Introduction to SPI and I2C

Dr. Farid Farahmand
PIC Serial Interface
PIC Out for MSSP
(Master Synchronous Serial Port)
MSSP Module - SPI

• Supports SPI and I2C Protocols
• SPI: Serial Peripheral Interface
  – Applications: Interface to EEPROM, ADC, Sensors, LCD
  – Characteristics: Slower than parallel port, fewer signals
• SPI General Operation:
  – 4- Wire communication (clock, data in, data out, CS)
  – Synchronous → high speed interfaces can be handled
  – Master/Slave configuration – the master chip controls all the slaves and provided clock
  – Uses serial data exchange protocols (MSB goes out first)
  – Data exchange can be on rising or falling edge of the clock
SPI Protocol
Interface

Write/Read Characters into SSPBUFF

SPI Master SSPM3:SSPM0 = 00xb
Serial Input Buffer (SSPBUF)
Shift Register (SSPSR)
Master Control
MSb
LSb
SCK
Serial Clock

SPI Slave SSPM3:SSPM0 = 010xb
Serial Input Buffer (SSPBUF)
Shift Register (SSPSR)
Slave
SS
SCK
Clock
Data In
Data Out
Select
SDI
SDO
SCK
SS
RC0
RC3
RC4
RC5
SPI Protocol Interface

Data is being broadcasted to all chips
SPI Application Example

Registers:
- SSPCON1
- SSPSTAT
- TRISC
SPI Programming Example

• Assume:
  – Fosc = 10 MHz;
  – Data Transfer is 2.5 MHz
  – Data is sampled at the falling edge of the clock
  – Data sent at the rising edge of the clock
  – Idle clock state is HIGH

• What will be the configuration for the Master and Slave Chips

• Write the code to transmit characters
SPI Programming Example (solution)

**Master Mode**

- SSPCON1 Register
- Set to SPI Master Mode $F_{osc}/4 = 2.5 \text{ MHz}$
- SSPCON1 bit.CKP = 1
- SSPCON1 bit.EN = 1
- $\Rightarrow$ **SSPCON1** = **0011 0000**

- SSPSTAT Register
- SSPSTAT bit.SMP = 0; rising edge; idle is high
- SSPSTAT bit.CKE = 1
- SSPSTAT bit.BF = 0
- $\Rightarrow$ **SSPSTAT** = **0100 0000**

- TRISC Register
- TRISCbit.SDO (RC5) = 0
- TRISCbit.SDI (RC4) = 1
- TRISCbit.SCK (RC3) = 0
- TRISCbit.SS (RC0) = 0
- $\Rightarrow$ **TRSC** = **0001 0000**
SPI Programming Example (solution)

**Slave Mode**

- SSPCON1 Register
  - \( \rightarrow \text{SSPCON1} = 0011\ 0100 \)

- SSPSTAT Register (same)
  - \( \rightarrow \text{SSPSTAT} = 0100\ 0000 \)

- TRISC Register
  - TRISCbit.SDO (RC5) = 0
  - TRISCbit.SDI (RC4) = 1
  - TRISCbit.SCK (RC3) = 1
  - TRISCbit.SS (RC0) = 1
  - \( \rightarrow \text{TRSC} = 0001\ 1001 \)
SPI Programming Example (solution)

**Master Mode Program**

- Setup TRISC, SSPCON1, & SSPSTAT
- Set ChiSel=0
- Write a character (0x55) into SSPBUF
- Loop: Check if SSPBUF.bit.BF = 1 (buffer is full)
- IF so: MOVF SSPBUF→WREG
- Set ChipSel = 1 (deactivate the slave chip)
Inter-Integrated Circuit (I2C) Interface

• Created by Philips Inc
• Designed to interface ICs on PCB boards (I2C)
• Characteristics:
  – 2-Wire (SCK & SDA)
  – Synchronous (100 Kbs or 400 Kbps)
  – Master/Slave modes of operation
  – Addressing can be 7 or 10 bit
Pin Out for MSSP
I2C Interface
I2C Interface

- 2-Wire Interface

Registers:
- SSPCON1
- SSPSTAT
- TRISC

- SSPCON2 (port idle or not)
- SSPADD (address)
- SSPSR
- SSPBUF (data read/write)
I2C Programming Example

• Assume:
  – $F_{osc} = 10$ MHz;
  – I2C Master Mode
  – Transmission Rate is 100 KHz
  – Enable slew rate for high speed
  – No error detection

• What will be the configuration for the Master

• Write the code to transmit characters
I2C Programming Example (solution)

Master Mode

- SSPCON1 Register
- Set to SPI Master Mode Fosc/4 = 2.5 MHz
- SSPCON1 bit.CKP = 1
- SSPCON1 bit.EN = 1
- $\rightarrow$ SSPCON1 = 0011 0000

- SSPSTAT Register
- SSPSTAT bit.SMP = 0 ; rising edge; idle is high
- SSPSTAT bit.CKE = 1
- SSPSTAT bit.BF = 0
- $\rightarrow$ SSPSTAT = 0100 0000

- TRISC Register
- TRISCbit.SDO (RC5) = 0
- TRISCbit.SDI (RC4) = 1
- TRISCbit.SCK (RC3) = 0
- TRISCbit.SS (RC0) = 0
- $\rightarrow$ TRSC = 0001 0000
Programing Example

W = SSCON2
If bit0-bit 4 of SSPCON2 = 1 → not in idle mode
Else: Check SSPSTATbit.RW; If 1 → Transmit

W = Character
MOVWF SSPBUF

Check SSPSTATbit.BF = 0
Go to next Character Transmission