

Giao tiếp SPI với vi điều khiển PIC

(Phần 1)

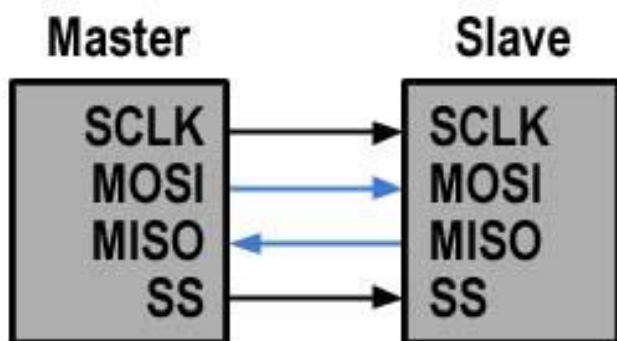


Giao tiếp SPI được hãng motorola giới thiệu ở giữa năm 1980 và được sử dụng trong các dòng vi điều khiển của họ. Ngày nay giao tiếp SPI được sử dụng phổ biến để giao tiếp với các vi điều khiển, eeprom, IC thời gian thực...

SPI là giao thức nối tiếp đồng bộ đa mục đích (general-purpose synchronous serial interface), trong khi thực hiện giao tiếp SPI dữ liệu truyền và nhận được diễn ra một cách đồng thời. Một xung clock đồng bộ dùng để dịch và lấy mẫu thông tin trên hai đường dữ liệu.

Thiết bị giao tiếp SPI sử dụng mối quan hệ master- slaver (chủ-tớ), do thiết bị tớ không được định địa chỉ nên khi giao tiếp một master với nhiều slaver ta cần đường chọn chip CS, để chọn slave nào cần truyền nhận dữ liệu. Vậy số bus cho giao tiếp này là $3 + (1....n)$ dây.

Các cách kết nối cơ bản:

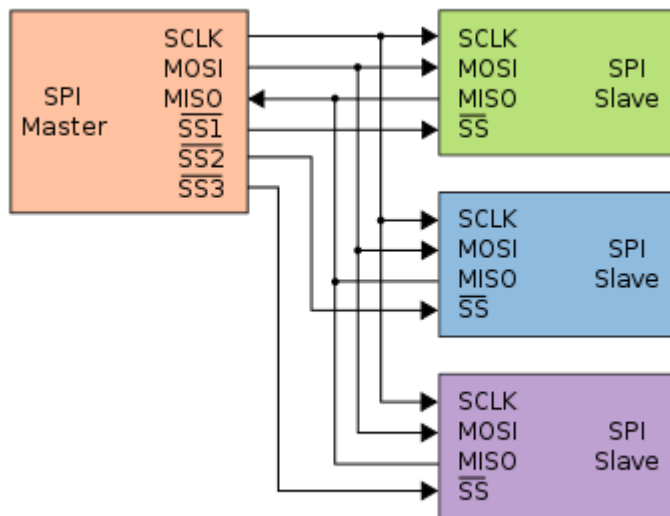


Tên và chức năng của các chân

Trong giao thức chủ-tớ, chỉ có thiết bị Master mới có thể điều khiển (phát ra) xung SCK. Dữ liệu sẽ không được truyền đi nếu như Master không cung cấp xung SCK và tất cả các thiết bị Slave đều được điều khiển bởi xung nhịp phát ra từ Master trong khi đó, Slave lại không có khả năng phát xung.

Các kiểu kết nối trong SPI

a. Kết nối song song nhiều slave:



Kết nối nối tiếp hay kết nối Daisy Chain hay kết nối Cascade

Nguồn: <http://icdesignvn.com/forum/index.php?threads/giao-thuc-truyen-nhan-noi-tiep-spi-serial-peripheral-interface.223/>

Diễn đàn Thiết kế Vi mạch và Vi Điện tử: ICdesignVN.com

Kết nối nối tiếp hay kết nối Daisy Chain hay kết nối Cascade

Nguồn: <http://icdesignvn.com/forum/index.php?threads/giao-thuc-truyen-nhan-noi-tiep-spi-serial-peripheral-interface.223/>

Diễn đàn Thiết kế Vi mạch và Vi Điện tử: ICdesignVN.co

Kết nối nối tiếp hay kết nối Daisy Chain hay kết nối Cascade

Nguồn: <http://icdesignvn.com/forum/index.php?threads/giao-thuc-truyen-nhan-noi-tiep-spi-serial-peripheral-interface.223/>

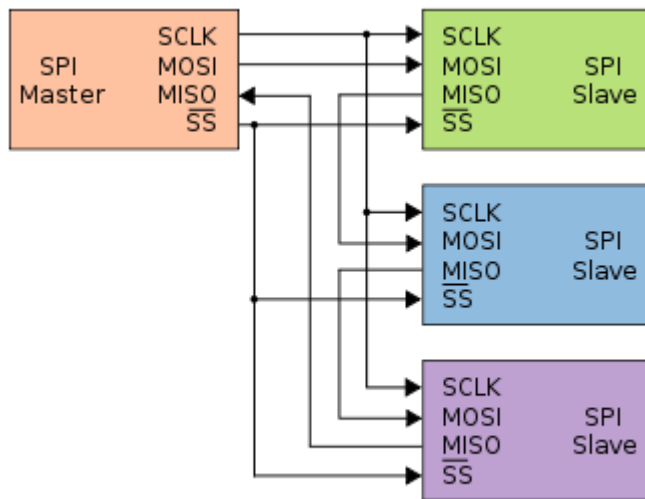
Diễn đàn Thiết kế Vi mạch và Vi Điện tử: ICdesignVN.com

Kết nối nối tiếp hay kết nối Daisy Chain hay kết nối Cascade

Nguồn: <http://icdesignvn.com/forum/index.php?threads/giao-thuc-truyen-nhan-noi-tiep-spi-serial-peripheral-interface.223/>

Diễn đàn Thiết kế Vi mạch và Vi Điện tử: ICdesignVN.com

b. Kết nối Daisy chain

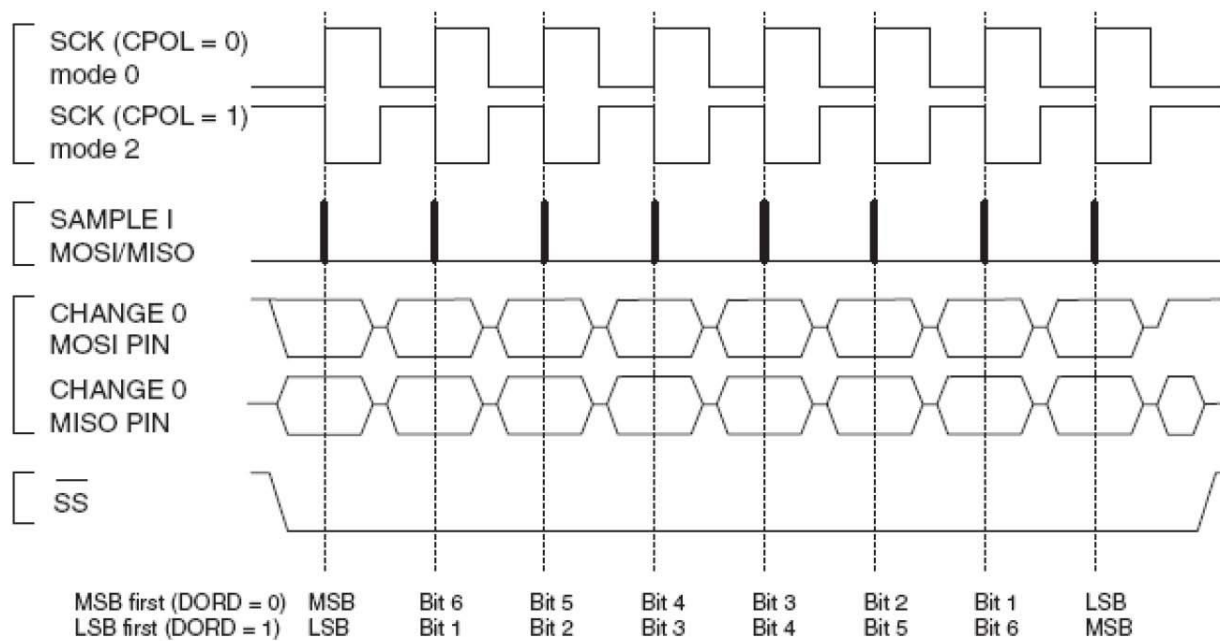


Các mode trong giao tiếp SPI:

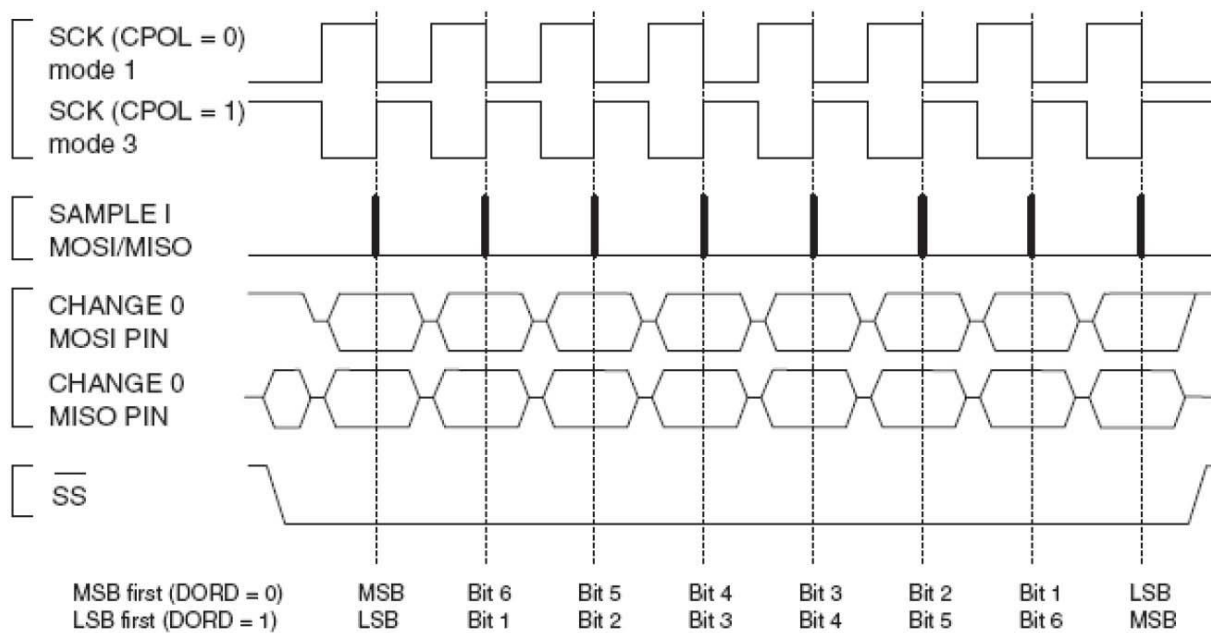
Tùy thuộc vào các cạnh đồng hồ sử dụng cho việc dịch có 4 mode SPI khác nhau:

SPI Mode	SCK (Cạnh bắt đầu)	SCK (cạnh kết thúc)
0	Lấy mẫu dữ liệu tại cạnh lên	Truyền dữ liệu mới tại cạnh xuống
1	Truyền dữ liệu mới tại cạnh lên	Lấy mẫu dữ liệu tại cạnh xuống
2	Lấy mẫu dữ liệu tại cạnh xuống	Truyền dữ liệu mới tại cạnh lên
3	Truyền dữ liệu mới tại cạnh xuống	Lấy mẫu dữ liệu tại cạnh lên

Hình bên dưới là các dạng sóng minh họa:



Dạng sóng của SPI mode 0 và Mode 2



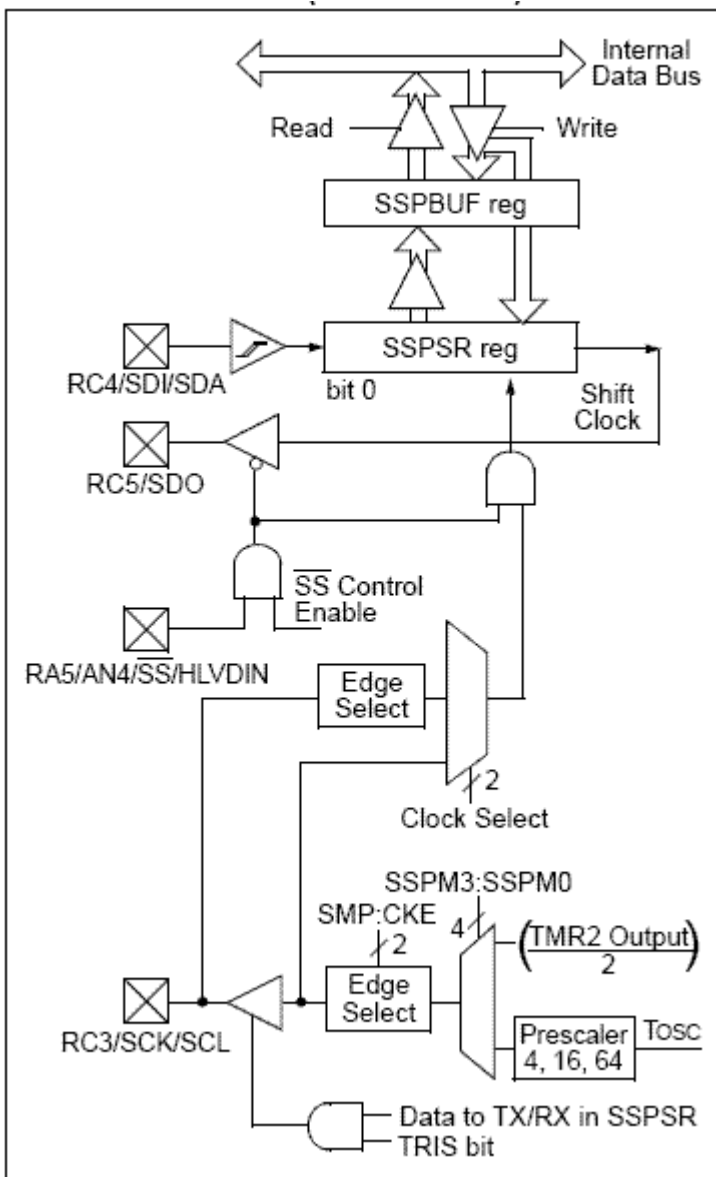
Dạng sóng của SPI mode 1 và Mode 3

SPI Mode trong PIC cho phép 8 bit dữ liệu được truyền nhận đồng bộ 1 cách đồng thời. Hỗ trợ hoạt động ở cả 4 Mode. Để tiến hành hoạt động giao tiếp, về cơ bản, 3 chân sau đây được sử dụng:

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

Ngoài ra, nếu trong hệ thống có nhiều hơn 1 Slave thì chân Chip Select sẽ được dùng:

- + Slave Select – RA5/AN4/**SS**/HLVDIN



Sơ đồ nguyên lý của khối SPI tích hợp trong Vi Điều Khiển PIC

Để sử dụng Mode SPI ta cần phải thiết lập các giá trị chính xác cho các thanh ghi Control. Module MSSP của PIC có 4 thanh ghi dành cho Mode SPI. Đó là:

- + MSSP Control Register 1 (SSPCON1)
- + MSSP Status Register (SSPSTAT)
- + Serial Receive/Transmit Buffer Register (SSPBUF)
- + MSSP Shift Register (SSPSR) – Không truy xuất trực tiếp được

SSPCON1 và SSPSTAT là các thanh ghi điều khiển và cho biết trạng thái hoạt động của Mode SPI. SSPCON1 là thanh ghi có thể đọc hoặc ghi. Trong khi đó, 6 bit thấp của SSPSTAT chỉ có thể đọc, còn 2 bit cao ta có thể đọc hoặc ghi.

SSPSR là thanh ghi có chức năng ghi dịch dữ liệu vào hoặc ra.

SSPBUF là thanh ghi Buffer, có chức năng lưu dữ liệu tạm thời trước khi ghi vào Slave hoặc sau khi đọc về.

Ở hoạt động nhận dữ liệu, SSPSR và SSPBUF kết hợp lại tạo thành bộ nhận dữ liệu được đệm kép. Khi SSPSR nhận 1 byte hoàn chỉnh, nó sẽ được chuyển đến SSPBUF và bit SSPIF được Set lên 1.

Trong suốt quá trình truyền dữ liệu, SSPBUF không được đệm kép. Nếu ta ghi dữ liệu vào SSPBUF thì dữ liệu sẽ được ghi vào cả 2 thanh ghi SSPBUF và SSPSR.

Chi tiết về các bit của các thanh ghi SPI Mode

+ SSPSTAT (MSSP Status Register)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/A	P	S	R/W	UA	BF
bit 7							bit 0

Bit7 SMP: bit Lấy Mẫu

SPI Master Mode:

1 = dữ liệu vào được lấy mẫu tại thời điểm cuối của dữ liệu ra

0 = dữ liệu vào được lấy mẫu tại giữa thời điểm dữ liệu ra

SPI Slave Mode:

Bit SMP phải được xóa khi sử dụng ở Slave Mode

Bit6 CKE: SPI Clock Select bit

1 = quá trình truyền diễn ra khi xung clock chuyển từ trạng thái tác động sang trạng thái nghỉ

0 = quá trình truyền diễn ra khi xung clock chuyển từ trạng thái nghỉ sang trạng thái hoạt động

Bit5 D/A: Data/Address bit, chỉ sử dụng ở Mode I2C

Bit4 P : Stop bitm chỉ sử dụng ở Mode I2C

Bit3 S : Start bit, chỉ sử dụng ở Mode I2C

Bit2 R/W : bit Read/Write, chỉ sử dụng ở Mode I2C

Bit1 UA : Update Address bit, chỉ sử dụng ở Mode I2C

Bit0 BF : Buffer Full Status bit (chỉ sử dụng ở Mode nhận)

1 = Đã nhận đủ dữ liệu, SSPBBUF đầy

0 = Chưa nhận đủ dữ liệu SSPBUF rỗng

+ SSPCON1 (MSSP Control Register)

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

Hình trên cho biết vị trí các bit của thanh ghi SSPCON1

Các bit điều khiển quá trình giao tiếp SPI là :

SSPOV : bit Cờ Tràn, cho biết Synchronous Serial Port bị tràn dữ liệu

SSPEN : viết tắt của Synchronous Serial Port Enable

CKP : điều khiển Clock Polarity

SSPM3:SSPM0 : các bit lựa chọn Mode SPI

Nếu là 0101 : SPI Slave Mode, clock=SCK pin, chân SS bị cấm.

0100 : SPI Slave Mode, clock=SCK pin, chân SS được cho phép.

0011 : SPI Master Mode, clock=TMR2 output/2

0010 : SPI Master Mode, clock= Fosc/64

0001 : SPI Master Mode, clock= Fosc/16

0000 : SPI Master Mode, clock= Fosc/4

HOẠT ĐỘNG CỦA SPI TRONG PIC

Khi khởi tạo giao tiếp SPI, ta phải chỉ rõ là thiết lập chế độ hoạt động nào bằng cách ghi vào thanh ghi SSPCON1 và SSPSTAT các giá trị thích hợp. Giá trị này sẽ cho biết các thông số sau được chọn hay không:

- Master Mode (SCK là ngõ ra xung clock)
- Slave Mode (SCK là ngõ vào xung clock)
- Clock Polarity (trạng thái nghỉ của xung SCK)

- Data Input Sample Phase (tại điểm giữa hoặc cuối của xung clock khi data out)
- Cạnh xung clock (dữ liệu được truyền đi tại cạnh lên/xuống của xung clock)
- Clock Rate (chỉ cho phép ở Master Mode)
- Slave Select Mode (chỉ sử dụng ở Slave Mode)

Như đã giới thiệu, Module MSSP của PIC có 2 thanh ghi chức năng

Truyền/Nhận là Thanh ghi-dịch SSPSR và thanh ghi đệm SSPBUF. SSPSR sẽ dịch chuyển data vào hoặc ra khỏi thiết bị, bit Msb trước. SSPBUF sẽ giữ dữ liệu mà dữ liệu này đã được ghi vào SSPSR cho đến khi nhận được byte dữ liệu kế tiếp. Nếu SSPBUF đã nhận đủ 8 bit dữ liệu thì bit BF (Buffer Full detect bit) và SSPIF (Interrupt Flag bit) của thanh ghi SSPSTAT được set. Bất cứ việc ghi/đọc dữ liệu trong thanh ghi SSPBUF trong quá trình truyền/nhận đều bị cấm và bit WCOL sẽ được set lên 1. Bit này cần được xóa để nếu dữ liệu tiếp theo được ghi vào SSPBUF hoàn tất thành công.

Cần phải đọc dữ liệu trong SSPBUF ngay sau khi nhận về để cho phép các dữ liệu tiếp theo được ghi vào. Bit BF sẽ cho biết khi nào SSPBUF đầy dữ liệu (quá trình truyền hoàn tất). Khi ta đọc giá trị trong thanh ghi SSPBUF, bit BF sẽ được xóa.

Một cách tổng quát, Ngắt của Module MSSP thường dùng để xét xem quá trình truyền/ nhận xong hay chưa. Nếu bit SSPIF = 1 thì chứng tỏ quá trình truyền nhận đã hoàn tất. Ngược lại, nếu bit SSPIF = 0 thì quá trình chưa hoàn thành.

THIẾT LẬP CHÂN VÀO/RA SPI:

Để thiết lập PORT nối tiếp, ta cần phải set bit cho phép SPI hoạt động – bit SSPEN của thanh ghi SSPCON1. Nếu muốn thiết lập lại cấu hình SPI, ta chỉ việc xóa bit SSPEN, khởi tạo lại giá trị cho thanh ghi SSPCON, sau đó lại Set bit SSPEN = 1. Lúc này, các chân của Port nối tiếp là SDI, SDO, SCK, SS. Muốn các chân này có đầy đủ chức năng của Port dữ liệu vào ra ta phải chọn hướng dữ liệu cho từng chân bằng cách thao tác lên thanh ghi TRISx như sau:

- + SDI : chiều dữ liệu tự động được điều khiển bởi Mode SPI
- + SDO : Chân dữ liệu ra nên cần phải xóa bit 5 của TRISC.
- + SCK : Nếu ở chế độ Master thì phải xóa bit 3 của TRISC
- + SCK : Nếu ở chế độ Slave thì bit 3 của TRISC cần được set lên 1
- + SS : Set bit 7 của TRISF

PIC Ở MODE MASTER SPI

Chip Master trong giao tiếp SPI có thể khởi tạo trao đổi dữ liệu bất cứ lúc nào vì nó có thể điều khiển xung SCK. Ở Master Mode, dữ liệu được truyền đi hoặc nhận về ngay khi được ghi vào SSPBUF. Nếu Master chỉ nhận mà không

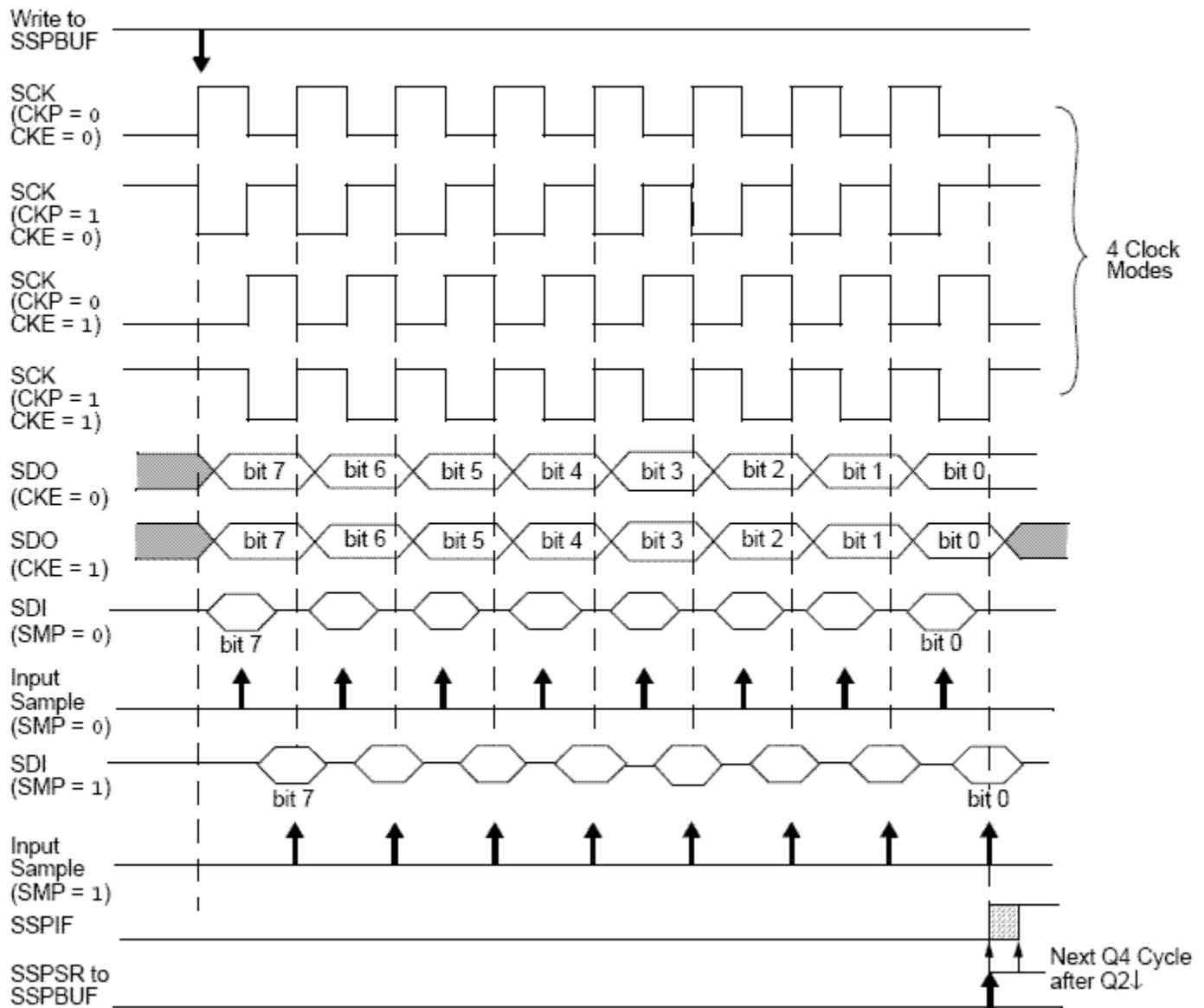
truyền dữ liệu thì chân SDO có thể sử dụng như chân ngõ vào. Thanh ghi SSPSR vẫn tiếp tục dịch dữ liệu bên trong nó đồng bộ với xung SCK. Cứ mỗi Byte nhận được, PIC sẽ lưu vào trong thanh ghi SSPBUF (cờ ngắt và các bit trạng thái có thể được set lên 1).

Cực tính của xung SCK được chọn bằng cách thao tác lên bit CKP (SSPCON1). Dạng sóng của giao tiếp SPI được biểu diễn như hình bên dưới với bit MSB được truyền đi trước. Ở Master Mode, tốc độ xung SCK của giao tiếp SPI được chọn một trong số 4 tùy chọn sau:

- $F_{osc}/4$ (hoặc T_{cy})
- $F_{osc}/16$ (hoặc $4 - T_{cy}$)
- $F_{osc}/64$ (hoặc $16 - T_{cy}$)
- Timer2 output/2

Tốc độ truyền dữ liệu có thể đạt giá trị cao nhất là 10 Megabit/giây (với tần số thạch anh ngoài là 40MHz).

Hình sau cho thấy dạng sóng của SPI Master Mode. Khi bit CKP = 1, dữ liệu trên dây SDO hợp lệ trước mỗi cạnh lên/xuống của xung SCK.



PIC Ở MODE SLAVE SPI

Ở chế độ Slave, dữ liệu chỉ được truyền đi hay nhận về khi có xung SCK. Khi bit cuối cùng đã được chốt, cờ ngắt SSPIF sẽ được set.

Trước khi cho phép PIC hoạt động ở Slave Mode ta cần phải chú ý đến xung SCK. Xung clock phải khớp với trạng thái nghỉ của xung SCK. Trạng thái nghỉ được lựa chọn bằng việc thao tác lên bit CKP.

Trong khi đang ở SPI Slave Mode, xung clock cần thiết cho việc trao đổi dữ liệu được lấy từ chân SCK. PIC có thể truyền/ nhận trong lúc đang 'ngủ đông'. Nếu có 1 byte dữ liệu được nhận thì PIC sẽ 'thức dậy'.

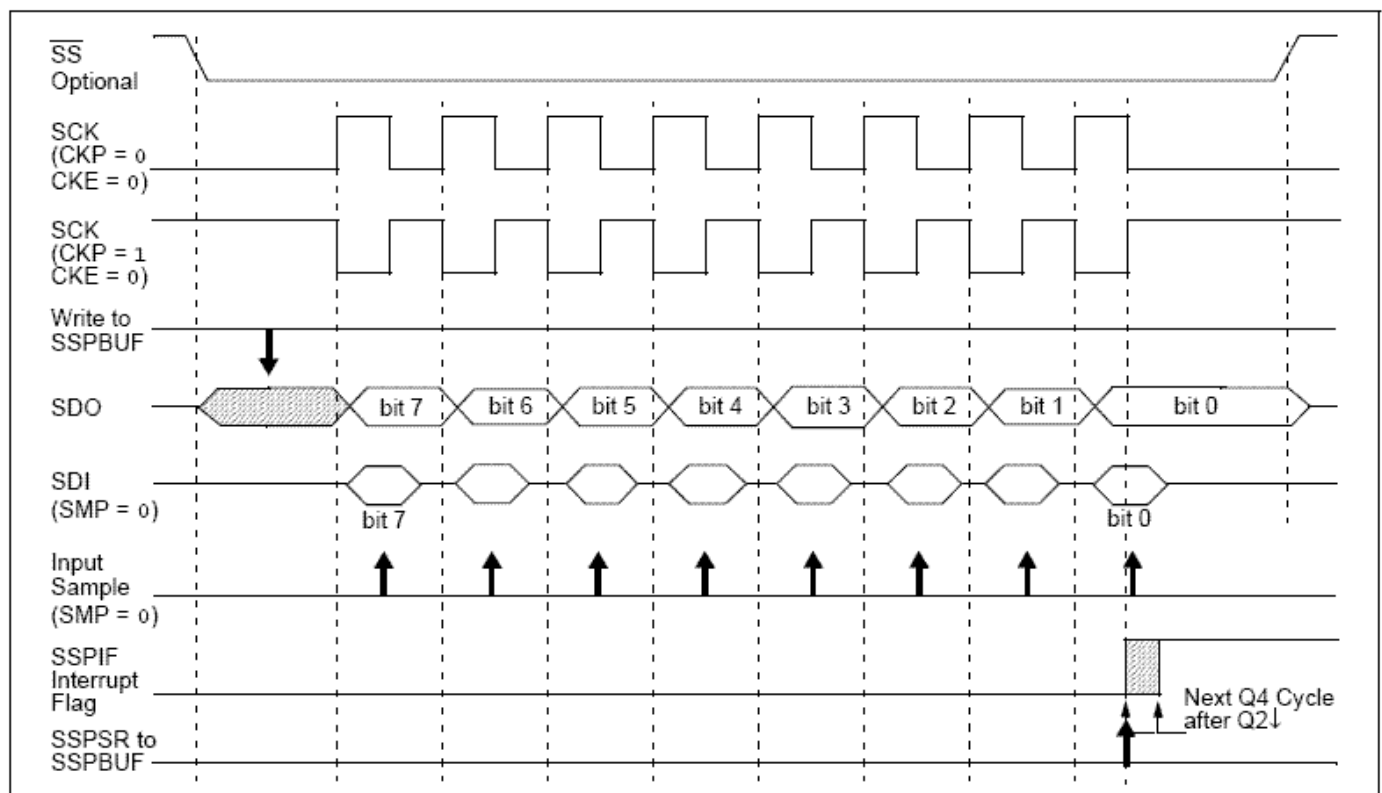
THIẾT BỊ SLAVE TỰ ĐỘNG LỰA CHỌN SỰ ĐỒNG BỘ HÓA

Chân Slave Select (SS) cho phép đồng bộ hóa ở chế độ SPI Slave. Do đó, nếu đặt cấu hình hoạt động ở chế độ Slave thì ta nên cho phép chân SS để điều khiển (đặt các bit SSPCON1 = 0x04). Khi chân SS được đưa xuống mức thấp, sự truyền/ nhận dữ liệu được cho phép và chân SDO sẽ được thúc để tải dữ liệu. Khi chân SS được đưa lên mức cao, chân SDO sẽ không được thúc nữa, dữ liệu sẽ không được lưu thông.

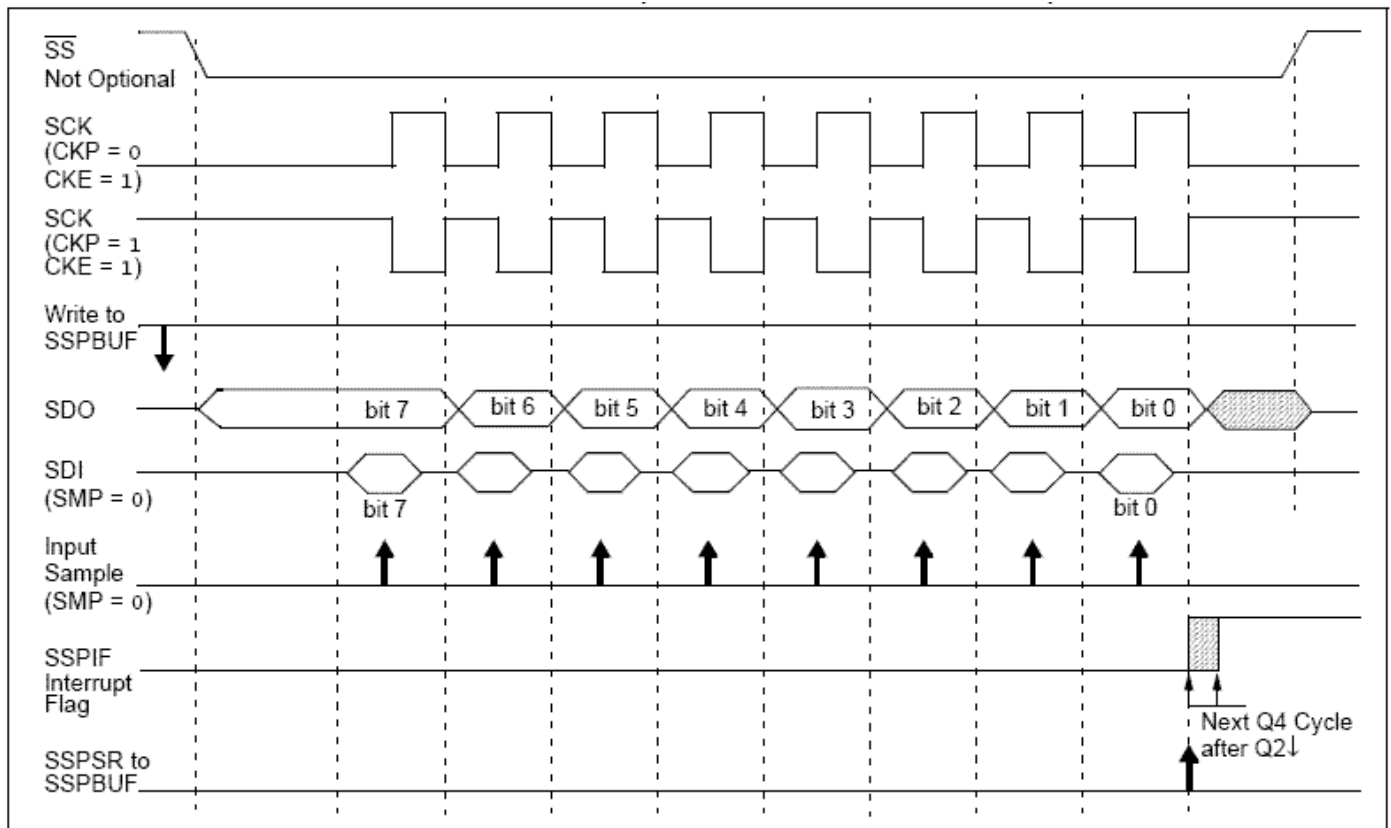
Nếu SPI Slave Mode có chân SS điều khiển được kích hoạt, thì mô-đun SPI sẽ reset khi chân SS được đưa lên Vdd. Lúc này bit counter bị xóa về '0'. Ta cũng có thể xóa bit counter bằng cách xóa bit SSPEN.

Cần phải sử dụng chân SS nếu như SPI Slave Mode được dùng với bit CKE được set.

DẠNG SÓNG CỦA SPI SLAVE MODE (CKE = 0)



DẠNG SÓNG CỦA SPI SLAVE MODE (CKE = 1)



Giao tiếp SPI với vi điều khiển PIC

Phần 3: Thiết lập giao tiếp SPI với CCS



1 / Setup_spi (mode) Setup_spi2 (mode)

_Dùng thiết lập giao tiếp SPI . Hàm thứ 2 dùng với VDK có 2 bộ SPI .

_Trong đó mode có thể là:

- SPI_MASTER , SPI_SLAVE , SPI_SS_DISABLED
- SPI_L_TO_H , SPI_H_TO_L
- SPI_CLK_DIV_4 , SPI_CLK_DIV_16 , SPI_CLK_DIV_64 , SPI_CLK_T2
- SPI_SAMPLE_AT_END, SPI_XMIT_L_TO_H

_Nhóm 1 xác định VDK là master hay slave ,slave select

_Nhóm 2 xác định clock cạnh lên hay xuống .

_Nhóm 3 xác định tần số xung clock , SPI_CLK_DIV_4 nghĩa là tần số = FOSC / 4 , tương ứng 1 chu kỳ lệnh / xung .

_Các nhóm này có thể kết hợp với nhau bởi dấu |

_Hàm không trả về trị .

_Ngoài ra ,tuỳ VDK mà có thêm 1 số tham số khác , xem file * .h .

Cách thiết lập các mode trong giao tiếp SPI trong CCS



```
#define SPI_MODE_0_0    (SPI_L_TO_H | SPI_XMIT_L_TO_H)
//SPI Mode 0
#define SPI_MODE_0_1    (SPI_L_TO_H)
//SPI Mode 1
#define SPI_MODE_1_0    (SPI_H_TO_L)
//SPI Mode 2
#define SPI_MODE_1_1    (SPI_H_TO_L | SPI_XMIT_L_TO_H)
//SPI Mode 3
```

Khi sử dụng hàm setup_spi ta chỉ đơn giản gọi ra như sau:

```
setup_spi(SPI_MASTER | SPI_MODE_0_0 | SPI_CLK_DIV_4);
```

2 / Spi_read (data) Spi_read2 (data)

_data có thể có thêm và là số 8 bit . Hàm thứ 2 cho bộ SPI thứ 2 .

_Hàm trả về giá trị 8 bit value = spi_read ()

_Hàm trả về giá trị đọc bởi SPI . Nếu value phù hợp SPI_read () thì data sẽ được phát xung ngoài

và data nhận được sẽ được trả về . Nếu không có data sẵn sàng , spi_read () sẽ đợi data .

_Hàm chỉ dùng cho SPI hardware (SPI phần cứng) .

3 / spi_write (value) Spi_write2 (value)

_Hàm không trả về trị . value là giá trị 8 bit .

- _Hàm này gửi value (1 byte) tới SPI , đồng thời tạo 8 xung clock .
- _ Hàm chỉ dùng cho SPI hardware (SPI phần cứng) .

4 / spi_data_is_in () Spi_data_is_in2 ()

_Hàm trả về TRUE (1) nếu data nhận được đầy đủ (8 bit) từ SPI , trả về false nếu chưa nhận đủ .

_Hàm này dùng kiểm tra xem giá trị nhận về SPI đã đủ 1 byte chưa để dùng hàm spi_read () đọc data vào biến .