

Microprocessor based digital Systems

Serial Communications: SPI

Guillermo Carpintero

Marta Ruiz

Universidad **Carlos III** de Madrid

Communication Standards

Interface	Format	Num. Devices	Separation max.	Tx Speed (bits/seg)
SPI	Serie Sinc.	8	3 m	2.1M
I2C	Serie Sinc.	40	5.5 m	400k

SPI, Fundamentals

Serial Peripheral Interface (SPI)

SPI is the name that Motorola gave to this interface on their microcontrollers. Corresponds to the **Microwire** interface (Trade Mark from National Semiconductor).

Oriented to be high speed communications interface among devices

Characteristics

Serial Synch

4 wires (SDI, SDO, SCK, SS)

Master-slave

line SCK, controlled by Master

Bidirectional

SPI, Fundamentals

Serial Synch Interface

three wire

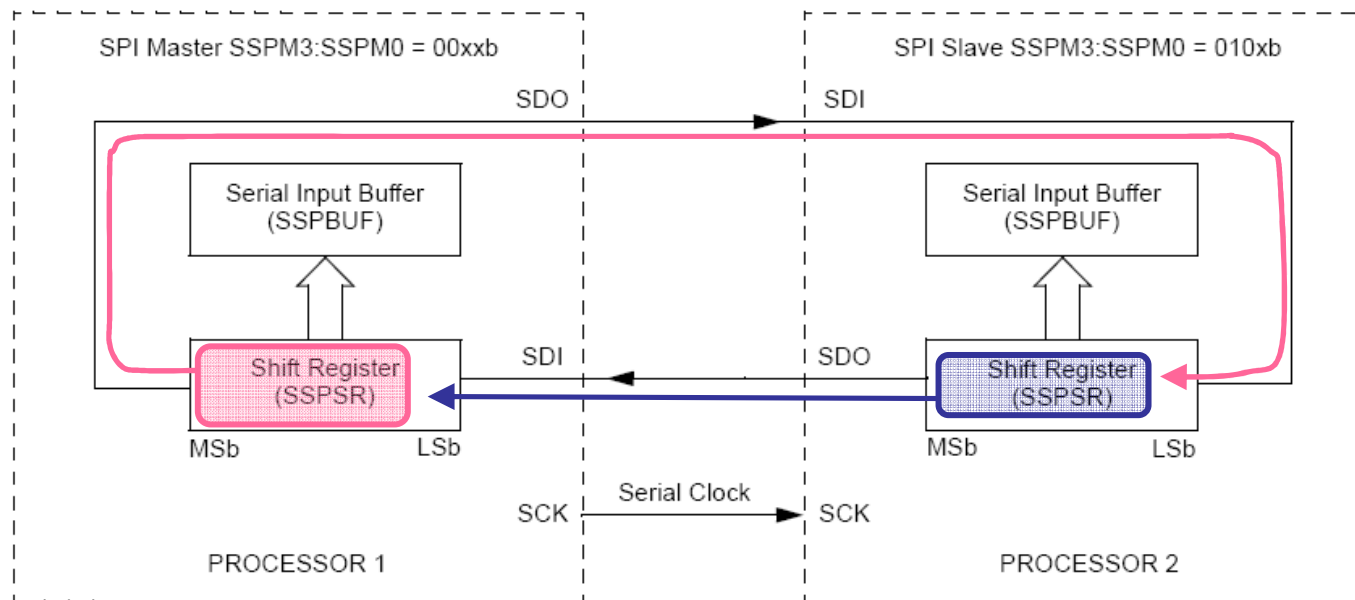
SCK (Serial CLOCK)

SDI (Serial Data Input)

SDO (Serial Data Output)

1 Master

Slave

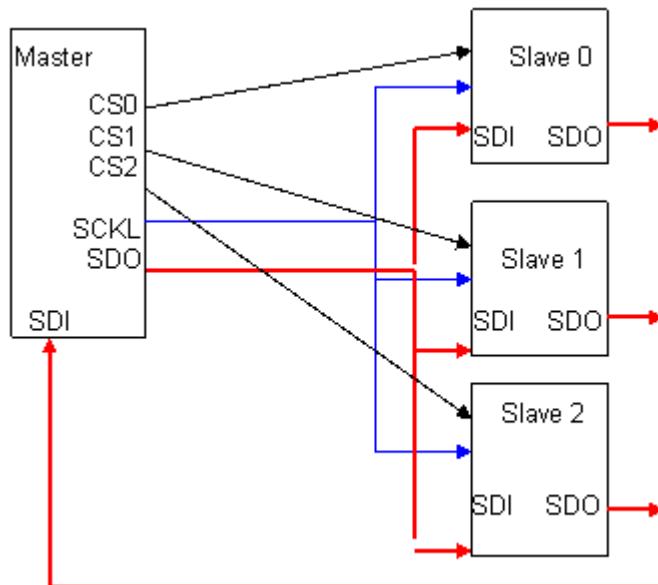
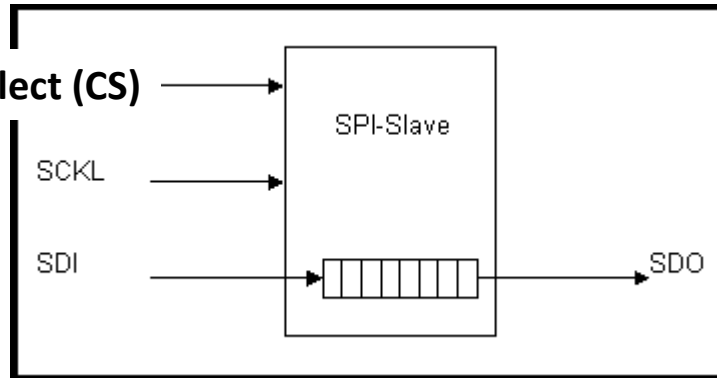


To send or receive a data through the SPI interface, the master must always write a data in the transmission buffer, SSPBUF. Three scenarios:

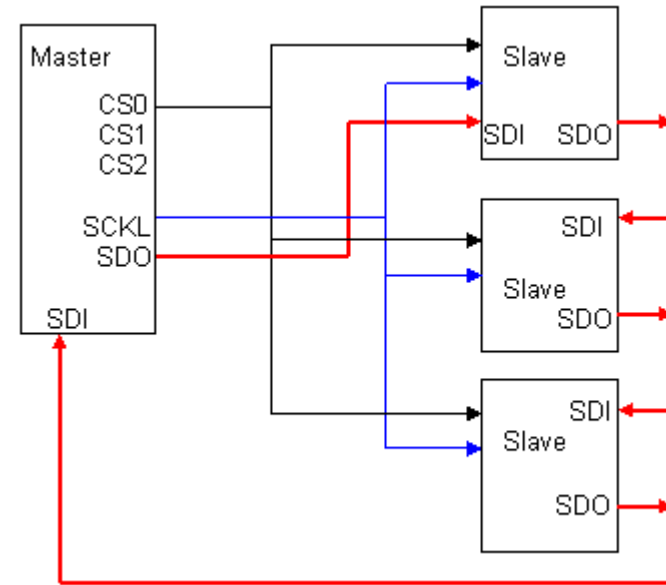
- Master sends data – (Slave sends dummy data)
- Master sends data – Slave sends data
- Master sends dummy data – Slave sends data

SPI, Slave Connexion Scheme

Slave Select (SS), Chip Select (CS)



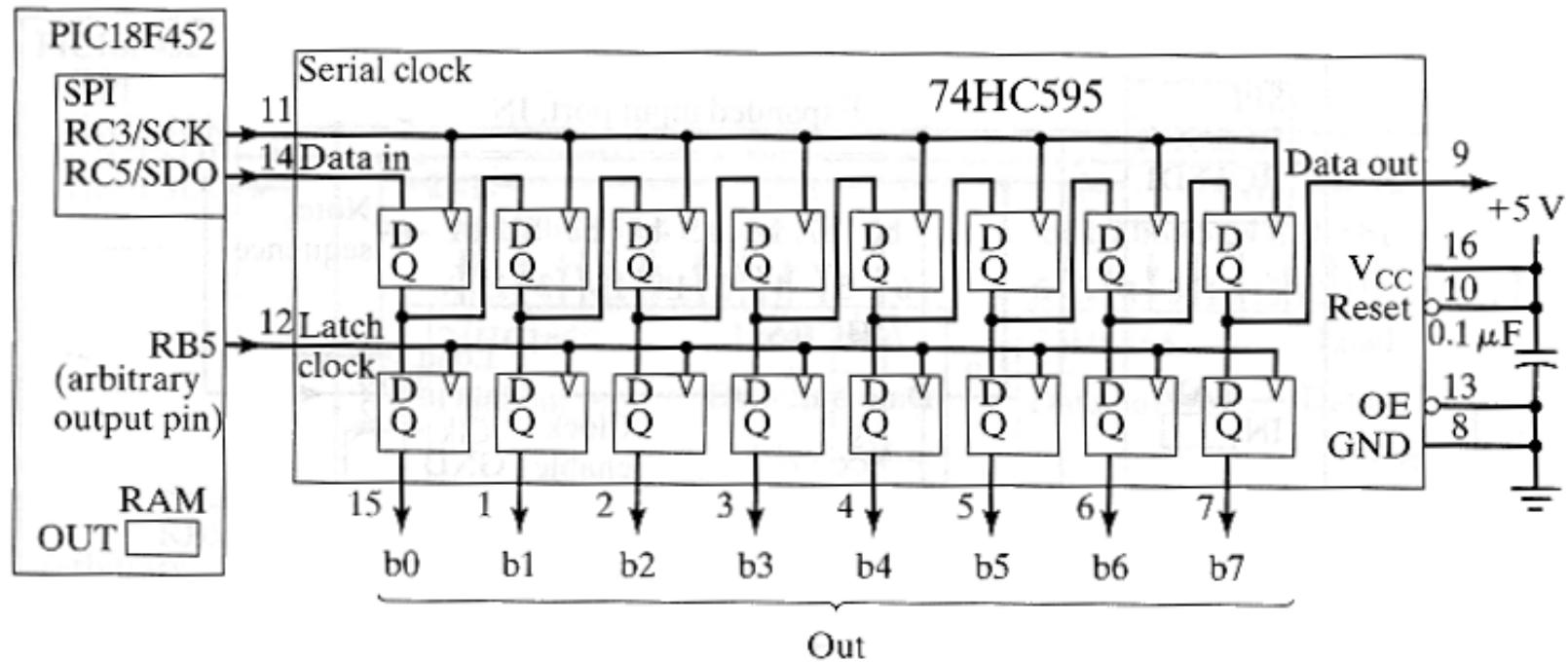
Parallel Scheme



Daisy Chain Scheme

SPI, Applications

PORT pin extension - OUTPUT



SPI, PIC18 Peripheral

Implements Master and Slave functions

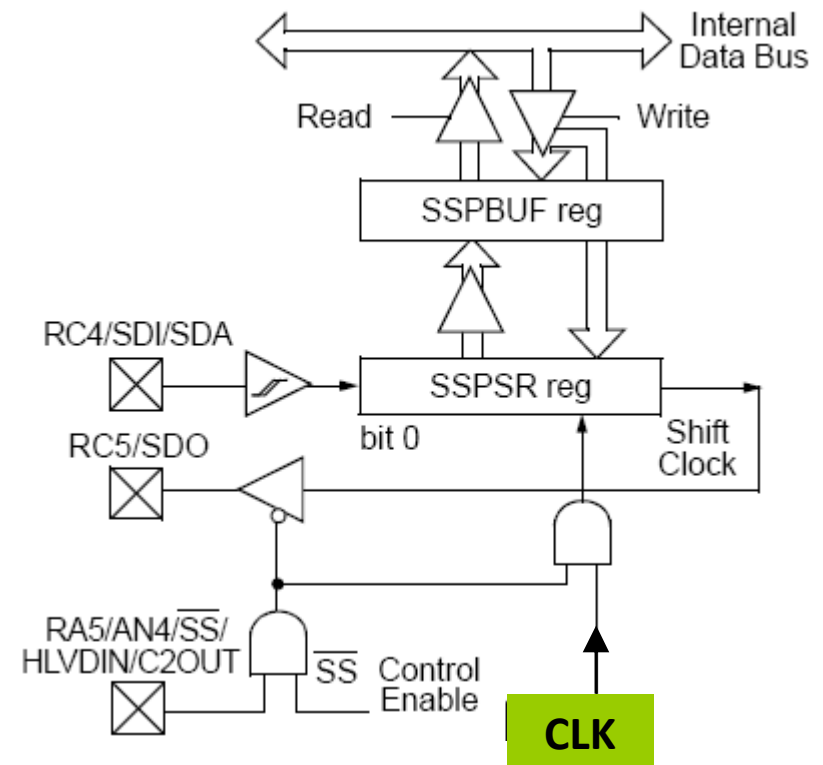
- Serial Data Out (SDO) – RC5/SDO
- Serial Data In (SDI) – RC4/SDI/SDA
- Serial Clock (SCK) – RC3/SCK/SCL

Additionally, a fourth pin may be used when in a Slave mode of operation:

- Slave Select (\overline{SS}) – RA5/AN4/ \overline{SS} /HLVDIN/C2OUT

TRIS Configuration:

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<5> bit cleared
- SCK (Master) must have TRISC<3> bit cleared
- SCK (Slave) must have TRISC<3> bit set
- SS must have TRISA<5> bit set



SPI, PIC18 Peripheral

SPI Activation

SSPCON1: MSSP CONTROL REGISTER 1 (SPI MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7						bit 0	

bit 5 **SSPEN:** Synchronous Serial Port Enable bit

1 = Enables serial port and configures SCK, SDO, SDI and \overline{SS} as serial port pins

0 = Disables serial port and configures these pins as I/O port pins

Note: When enabled, these pins must be properly configured as input or output.

bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits

0101 = SPI Slave mode, clock = SCK pin, \overline{SS} pin control disabled, \overline{SS} can be used as I/O pin

0100 = SPI Slave mode, clock = SCK pin, \overline{SS} pin control enabled

0011 = SPI Master mode, clock = TMR2 output/2

0010 = SPI Master mode, clock = $F_{osc}/64$

0001 = SPI Master mode, clock = $F_{osc}/16$

0000 = SPI Master mode, clock = $F_{osc}/4$

Note: Bit combinations not specifically listed here are either reserved or implemented in I²C mode only.

SPI, PIC18 Peripheral

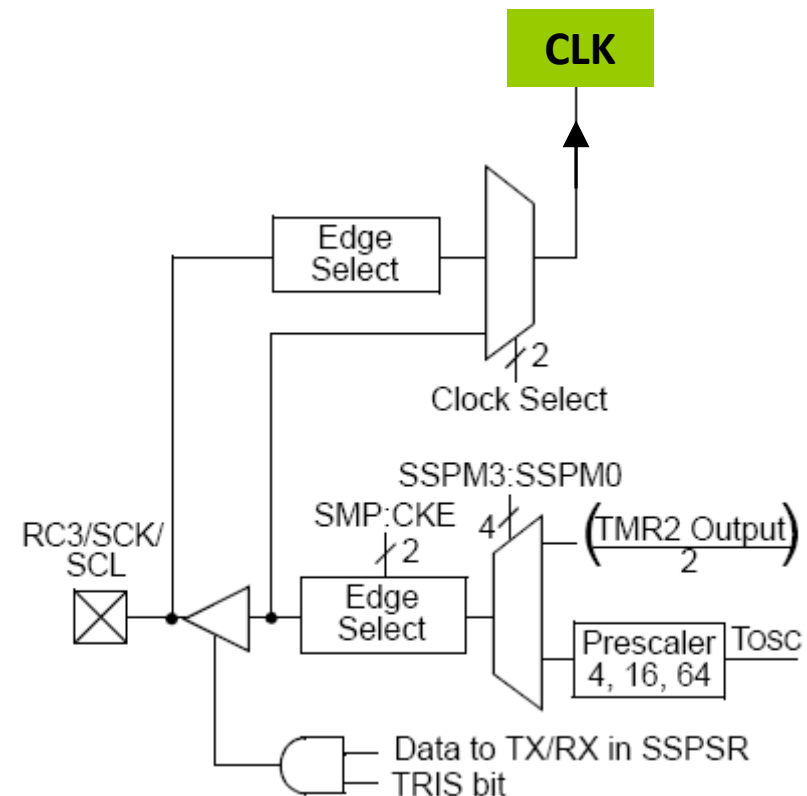
Transfer Clock Generation

The transfer speed, which is the frequency of the SPI clock signal (SCK), is set by the slowest peripheral connected to the bus

SCK frequencies:

- FOSC/4,
- FOSC/16, o
- FOSC/64.

If non of these is valid, can be generated from TMR2.



SPI, PIC18 Peripheral

Clock options on the SCK signal

- Clock Polarity – Idle state of SCK

CKP bit, SSPCON1<4>

1 = Idle state for clock is a high level

0 = Idle state for clock is a low level

- Input line sample instant – In the **Middle** or at the **End**

SMP bit, SSPSTAT<7>

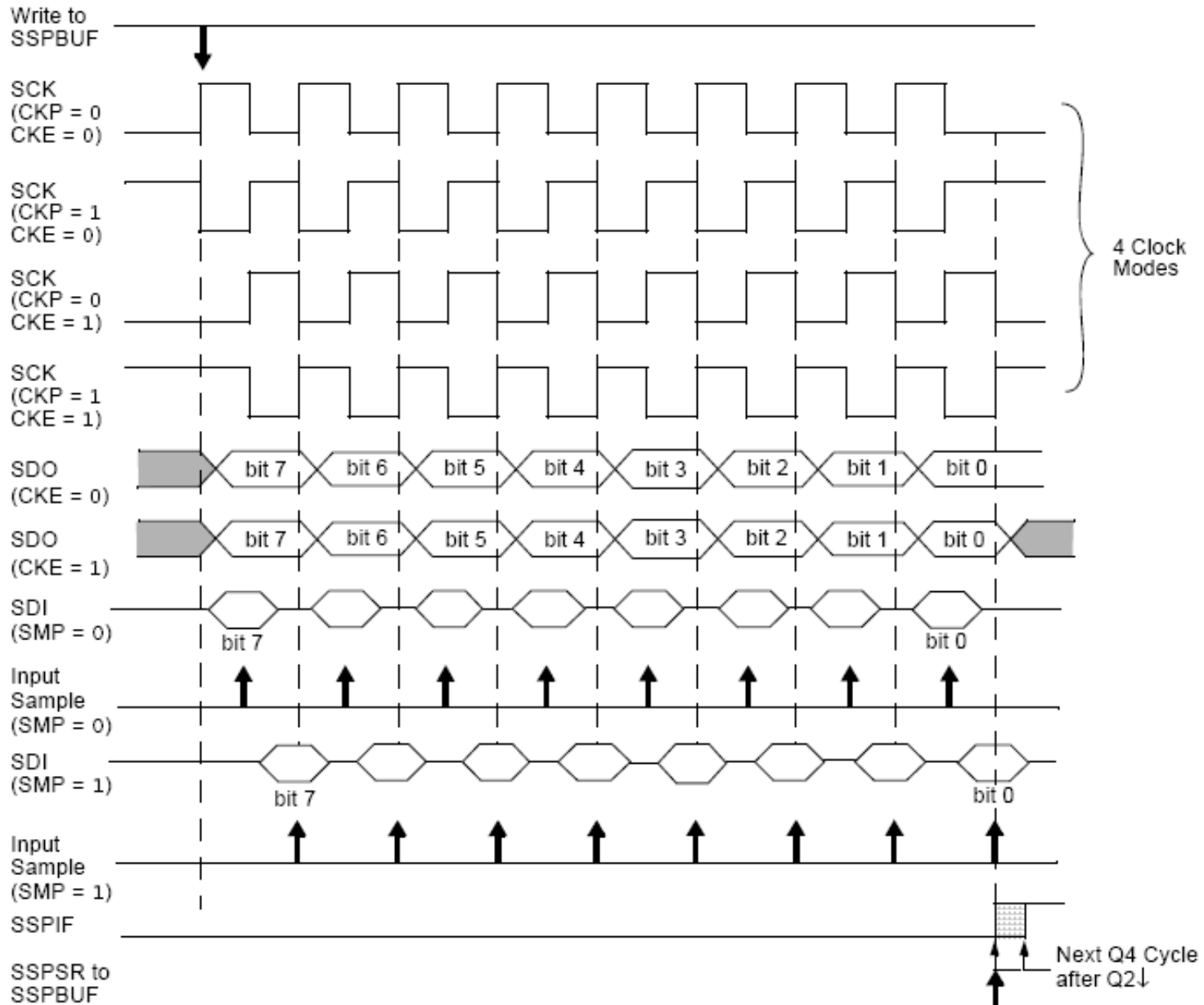
- Clock Edge – Data Bits change on the positive/negative edge of SCK

CKE bit , SSPSTAT<6>

1 = Transmit occurs on transition from active to Idle clock state

0 = Transmit occurs on transition from Idle to active clock state

SPI, PIC18 Peripheral



SPI, C18 Libraries

TABLE 2-10: SINGLE SPI™ PERIPHERAL FUNCTIONS

Function	Description
CloseSPI	Disable the SSP module used for SPI™ communications.
DataRdySPI	Determine if a new value is available from the SPI buffer.
getcSPI	Read a byte from the SPI bus.
getsSPI	Read a string from the SPI bus.
OpenSPI	Initialize the SSP module used for SPI communications.
putcSPI	Write a byte to the SPI bus.
putsSPI	Write a string to the SPI bus.
ReadSPI	Read a byte from the SPI bus.
WriteSPI	Write a byte to the SPI bus.