO	Input	Output
SSEL	Output	Input
SCK	Output	Input

LPC2148 and Atmega16 with SPI Interface



Initialization for Master (LPC2148)

- Using pin select register (PINSEL), configure P0.4, P0.5, P0.6 and P0.7 as SCK0, MISO0, MOSI0 and GPIO respectively. SSEL (P0.7) is configured as general purpose output pin in order to select slave device.
- Using S0SPCR, SPI master mode is selected with 8-bit data and CPOL = CPHA =0. S0SPCR value will change according to configuration required.
- Select clock pre-scalar using S0SPCCR.

```
void SPI_Master_Init()

{

        PINSEL0 = PINSEL0 | 0x00001500; /* Select P0.4, P0.5, P0.6, P0.7 as SCK
0, MISO0, MOSI0 and GPIO */

        S0SPCR = 0x0020; /* SPI Master mode, 8-bit data, SPI0 mode */

        S0SPCCR = 0x10; /* Even number, minimum value 8, pre scalar for SPI Cl
ock */

}
```

Steps for SPI0 Data Transfer

SPI0 Master Write Mode (LPC2148)

- Make SSEL pin low (we have configured SSEL pin as GPIO to select slave) using IOCLR to select slave.
- Load data to be written into the data register.
- Wait till transmission is completed, i.e. till the SPIF bit becomes 1.
- Make SSEL pin high using IOSET to deselect the slave and to disable SPI transmission to that device.

```
void SPI_Master_Write(char data)
{
        char flush;
        IO0CLR = (1<<7);  /* SSEL = 0, enable SPI communication with slave */
        S0SPDR = data;  /* Load data to be written into the data register */
        while ( (S0SPSR & 0x80) == 0 );  /* Wait till data transmission is comp
leted */
        flush = S0SPDR;
        IO0SET = (1<<7);  /* SSEL = 1, disable SPI communication with slave */
}
```

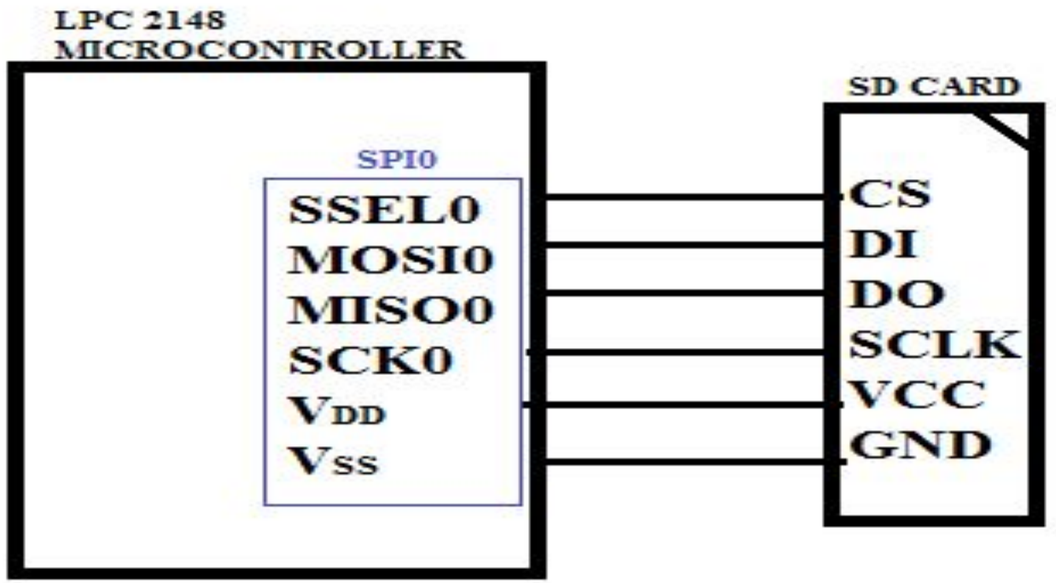
SPI0 Master Read Mode (LPC2148)

- Make SSEL pin low (we have configured SSEL pin as GPIO to select slave) using IOCLR to select slave.
- Transmit Flush Byte.
- Wait till data transmission is completed. When data transmission is completed, data sent by slave will be received.
- Make SSEL pin high using IOSET to deselect the slave and to disable SPI transmission to that device.
- Read the data in the data register.
  This is the received data.

```
char SPI_Master_Read()
{
        IO0CLR = (1<<7);  /* SSEL = 0, enable SPI communication with slave */
        S0SPDR = 0xFF;  /* Transmit Flush byte */
        while ( (S0SPSR & 0x80) == 0 );  /* Wait till data transmission is comp
leted */
        IO0SET = (1<<7);  /* SSEL = 1, disable SPI communication with slave */
        return S0SPDR;  /* Return the data received */
}
```

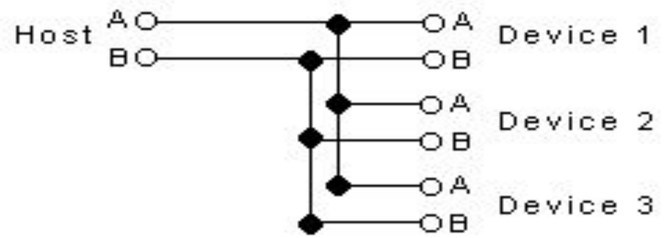
(

SD card interface :



Characteristics of RS232 and RS485		
Parameter	**RS232**	**RS485**
Cabling	Single-edned	Differential
Numbers of devices	1 transmitter 1 receiver	32 transmitters 32 receivers
Mode of operation	Simplex or full duplex	Simplex or half duplex
Maximum cable length	50 feet	4000 feet
Maximum data rate	20 kbits/s	10 Mbits/s
Signaling	unbalanced	balanced
Typical logic levels	±5 ~ ±15V	±1.5 ~ ±6V
Minimun receiver input impedance	3~7kΩ	12kΩ
Receiver sensitivity	±3V	±200mA

Typical 2-Wire RS-485 Wiring



(

Course Handling faculty

Name	Dr. Badari Nath K.
Mobile	9945124747
Email	badarinath.kb@rvce.edu.in