SPI REGISTERS IN AVR ATMEGA CONTROLLERS

SPI Registers:
There are three SPI registers in AVR namely:
SPDR (SPI data register)
SPCR (SPI control register)
SPSR (SPI status register)

SPDR (SPI data register)
This register is a read / write register.to write into SPI shift register , the data must be written in SPDR . writing to SPDR initiates data transmission . while previous transmission is in progress,writing into SPDR is restricted (write in SPDR only when the last byte is transmitted completely) . To read the SPI shift register we need to read the SPDR.

7	6	5	4	3	2	1	0
MSB							LSB

SPCR (SPI control register)

7	6	5	4	3	2	1	0
SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0

- SPIE : Stands for SPI interrupt enable .
  - Setting this to 1 enables the SPI interrupt
- SPE : Stands for SPI enable
  - Setting this bit to 1 enables SPI
- DORD : Stands for DATA ORDER
  - THis bit lets you choose which bit to transmit first .
  - Setting this to one transmits LSB first
  - Resetting this bit to zero transmits MSB first
- MSTR : stands for master/slave select
  - Setting this bit to one selects master mode.
  - Resetting this bit to zero selects slave mode .
  - *If ss pin is configured to input and is driven low while master is set , it resets (clears) the MSTR bit . and SPIF flag gets set.*
- CPOL : Stands for clock polarity
  - This bit sets base value of clock when it is idle.
  - CPOL = 0 sets base value of clock zero

        - CPOL = 1 sets base value of clock one

- CPHA : Stands for clock phase.
        - Decides when to sample and when to toggle the bit
        - CPHA = 0 means sample on leading (first) clock edge and toggle on trailing (second) clock edge.
        - CPHA = 1 means sample on trailing (second) clock edge and toggle on leading (first) clock edge.

- SPR1 : Stands for SPI clock rate select bit1
- SPR0 : Stands for SPI clock rate select bit1
        - These two bits (SPR1 , SPR0) controls the serial clock rate of device in master mode.

Mode	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

SPSR (SPI status register)

7	6	5	4	3	2	1	0
SPIF	WCOL						SPI2X

- SPIF : Stands for SPI interrupt flag .
        - In master mode , this bit is set in two situations:
            - When serial transfer is complete
            - When ss pin is an input and is driven low by external device.
        - Setting this bit one will cause an interrupt if SPIE in SPCR is set and global interrupts are enabled.

- WCOL : Stands for write collision flag .
        - This bits sets if you write on SPDR during ongoing data transfer.

- SPI2X : Stands for double SPI speed
        - Setting this bit doubles the spi speed , by doubling the clock rate.

Serial clock frequency:

SPI2X	SPR1	SPR0	SCK frequency
0	0	0	Fosc / 4
0	0	1	Fosc / 16
0	1	0	Fosc / 64
0	1	1	Fosc / 128
1	0	0	Fosc / 2 (not recommended)
1	0	1	Fosc / 8
1	1	0	Fosc / 32
1	1	1	Fosc / 64

Slave select PIN (SS)

- Used to initiate and terminate the data transfer.
- *In master mode :*
    - This pin can be made either output or input
        - If output
            - Spi circuit of avr will not control SS pin and we can make it zero or one by the software
        - If input
            - It will control the function of SPI .
            - We need to make this pin high externally to ensure master operation.
            - If the external device makes this pin low , the SPI module stops working in master mode and switches to slave mode by clearing the MSTR bit i SPCR and then sets SPIF bit in SPSR.
    - It is recommended to make SS pin output while working in master mode.
- *In Slave mode :*
    - SS pin is always input and cannot be controlled by the software.
    - We should hold it externally low to activate the SPI
    - If it is high
        - SPI is disabled / resets
        - All SPI pin are made input
        - SPI module will immediately clear any partially received data in shift register.
- In slave mode if SS in is driven high by an external device ,The spi module resets but not disabled and is not necessary to enable it again.

Master mode operation

- MSTR bit shall be set to one (high)
- Set serial clock frequency by setting the values of SPI2X,SPR1,SPR0 according to Serial clock frequency table
- Enable SPI by setting the SPIE bit to one
- Write a byte to SPDR register (starts data exchange by starting SPI clock generator)
- After shifting 8 bits SPI clock generator stops and SPIF flag sets to one.
- To get the received data , SPDR needs to be read before the next byte arrives.
- Either interrupts can be used or we can poll SPIF to know when data is exchanged.
- In case of multi byte brust write , master continues to shift the next byte by writing into SPDR , in order to indicate end of packet SS line is pulled high.
- When AVR is configured as master, spi line will not control SS pin ,if we want to make SS high or low , we have to do it by writing 1 or 0 to SS bit of PORT B.

Source code:

Initialize spi for master in mode 0 having clk frequency Fosc/16 ,transmit “G continuously and display the received data on port A:

```
#include<avr/io.h>
#define F_CPU 1000000UL
#define CPOL   3
#define CPHA   2
#define MOSI     5
#define SCLK    7

Int main(void)
{
DDRB = (1<<MOSI) | (1<< SCLK);
DDRA = 0xff;
SPCR = ((1<<SPE) | (1<<MSTR) | (1<SPR0) | (1<<CPOL) | (1<< CPHA));
while(1)
{
SPDR = ‘G’;
While ( ! (SPSR & (1<< SPIF) ) );
PORTA = SPDR;
}
Return 0;
}
```

Slave mode operation

- MSTR bit shall be set to zero (low)
- Set MISO as OUTPUT
- Set SPE for enabling the spi
- When AVR is configured as slave the function of SPI interface depends on SS pin.
    - If SS pin is driven high , MISO is tristated and SPI interface sleeps, only the content of SPDR can be updated in this state.
    - When SS is driven low , the data will be shifted by incoming clock pulse on the SCK pin.
        - SPIF changes to one when last bit of a byte is shifted completely.
- Slave can put new data to be sent in SPDR before reading the incoming data. This is because AVR has two one byte buffers to store received data.
- In slave mode there is no need to set SCK frequency because the SCK is generated by the master
- Spi mode must be selected
- Enable the SPI by setting SPIE bit.

Source code :

Initialize spi for slave in mode 0 having clk frequency Fosc/16 ,transmit 'S' continuously and display the received data on port A:

```
#include <avr/io.h>
#define CPOL   3
#define CPHA   2
#define MISO    6
Int main (void)
{
DDRA = 0xff;
DDRB = (1<< MISO);
SPCR = ( (1<< SPE) | (1<< CPOL) | (1<< CPHA) );
While (1)
{
SPDR = 'A';
while(! ( SPSR & (1<<SPIF) ) );
PORTA = SPDR;
}
return 0;
}
```