



Chương 9 - Giao tiếp nổi tiếp



1. Đặc điểm giao tiếp nổi tiếp.
2. SPI-MSSP.
3. USART.

Đặc điểm giao tiếp nối tiếp

- ❖ PIC 18F8722 có các khả năng giao tiếp nối tiếp sau :
 - Khối MSSP (Master Synchronous Serial Port) cung cấp khả năng giao tiếp với các vi điều khiển khác và các vi mạch ngoại vi (EEPROM, thanh ghi dịch, vi mạch lái màn hình, vi mạch chuyển đổi AD. . .) theo hai phương pháp :
 - SPI (Serial Peripheral Interface).
 - I²CTM (Inter-Integrated Circuit).
 - Khối USART (Universal Synchronous Asynchronous Receiver Transmitter) cho phép hoạt động theo các chế độ :
 - Truyền nhận bất đồng bộ (song công).
 - Truyền nhận đồng bộ chủ (bán song công).
 - Truyền nhận đồng bộ tớ (bán song công).

Chế độ SPI của MSSP

- ❖ Chế độ SPI (Serial Peripheral Interface) cho phép truyền nhận đồng thời dữ liệu đồng bộ 8 bit.
- ❖ Các tín hiệu sử dụng trong chế độ này :
 - SDO - Dữ liệu truyền nối tiếp.
 - SDI - Dữ liệu nhận nối tiếp.
 - SCK - Xung clock
 - \overline{SS} - Chọn slave (chỉ dùng khi chọn chế độ slave).

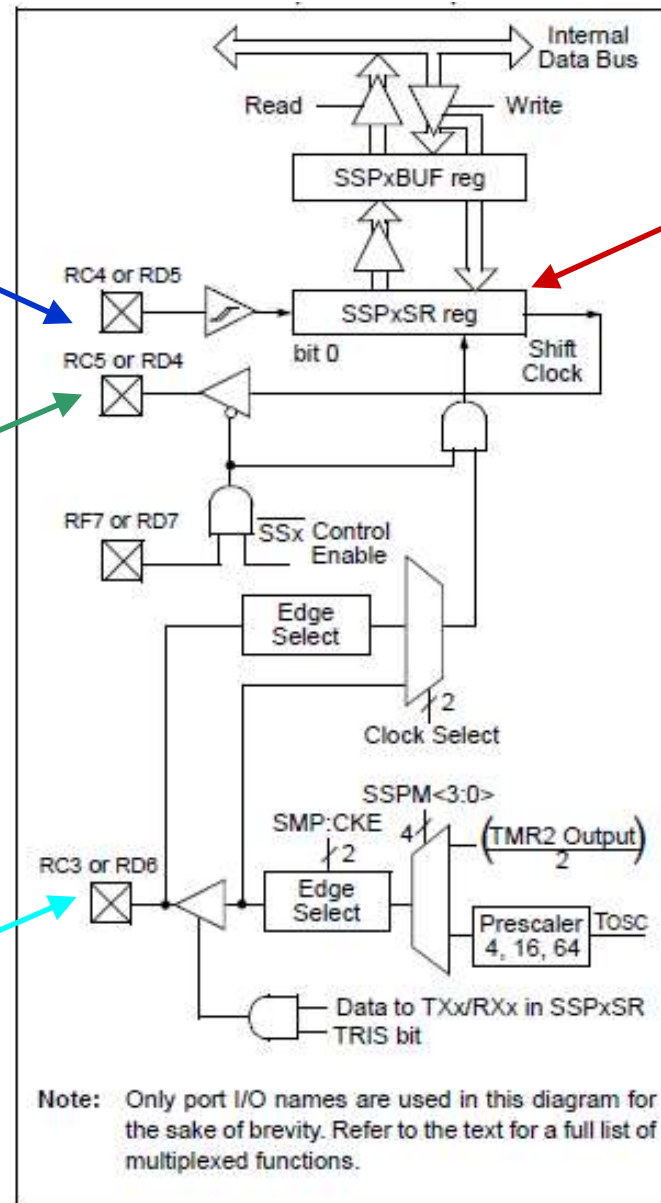
Sơ đồ khối chế độ SPI

Dữ liệu vào

Dữ liệu ra

Xung clock

thanh ghi dịch



Thanh ghi SSPxSTAT

SMP	CKE	D / \bar{A}	P	S	R / \bar{W}	UA	PS
-----	-----	---------------	---	---	---------------	----	----

- ❖ (bit 7) SMP : lấy mẫu bit
 - Trong chế độ SPI chủ :
 - 1= lấy mẫu dữ liệu ở cuối bit
 - 0= lấy mẫu dữ liệu ở giữa bit
 - Trong chế độ SPI tớ : phải được xóa về 0.
- ❖ (bit 6) CKE : Chọn cạnh tác động của xung clock truyền
 - Khi CKP=0:
 - 1= truyền dữ liệu theo cạnh lên
 - 0= truyền dữ liệu theo cạnh xuống
 - Khi CKP=1
 - 1= truyền dữ liệu theo cạnh xuống
 - 0= truyền dữ liệu theo cạnh lên

Thanh ghi SSPxSTAT (tt.)

- ❖ (bit 5) D / \overline{A} : chọn dữ liệu / địa chỉ (chỉ trong chế độ I²C)
- ❖ (bit 4) P : Stop bit (chỉ dùng với chế độ I²C)
- ❖ (bit 3) S : Start bit (chỉ dùng với chế độ I²C)
- ❖ (bit 2) R / \overline{W} : chọn hoạt động đọc / ghi (chỉ dùng với I²C)
- ❖ (bit 1) UA : update address bit (chỉ dùng với I²C)
- ❖ (bit 0) BF : bit trạng thái báo đệm đầy (Buffer Full status bit)
 - Trong chế độ truyền :
 - 1= dữ liệu đang truyền, SSPBUF đầy.
 - 0= dữ liệu đã truyền xong, SSPBUF rỗng.
 - Trong chế độ nhận :
 - 1= đã nhận xong, SSPBUF đầy.
 - 0= chưa nhận xong, SSPBUF rỗng.

Thanh ghi SSPxCON1

WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
------	-------	-------	-----	-------	-------	-------	-------

- ❖ (bit 7) WCOL : phát hiện lỗi truyền
 - =1 : xuất dữ liệu ra SSPBUF khi chưa truyền xong dữ liệu trước
 - =0 : không bị lỗi.
- ❖ (bit 6) SSPOV : lỗi nhận tràn dữ liệu (nhận dữ liệu mới trong khi SSPBUF vẫn còn giữ dữ liệu nhận trước đó.
- ❖ (bit 5) SSPEN : cho phép dùng giao tiếp MSSP.
 - 1= cho phép.
 - 0= cấm
- ❖ (bit4) CKP : chọn cực cho xung clock.
 - 1= chọn mức cao làm trạng thái tắt của clock.
 - 0= chọn mức thấp làm trạng thái tắt của clock.

Thanh ghi SSPxCON1 (tt.)

- ❖ (bit 3 - 0) SSPM3 - SSPM0 : chọn chế độ hoạt động
 - 0101 : SPI tớ, clock=SCK, không dùng SS.
 - 0100 : SPI tớ, clock=SCK, dùng SS.
 - 0011 : SPI chủ, clock=(đầu ra Timer2) / 2
 - 0010 : SPI chủ, clock=FOSC/64
 - 0001 : SPI chủ, clock=FOSC/16
 - 0000 : SPI chủ, clock=FOSC/4

Ví dụ truyền nhận dùng polling

- ❖ Lặp vòng chờ nhận dữ liệu

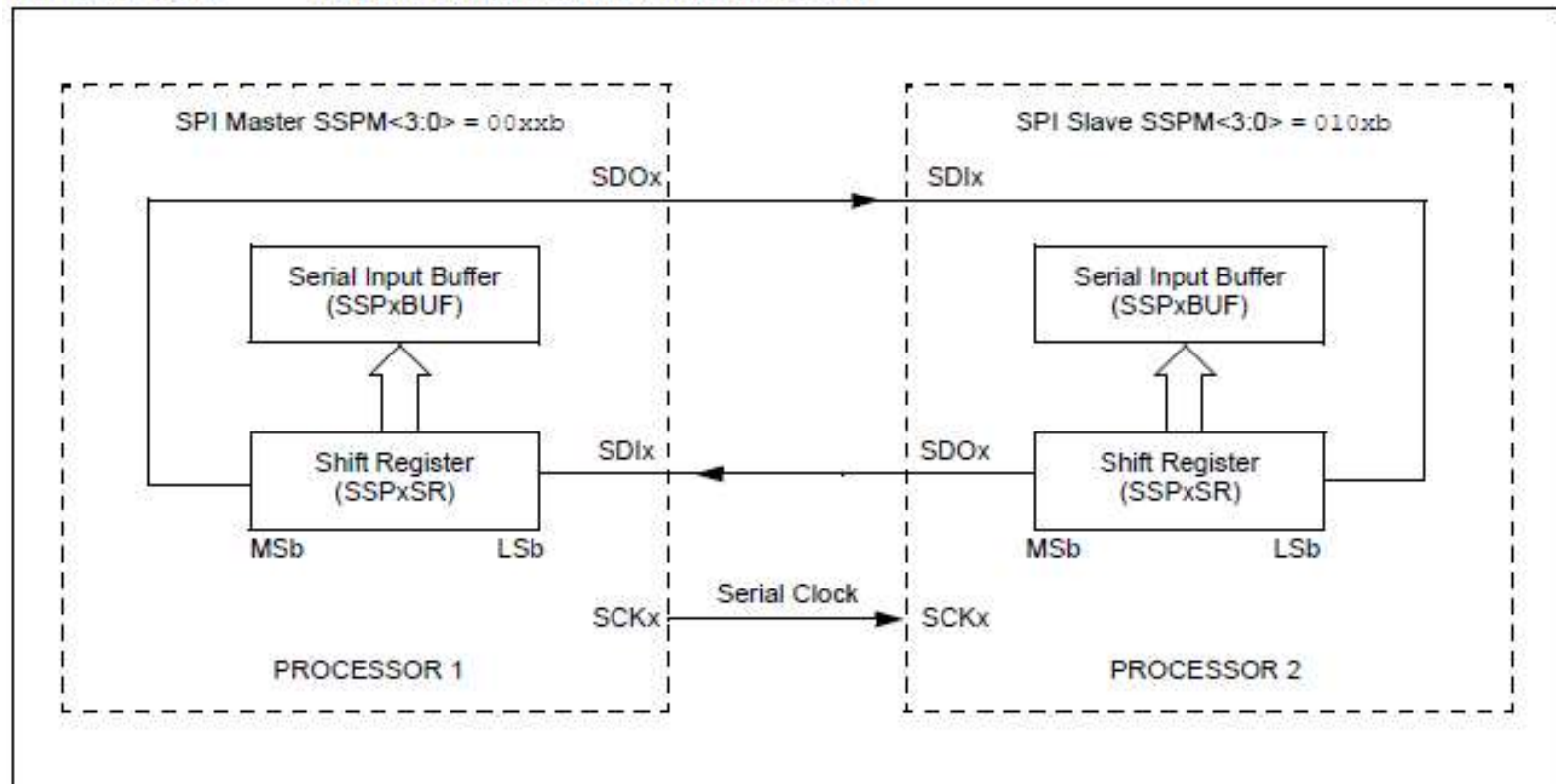
```
while (SSP1STATbits.BF==0) ; //Buffer Full bit=1?  
rxdata=SSP1BUF;           //yes
```

- ❖ Truyền dữ liệu

```
SSP1BUF=txdata;
```

Hoạt động chủ/tổ

- ❖ Chế độ SPI chủ / tớ dùng đối với 2 MCU:
 - MCU chủ xuất xung SCK, MCU tớ nhận SCK.
 - Hai MCU phải chọn cực của xung clock (CKP) như nhau.
 - Cả hai MCU có thể truyền nhận cùng lúc.

FIGURE 19-2: SPI MASTER/SLAVE CONNECTION

Lập trình - Cấu hình

- ❖ Cho phép chế độ SSP (SSPEN=1).
- ❖ Muốn lập trình lại chế độ SSP ta làm theo trình tự sau :
 - Xóa SSPEN - lập trình SSPCON - lập SSPEN
- ❖ Cấu hình các bit SDI, SDO, SCK, \overline{SS} như sau :
 - SDI : lập bit TRISC₄
 - SDO : xóa bit TRISC₅
 - SCK (chủ) : xóa bit TRISC₃
 - SCK (tớ) : lập bit TRISC₃
 - \overline{SS} : lập bit TRISA₅
- ❖ Trong chế độ SPI tớ, \overline{SS} hỗ trợ sử dụng một chủ - nhiều tớ.
- ❖ Kết hợp với chế độ ngủ, dùng để đánh thức MCU bằng ngắt quãng MSSP.

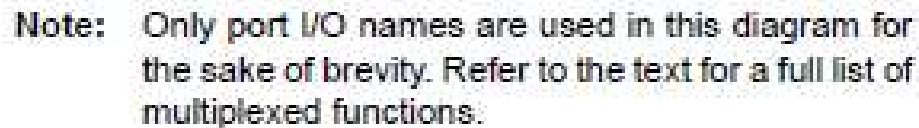
The diagram illustrates the timing for the Q4 cycle of an SPI transaction. It shows the relationship between the following signals:

- Write to SSPxBUF:** A single pulse at the start of the cycle.
- SCKx (CKP = 0, CKE = 0):** Master clock, high for the first half of each bit period.
- SCKx (CKP = 1, CKE = 0):** Master clock, low for the first half of each bit period.
- SCKx (CKP = 0, CKE = 1):** Slave clock, high for the first half of each bit period.
- SCKx (CKP = 1, CKE = 1):** Slave clock, low for the first half of each bit period.
- SDOx (CKE = 0):** Master data output, driving bits 7 down to 0.
- SDOx (CKE = 1):** Slave data output, driving bits 7 down to 0.
- SDIx (SMP = 0):** Master data input, sampling bits 7 down to 0.
- Input Sample (SMP = 0):** Sampling clock for the master, rising at the midpoint of each bit period.
- SDIx (SMP = 1):** Slave data input, sampling bits 7 down to 0.
- Input Sample (SMP = 1):** Sampling clock for the slave, rising at the midpoint of each bit period.
- SSPxIF:** Interrupt flag, which becomes active at the end of the Q4 cycle.
- SSPxSR to SSPxBUF:** Shift register to buffer transfer, which becomes active at the end of the Q4 cycle.

A bracket on the right indicates that the four SCKx waveforms represent "4 Clock Modes". A note at the bottom right indicates the "Next Q4 Cycle after Q2↓".

Chế độ I²C của MSSP

- ❖ Chế độ I²C (Inter-IC) của PIC có thể chọn chủ/tớ.
- ❖ Các tín hiệu sử dụng trong chế độ này :
 - Dữ liệu nối tiếp - RC4/SDI/**SDA**
 - Xung clock - RC3/SCK/**SCL**
- ❖ Các thanh ghi dùng trong chế độ I²C :
 - SSPxCON1 - điều khiển SSP.
 - SSPxCON2
 - SSPxSTAT - trạng thái SSP.
 - SSPxBUF - đệm truyền nhận.
 - SSPxADD - giữ địa chỉ nhận dạng của IC tớ.
 - SSPxSR - thanh ghi dịch (không thể tác động trực tiếp).
- ❖ Các nội dung khác về I²C tham khảo tài liệu.



Giao tiếp nối tiếp EUSART - đặc điểm

- ❖ Hai khối EUSART1,2 cung cấp các chế độ giao tiếp nối tiếp:
 - Bất đồng bộ (Asynchronous), song công (full duplex):
 - Tự đánh thức khi nhận ký tự.
 - Tự chỉnh tốc độ baud.
 - Truyền được ký tự BREAK 12-bits.
 - Đồng bộ chủ (Synchronous - Master), bán song công (half duplex). Có thể chọn mức cho xung clock.
 - Đồng bộ tớ (Synchronous - Slave), bán song công. Có thể chọn mức cho xung clock.

Giao tiếp EUSART - đặc điểm(tt.)

❖ Các chân sử dụng :

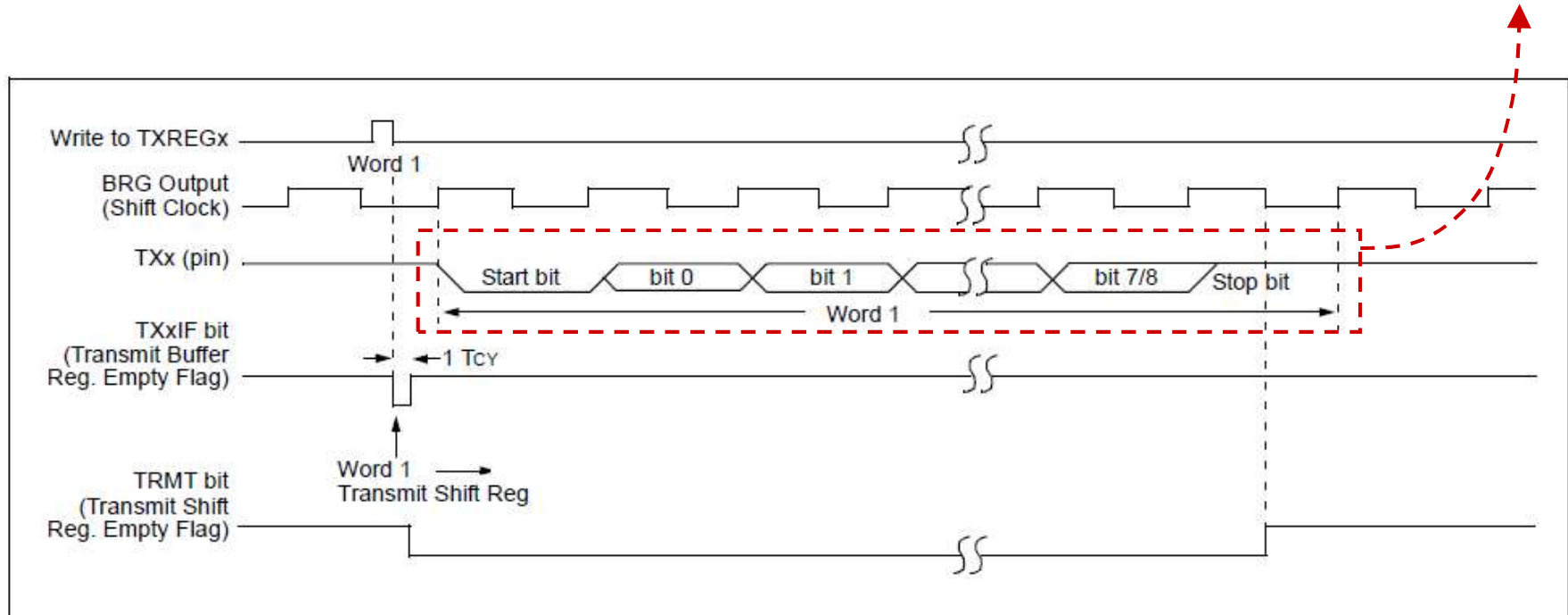
- RC6/**TX1**/CK1, RG1/**TX2**/CK2 : là ngõ truyền dữ liệu nối tiếp.
- RC7/**RX1**/DT1, RG2/**RX2**/DT2: là ngõ nhập dữ liệu nối tiếp.

❖ Các thanh ghi sử dụng :

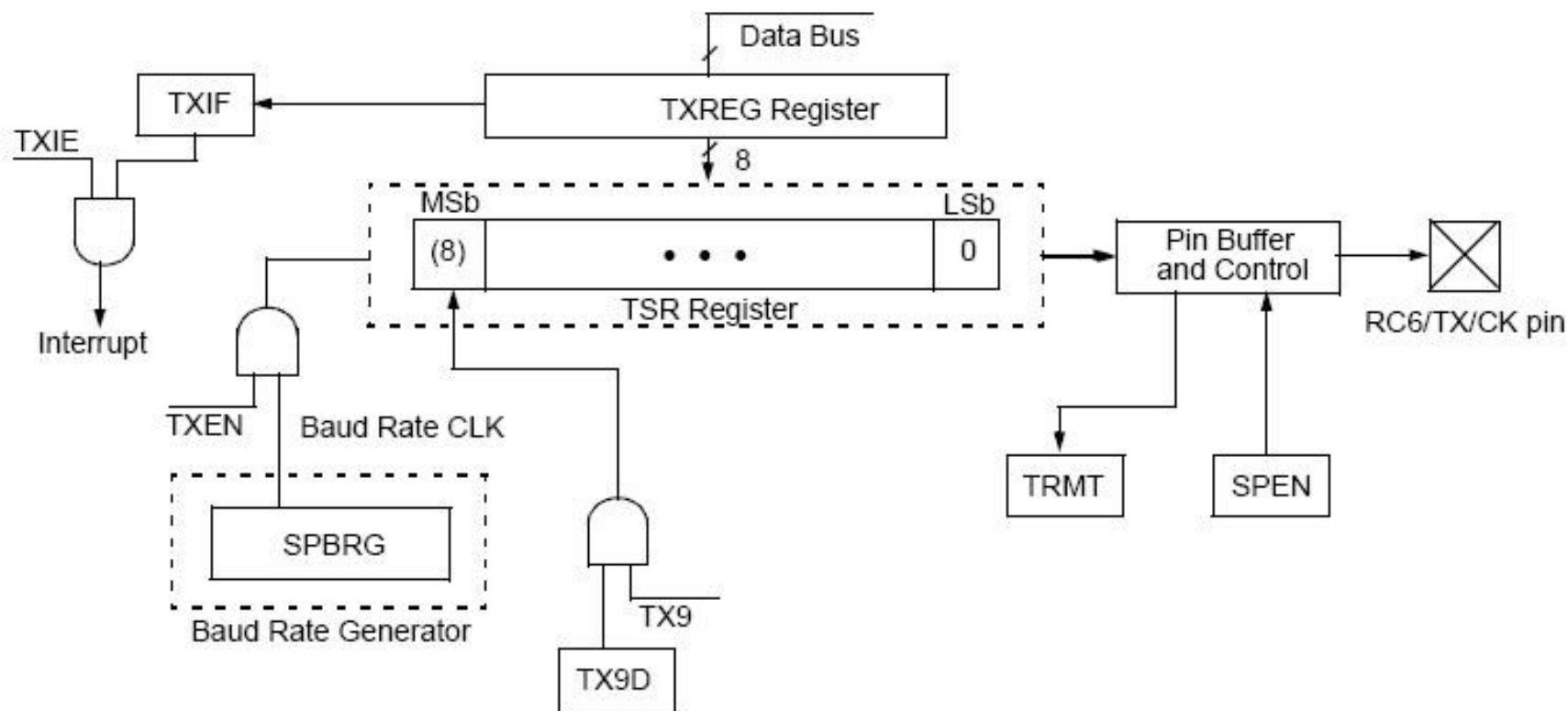
- TXSTAx, RCSTAx : điều khiển/trạng thái truyền/nhận.
- BAUDCONx: điều khiển chọn chế độ cho xung truyền /nhận.
- TXREGx, RCREGx: đệm dữ liệu truyền/nhận.

Chế độ bất đồng bộ

- ❖ Dữ liệu truyền/nhận theo khung dữ liệu.
- ❖ Thực hiện đồng bộ theo từng khung dữ liệu (thông qua cạnh xuống và mức 0 của start bit).
- ❖ Khung dữ liệu : 1 start bit + 8(hoặc 9) data bits + 1 stop bit



Sơ đồ khối truyền bất đồng bộ



Thanh ghi TXSTAx

CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D
------	-----	------	------	---	------	------	------

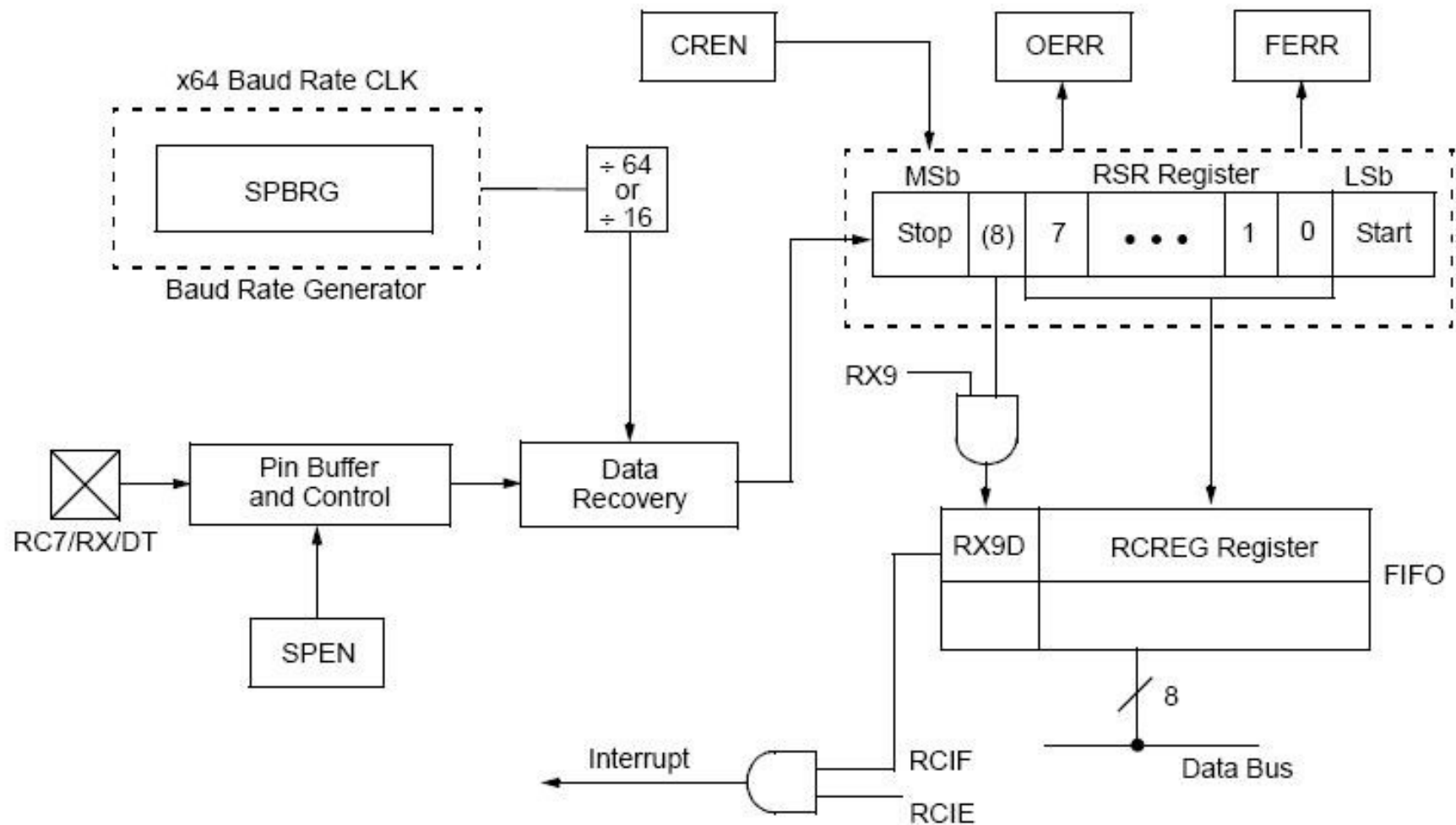
- ❖ CSRC : chọn nguồn xung clock (chỉ dùng cho chế độ đồng bộ)
 - 1= chế độ chủ (tạo xung clock từ bộ phát xung BRG).
 - 0= lấy mẫu dữ liệu ở giữa bit
- ❖ TX9 : Chọn khung truyền 8 bit hay 9 bit
 - 1= khung 9 bit (1-8-1-1)
 - 0= khung 8 bit (1-8-1)
- ❖ TXEN : cho phép truyền
 - 1= cho phép
 - 0= cấm
- ❖ SYNC : chọn chế độ đồng bộ hay bất đồng bộ
 - 1= chế độ đồng bộ
 - 0= chế độ bất đồng bộ

Thanh ghi TXSTAx (tt.)

CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D
------	-----	------	------	---	------	------	------

- ❖ BRGH : chọn tốc độ truyền nhận cao hay thấp (chỉ dùng cho chế độ bất đồng bộ)
 - 1= tốc độ cao
 - 0= tốc độ thấp
- ❖ TRMT : trạng thái thanh ghi dịch truyền (TSR)
 - 1= TSR rỗng
 - 0= TSR chưa rỗng
- ❖ TX9D : bit dữ liệu truyền thứ 9 (dùng cho chế độ 9 bit), có thể là bit phân biệt dữ liệu/địa chỉ hay bit kiểm tra chẵn lẻ.

Sơ đồ khối nhận bất đồng bộ



Thanh ghi RCSTAx

SPEN

RX9

SREN

CREN

ADDEN

FERR

OERR

RX9D

- ❖ SPEN : cho phép cổng nối tiếp
 - 1= cho phép cổng giao tiếp nối tiếp.
 - 0= cấm (mặc định sau Reset).
- ❖ RX9 : Chọn khung nhận (1= khung 9 bit, 0= khung 8 bit)
- ❖ SREN : cho phép nhận đơn (chế độ đồng bộ chủ)
 - 1= cho phép nhận 1 ký tự
 - 0= cấm
- ❖ CREN : cho phép nhận liên tục
 - Chế độ bất đồng bộ
 - 1= cho phép bộ nhận
 - 0= cấm
 - Chế độ đồng bộ
 - 1= cho phép nhận liên tục
 - 0= cấm

Thanh ghi RCSTAx (tt.)

SPEN

RX9

SREN

CREN

ADDEN

FERR

OERR

RX9D

- ❖ **ADDEN** : cho phép phát hiện địa chỉ (chỉ dùng trong chế độ đồng bộ 9 bit)
 - 1= cho phép phát hiện địa chỉ, cho phép ngắt quãng và đọc đệm nhận khi bit 8 của thanh ghi dịch bộ nhận RSRx =1.
 - 0= cấm (mặc định sau Reset).
- ❖ **FERR** : lỗi sai về khung dữ liệu
 - 1= lỗi (thay đổi khi thực hiện đọc RCREGx hoặc nhận ký tự mới hợp lệ)
 - 0= không có lỗi
- ❖ **OERR** : lỗi tràn ký tự
 - 1= lỗi (bị xóa khi thực hiện xóa bit CREN)
 - 0= không có lỗi
- ❖ **RX9D** : bit dữ liệu nhận thứ 9.

Thanh ghi BAUDCONx

ABDOVF	RCIDL	-	SCKP	BRG16	-	WUE	ABDEN
--------	-------	---	------	-------	---	-----	-------

- ❖ ABDOVF : cho phép phát hiện địa chỉ (chỉ dùng trong chế độ đồng bộ 9 bit)
 - 1= có xoay vòng phát hiện tốc độ nhận.
 - 0= không có xoay vòng tốc độ.
- ❖ RCIDL : trạng thái hoạt động nhận
 - 1= đang không nhận dữ liệu.
 - 0= đang nhận dữ liệu.
- ❖ SCKP : chọn mức xung clock đồng bộ.
 - 1= mức nghỉ là mức cao.
 - 0= mức nghỉ là mức thấp.

Thanh ghi BAUDCONx

ABDOVF	RCIDL	-	SCKP	BRG16	-	WUE	ABDEN
--------	-------	---	------	-------	---	-----	-------

- ❖ BRG16 : dùng thanh ghi BRG 16 bit hay 8 bit
 - 1= 16 bit, SPBRGHx và SPBRGx.
 - 0= bit, chỉ dùng SPBRGx.
- ❖ WUE : cho phép chế độ đánh thức (chỉ chế độ bất đồng bộ).
 - 1= EUSART liên tục kiểm tra ngõ RXx, phát ra ngắt quãng khi phát hiện cạnh xuống (bit này tự động bị xóa khi có cạnh lên xung RXx).
 - 0= không sử dụng.
- ❖ ABDEN : tự động phát hiện tốc độ truyền/nhận (chỉ chế độ bất đồng bộ).
 - 1= đo tốc độ nhận với ký tự Sync (55h).
 - 0= không sử dụng.

Bộ phát xung truyền nhận BRG

- ❖ Xung clock dùng trong giao tiếp nối tiếp được tạo ra bởi bộ tạo xung BRG (Baud rate generator).
- ❖ Tốc độ Baud được tính theo công thức trong bảng sau :

Configuration Bits			BRG/EUSART Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	$F_{osc}/[64 (n + 1)]$
0	0	1	8-bit/Asynchronous	
0	1	0	16-bit/Asynchronous	$F_{osc}/[16 (n + 1)]$
0	1	1	16-bit/Asynchronous	
1	0	x	8-bit/Synchronous	$F_{osc}/[4 (n + 1)]$
1	1	x	16-bit/Synchronous	

$n = \text{SPBRGHx:SPBRGx (16 bit)}$

$\text{BRG16} = \text{BAUDCONxbits.BRG16}$

$\text{BRGH} = \text{TXSTAxbits.BRGH}$

Bộ phát xung truyền nhận BRG

- ❖ Ví dụ : nếu chọn chế độ bất đồng bộ, tốc độ là 9600 bauds, tần số xung dao động $F_{osc}=16\text{ MHz}$, 8 bit BRG thì số đếm X là
- $X = ((F_{osc}/\text{tốc độ})/64) - 1 = ((16000000/9600)/64) - 1 = 25.042$
 - Lấy tròn số: $X = 25$
 - Tính ngược lại tốc độ:
 $\text{Tốc độ} = 16000000/(64(25+1)) = 9615.$
 - Như vậy, sai số giữa tốc độ thực và tốc độ mong muốn là:
 $\text{Sai số} = (9615-9600)/9600 = 0.16\%$

Bảng số đếm tham khảo 1

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1.201	-0.16	103
2.4	2.441	1.73	255	2.404	0.16	129	2.404	0.16	64	2.403	-0.16	51
9.6	9.615	0.16	64	9.766	1.73	31	9.766	1.73	15	9.615	-0.16	12
19.2	19.531	1.73	31	19.531	1.73	15	19.531	1.73	7	—	—	—
57.6	56.818	-1.36	10	62.500	8.51	4	52.083	-9.58	2	—	—	—
115.2	125.000	8.51	4	104.167	-9.58	2	78.125	-32.18	1	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.16	207	0.300	-0.16	103	0.300	-0.16	51
1.2	1.202	0.16	51	1.201	-0.16	25	1.201	-0.16	12
2.4	2.404	0.16	25	2.403	-0.16	12	—	—	—
9.6	8.929	-6.99	6	—	—	—	—	—	—
19.2	20.833	8.51	2	—	—	—	—	—	—
57.6	62.500	8.51	0	—	—	—	—	—	—
115.2	62.500	-45.75	0	—	—	—	—	—	—

Bảng số đếm tham khảo 2

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0											
	FOSC = 40.000 MHz			FOSC = 20.000 MHz			FOSC = 10.000 MHz			FOSC = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	2.441	1.73	255	2.403	-0.16	207
9.6	9.766	1.73	255	9.615	0.16	129	9.615	0.16	64	9.615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19.230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55.555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 0								
	FOSC = 4.000 MHz			FOSC = 2.000 MHz			FOSC = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	0.300	-0.16	207
1.2	1.202	0.16	207	1.201	-0.16	103	1.201	-0.16	51
2.4	2.404	0.16	103	2.403	-0.16	51	2.403	-0.16	25
9.6	9.615	0.16	25	9.615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

Khởi động truyền bất đồng bộ

- ❖ Khởi động số đếm SPBRGHx:SPBRGx tùy theo tốc độ. Chọn bộ đếm 8 bit hay 16 bit
 - TXSTAxbits.BRGH và
 - BAUDCONxbits.BRG16
- ❖ Chọn chế độ bất đồng bộ:
 - TXSTAxbits.SYNC=0 và
 - RCSTAxbits.SPEN=1

Khởi động truyền bất đồng bộ (tt.)

❖ Nếu dùng ngắt quãng:

- $\text{PIE1bits.TX1IE}=1$, chọn IPR1bits.TX1IP hoặc
- $\text{PIE3bits.TX2IE}=1$, chọn IPR3bits.TX2IP
- $\text{INTCONbits.GIE}=1$ và
- $\text{INTCONbits.PEIE}=1$

❖ Nếu dùng 9 bit dữ liệu:

- $\text{TXSTAxbits.TX9}=1$.

❖ Cho phép truyền:

- $\text{TXSTA1bits.TXEN}=1$, $\text{PIR1bits.TX1IF}=1$ hoặc
- $\text{TXSTA2bits.TXEN}=1$, $\text{PIR3bits.TX2IF}=1$.

Hoạt động truyền bất đồng bộ

- ❖ Kiểm tra truyền xong ký tự trước:
 - TXSTAxbits.TRMT=1 : sẵn sàng truyền
- ❖ Nếu dùng bit 9, nạp bit TXSTAxbits.TX9D.
- ❖ Nạp dữ liệu truyền vào thanh ghi TXREGx.

Khởi động nhận bất đồng bộ

- ❖ Khởi động số đếm SPBRGHx:SPBRGx tùy theo tốc độ. Chọn bộ đếm 8 bit hay 16 bit
 - TXSTAxbits.BRGH và
 - BAUDCONxbits.BRG16
- ❖ Chọn chế độ bất đồng bộ:
 - TXSTAxbits.SYNC=0 và
 - RCSTAxbits.SPEN=1

Khởi động nhận bất đồng bộ (tt.)

❖ Nếu dùng ngắt quãng:

- $\text{PIE1bits.RC1IE}=1$, chọn IPR1bits.RC1IP hoặc
- $\text{PIE3bits.RC2IE}=1$, chọn IPR3bits.RC2IP
- $\text{INTCONbits.GIE}=1$ và
- $\text{INTCONbits.PEIE}=1$

❖ Nếu dùng 9 bit dữ liệu:

- $\text{RCSTAxbits.RX9}=1$.

❖ Cho phép nhận:

- $\text{RCSTA1bits.CREN}=1$ hoặc
- $\text{RCSTA2bits.CREN}=1$

Hoạt động nhận bất đồng bộ

- ❖ Sau khi khởi động xong, khối EUSARTx sẽ sẵn sàng nhận ký tự đến trên ngõ RXx của nó.
- ❖ Khi nhận xong 1 ký tự:
 - Cờ PIR1bits.RC1IF (hoặc PIR3bits.RC2IF) lên 1
 - Ngắt quãng xảy ra (nếu cho phép).
- ❖ Đọc RCSTAx để lấy bit 9 và kiểm tra lỗi:
 - RCSTAxbits.RX9D
 - RCSTAxbits.FERR và RCSTAxbits.OERR
- ❖ Nếu không có lỗi, đọc dữ liệu từ thanh ghi RCREGx.
- ❖ Nếu có lỗi, xử lý lỗi.
- ❖ Nếu dùng ngắt, phải xóa cờ ngắt:
 - PIR1bits.RC1IF=0 hoặc
 - PIR3bits.RC2IF=0

Sử dụng đệm trong truyền nhận

- ❖ Tổ chức đệm truyền nhận theo dạng đệm vòng.
- ❖ Xây dựng hàm đưa dữ liệu vào đệm.
- ❖ Xây dựng hàm lấy dữ liệu ra khỏi đệm.
- ❖ Xây dựng hàm kiểm tra số dữ liệu có trong đệm.

