

Formation Automatique et Informatique Industrielle

Master 1 S2

Matière : Systèmes Embarqués et Systèmes
Temps Réel SE-STR

Par : ATOUI Hamza

Plan du cours

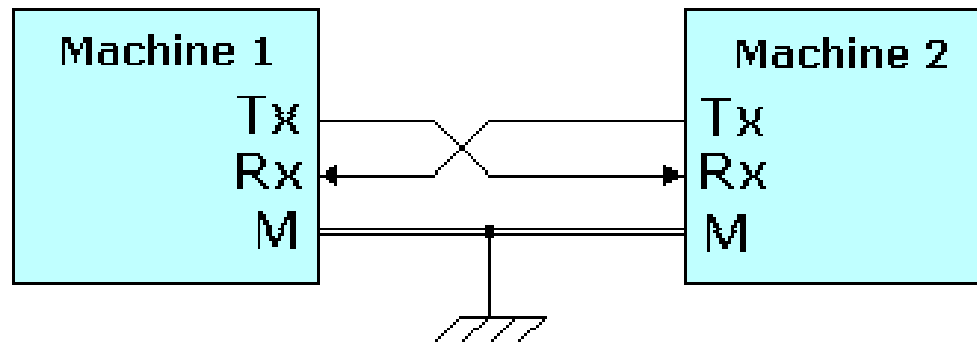
- La communication (liaison) série.

La communication (liaison) série

- Une liaison série est une ligne où les bits d'information (1 ou 0) arrivent **successivement**, soit à **intervalles réguliers** (*transmission synchrone*), soit à des **intervalles aléatoires, en groupe** (*transmission asynchrone*) La liaison **RS232** est une liaison série **asynchrone**.

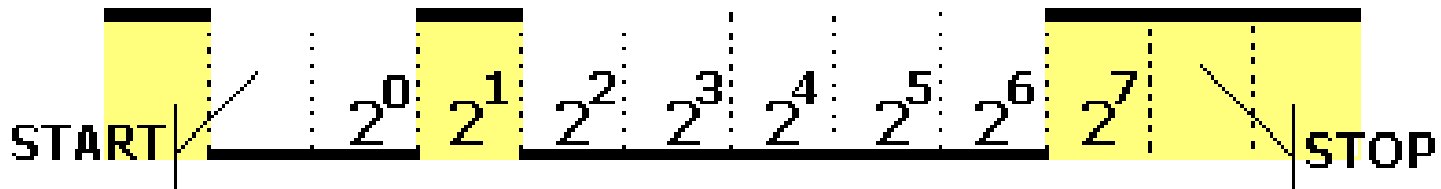
La communication (liaison) série

- Principe de RS232: l'octet à transmettre est envoyé bit par bit (poids faible en premier) par l'émetteur sur la ligne **Tx**, vers le récepteur (ligne **Rx**) qui le reconstitue par le format standard **NRZ** (**N**ON-**R**ETURN-to-**Z**ERO). La vitesse de transmission de l'émetteur doit être identique à la vitesse d'acquisition du récepteur. Ces vitesses sont exprimées en BAUDS (**1 baud** correspond à **symbole / seconde**, et dans notre cas le symbole est un bit donc, 1bit/seconde). Il existe différentes vitesses normalisées: 9600, 4800, 2400, 1200... Bauds. La communication peut se faire dans les deux sens (**duplex**), soit émission d'abord, puis réception ensuite (**half-duplex**), soit émission et réception **simultanées** (**full-duplex**). La transmission étant du type **asynchrone** (pas d'horloge commune entre l'émetteur et le récepteur), des **bits supplémentaires** sont indispensables au fonctionnement: **bit de début** de mot (**START**), **bit(s) de fin** de mot (**STOP**) D'autre part, l'utilisation éventuelle d'un **bit de parité**, permet la détection d'erreurs dans la transmission.



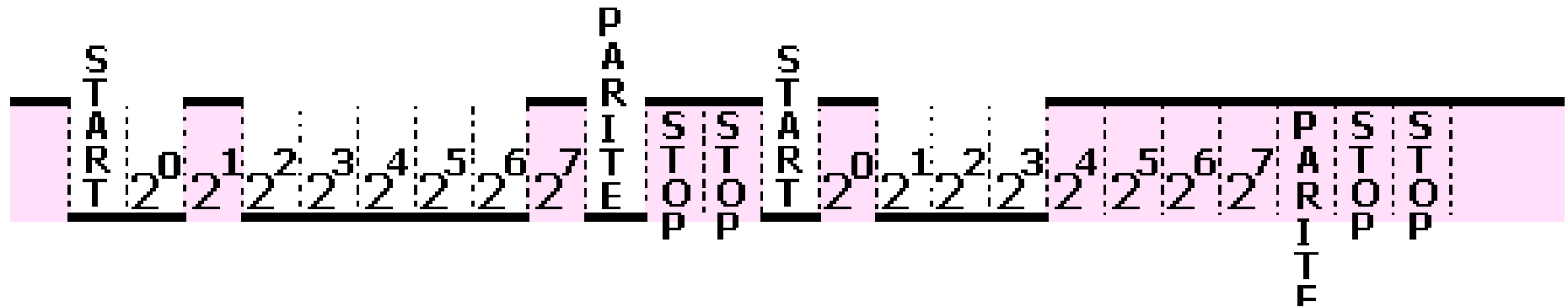
La communication (liaison) série

- Exemples: Transmission du code **0x82** avec **1 bit de stop, sans bit de parité**. 0x82 donne en binaire **10000010**.

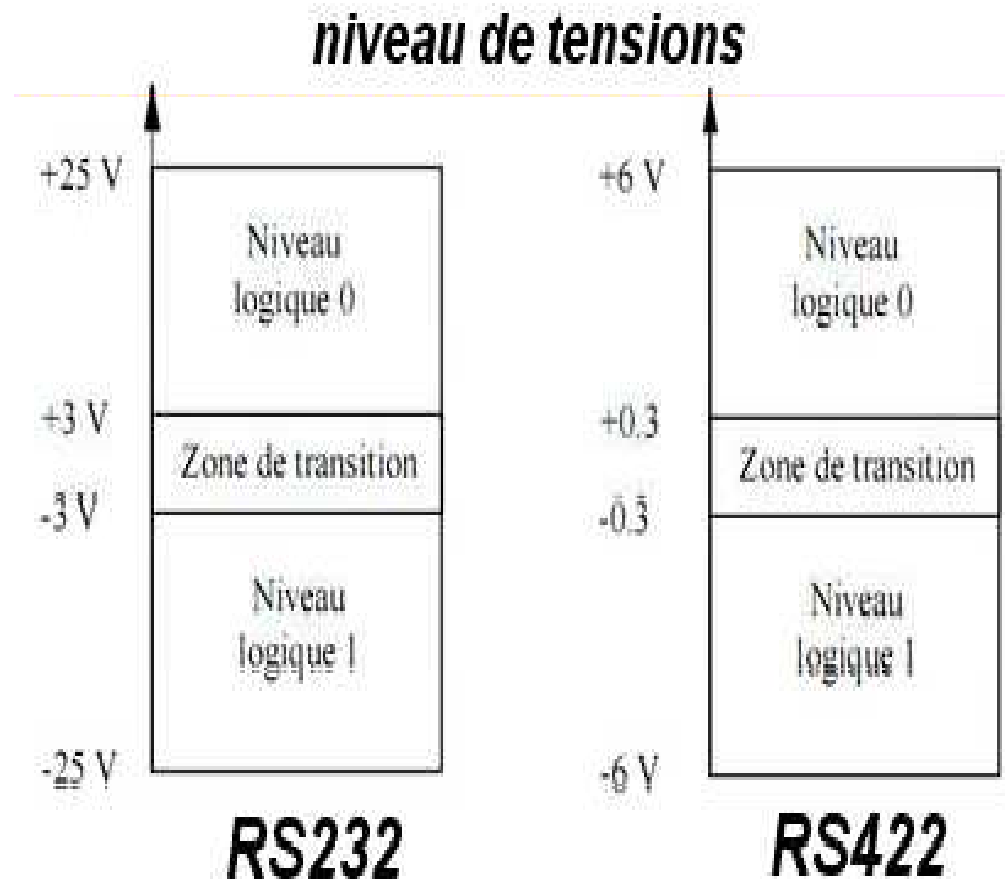


- Exemple avec parité paire: La parité est une technique qui permet de vérifier que le contenu d'un mot n'a pas été changé accidentellement lors de sa transmission. L'émetteur compte le nombre de **1** dans le mot et met le bit de parité à **1** si le nombre trouvé est impair, ce qui rend le total pair. On peut aussi utiliser la parité impaire.
- transmission de **0x82, puis 0xF1**, avec **parité paire** et **2 bits de stop**.

La communication (liaison) série



- Le port série des μ C utilise les **niveaux TTL**
 - 0 logique : **0 volt**,
 - 1 logique : **5 volt**).
- Mais le port série des ordinateurs utilise la norme **V28**.
 - 0 logique : entre **+3 volt et 25 volt**.
 - 1 logique : entre **-3 volt et -25 volt**).



La communication (liaison) série

- Brochage du connecteur DB9 des ordinateurs

-	connecteur DB 25	connecteur DB 9	-
TXD	2	3	Transmitted Data
RXD	3	2	Received Data
RQS ou RTS	4	7	Request To Send
CTS	5	8	Clear To Send
DSR	6	6	Data Set Ready
SG	7	5	Masse du signal
DTR	20	4	Data Terminal Ready
CD	8	1	Carrier Detect
RI	22	9	-



IBM RS232



MAC RS422

1	Masse
6	Sortie contrôle de flux
2	5 volts
7	Entrée contrôle de flux ou horloge externe
3	Masse
8	Réception de données +
4	Transmission de données +
9	Réception de données -
5	Transmission de données -

La communication (liaison) série

- Les μ C PIC16 possèdent le hardware USART (**U**NIVERSAL **S**YNCHRONOUS **A**SYNCHRONOUS **R**ECEIVER **T**RANSMITTER), c'est d'un module travaille en trois mode:
 - Asynchrone FULL-DUPLEX,
 - Synchrone MASTER HALF-DUPLEX,
 - Synchrone SLAVE HALF-DUPLEX.
- Dans ce qui suit, on va étudier seulement le mode asynchrone FULL-DUPLEX de notre μ C.

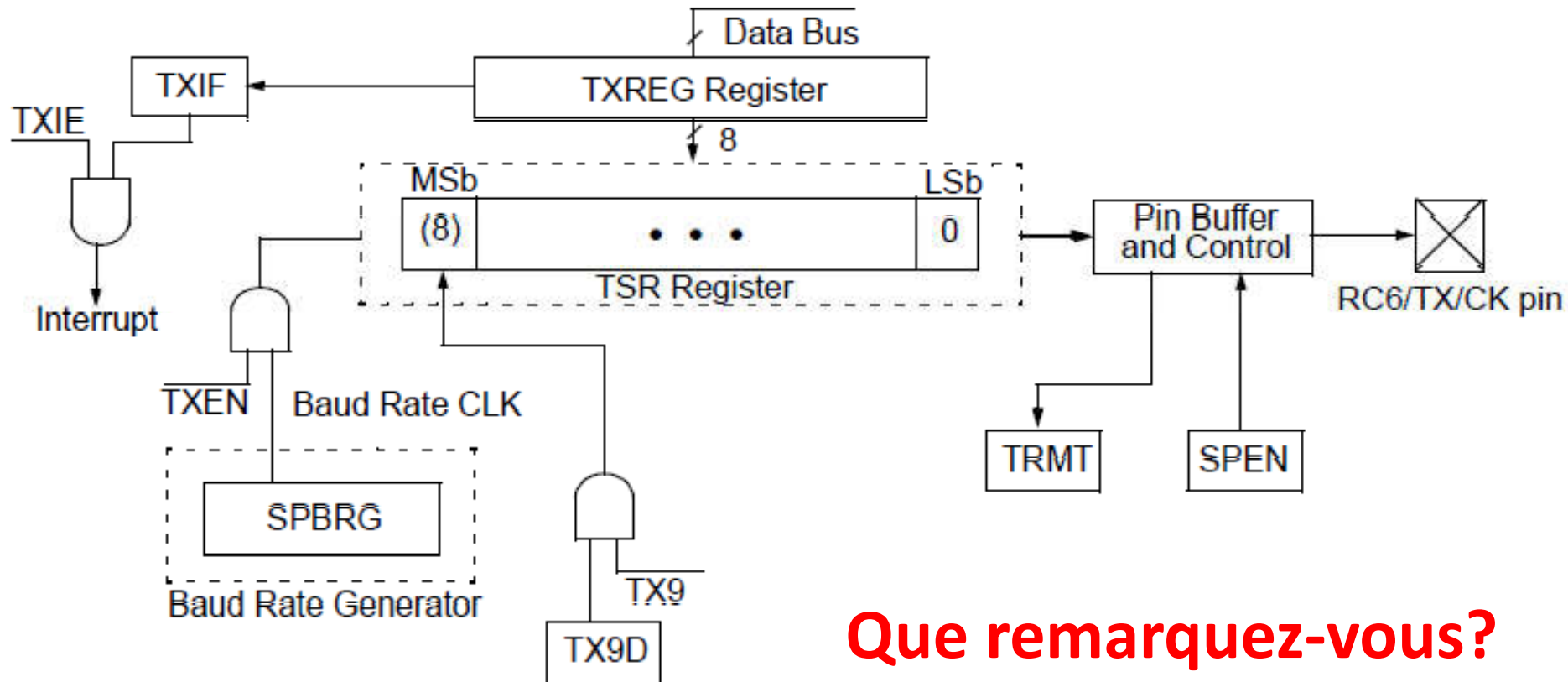
La communication (liaison) série

- Le module USART dans le mode Asynchrone se compose par 4 éléments:
 - Circuit générateur de débit en baud (BAUD RATE).
 - Circuit d'échantillonnage.
 - Circuit asynchrone de transmission.
 - Circuit asynchrone de réception.

La communication (liaison) série

- Circuit de transmission:

USART TRANSMIT BLOCK DIAGRAM

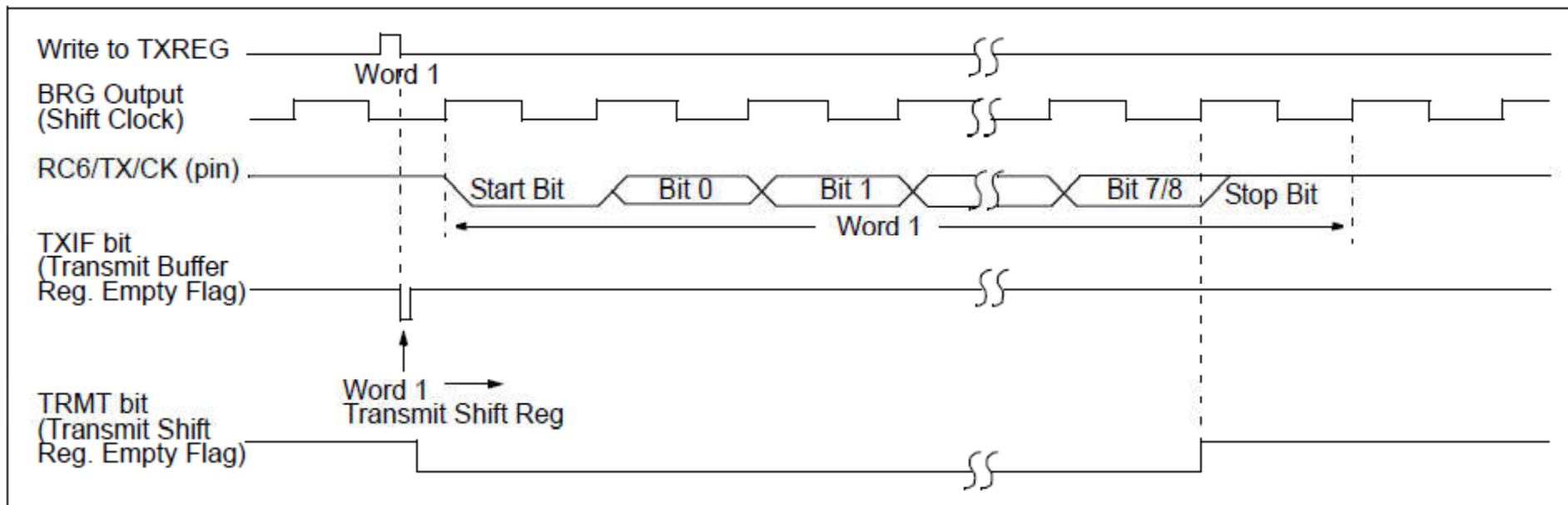


Que remarquez-vous?

La communication (liaison) série

- Chronogrammes de transmission d'un mot:

ASYNCHRONOUS MASTER TRANSMISSION

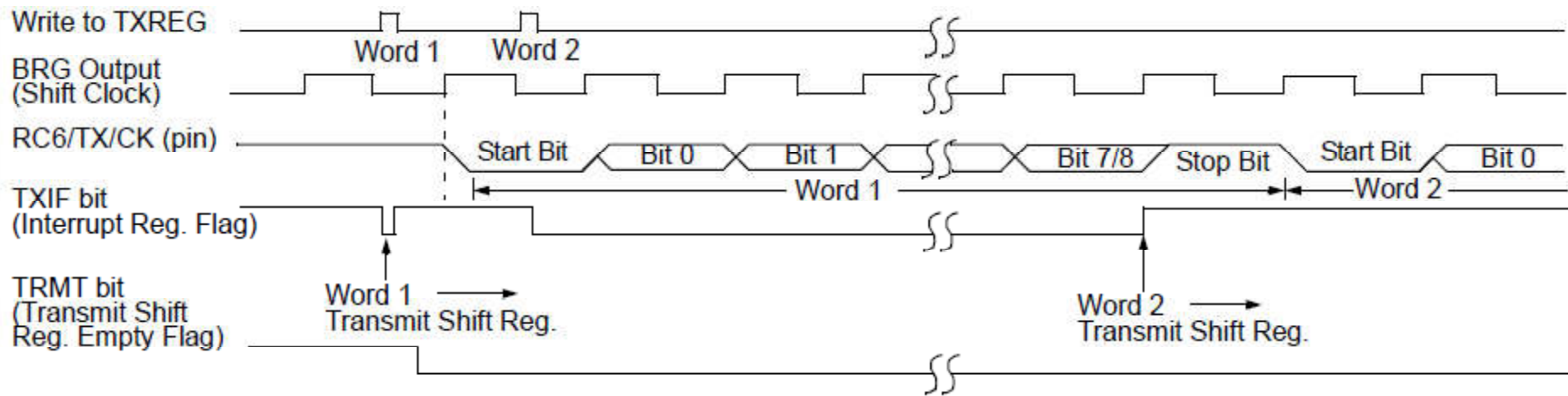


Que remarquez-vous?

La communication (liaison) série

- Chronogrammes de transmission BACK TO BACK

ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)



Note: This timing diagram shows two consecutive transmissions.

Que remarquez-vous?

La communication (liaison) série

- Comme indique le schéma block de circuit de transmission, on a plusieurs bits de contrôle; tous ces bits se trouvent dans le registre de contrôle TXSTA.

TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit 7							bit 0

CSRC: Clock Source Select bit
Asynchronous mode:
Don't care.

SYNC: USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

TX9: 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

BRGH: High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed

TXEN: Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled

TRMT: Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full

TX9D: 9th bit of Transmit Data

La communication (liaison) série

REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

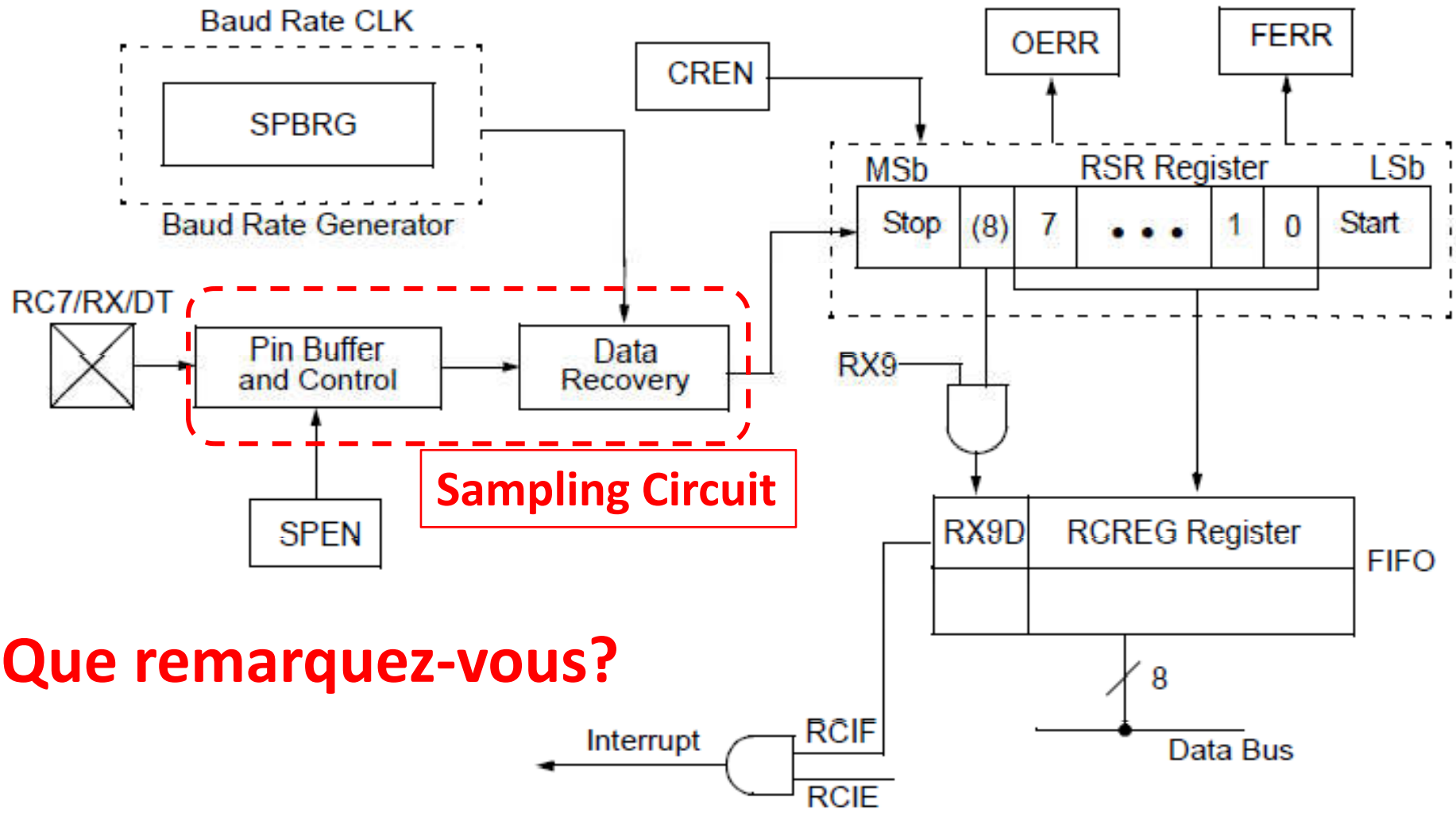
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.

La communication (liaison) série

- Circuit de réception:

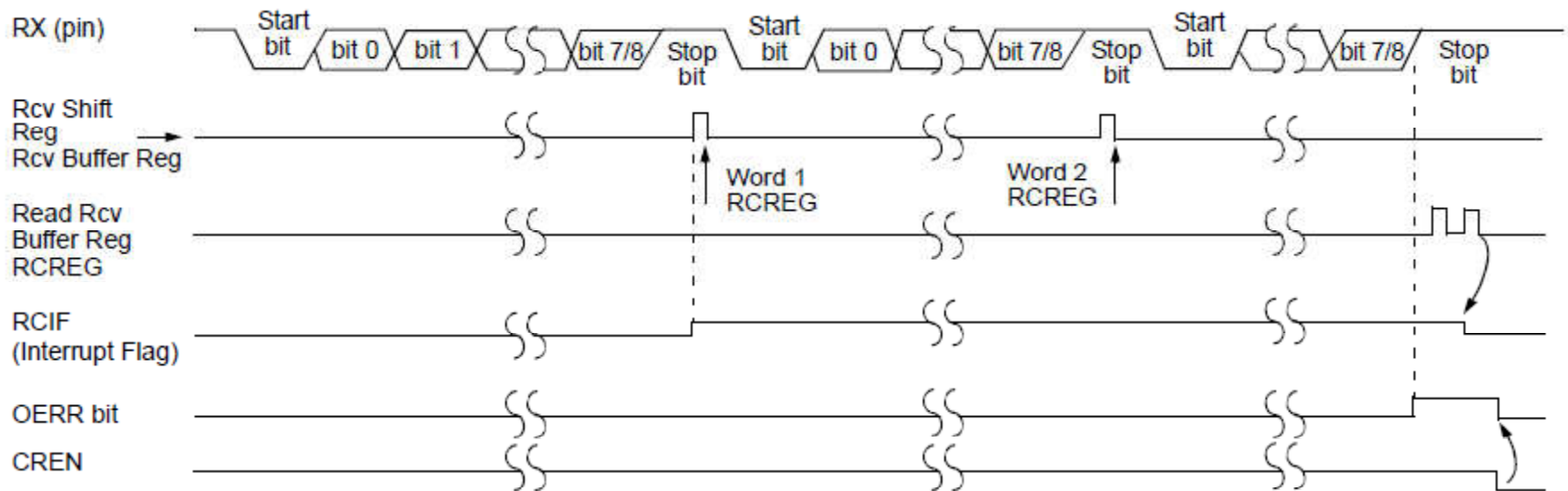
USART RECEIVE BLOCK DIAGRAM



La communication (liaison) série

- Chronogrammes de réception :

ASYNCHRONOUS RECEPTION



Note: This timing diagram shows three words appearing on the RX input. The RCREG (Receive Buffer) is read after the third word, causing the OERR (Overrun Error) bit to be set.

Que remarquez-vous?

La communication (liaison) série

- Comme le circuit de transmission, on a un autre registre de contrôle associé à la réception; c'est le registre de contrôle RCSTA.

RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7				bit 0			

SPEN: Serial Port Enable bit
1 = Serial port enabled
(configures RC7/RX/DT and
RC6/TX/CK pins as serial port pins)
0 = Serial port disabled

RX9: 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

SREN: Single Receive Enable bit
Asynchronous mode:
Don't care.

FERR: Framing Error bit
1 = Framing error
0 = No framing error

RX9D: 9th bit of Received Data

CREN: Continuous Receive Enable bit
Asynchronous mode:
1 = Enables continuous receive
0 = Disables continuous receive

ADDEN: Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection
0 = Disables address detection

OERR: Overrun Error bit
1 = Overrun error
0 = No overrun error

La communication (liaison) série

REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

La communication (liaison) série

- La liaison RS232 entre deux terminaux nécessite une convention sur le débit à utiliser au début avant de communiquer! Dans ce cas il faut utiliser tout un hardware pour générer l'horloge de cadencement, ce circuit prend le nom générateur de débit en baud se base le registre **SPBRG** (BAUD RATE GENERATOR REGISTER).
- D'après le DATASHEET, la formule de calcul de débit dépend par la valeur de registre **SPBRG** et les deux bits TXSTA(**SYNC**) et TXSTA(**BRGH**):

La communication (liaison) série

- Formule de calcul de débit en baud:

BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{OSC}/(64 (X + 1))$	Baud Rate = $F_{OSC}/(16 (X + 1))$

Legend: X = value in SPBRG (0 to 255)

Que remarquez-vous?

La communication (liaison) série

CALCULATING BAUD RATE ERROR

- Exemple de calcul:

Example shows the calculation of the baud rate error for the following conditions:

FOSC = 16 MHz

Desired Baud Rate = 9600

BRGH = 0

SYNC = 0

$$\text{Desired Baud Rate} = \frac{F_{osc}}{64(x + 1)}$$

$$9600 = \frac{16000000}{64(x + 1)}$$

$$x = 25.042$$

$$\text{Calculated Baud Rate} = \frac{16000000}{64(25 + 1)} = 9615$$

$$\text{Error} = \frac{(\text{Calculated Baud Rate} - \text{Desired Baud Rate})}{\text{Desired Baud Rate}}$$

$$= \frac{9615 - 9600}{9600} = 0.16\%$$

SPBRG $\Leftrightarrow$ x

La communication (liaison) série

- Pour communiquer avec un ordinateur personnel (compatible IBM-PC) en utilisant un des débits suivants:
- 110, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200, 128000 et 256000 baud.
- À vous, si $F_{osc} = 4\text{MHz}$ et le débit désiré = 9600 baud, quelle est la formule qu'on va utiliser pour trouver la valeur de SPBRG qui assure l'erreur minimale entre le débit désiré et le débit calculé ?

La communication (liaison) série

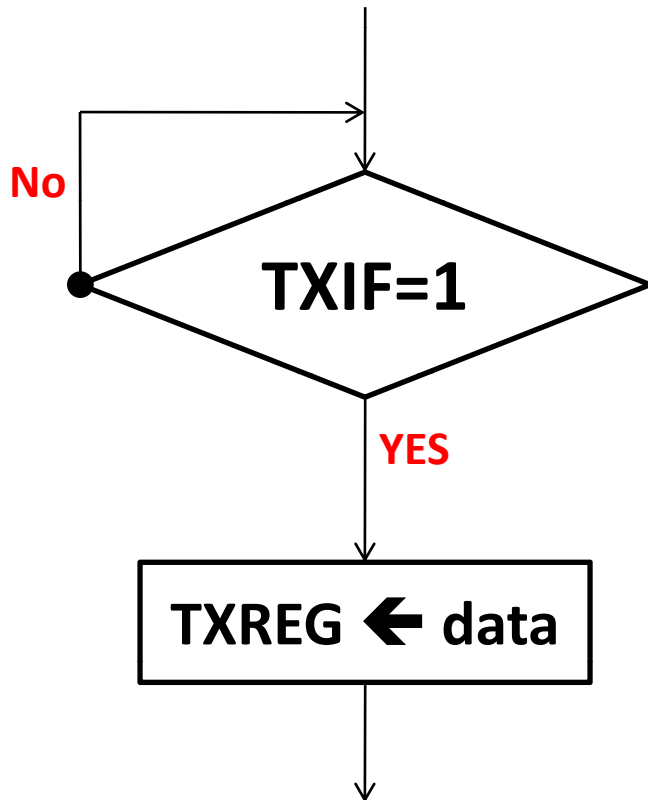
- **Attention**: la mise en marche de l'USART se fait par:
 - *Le positionnement de bit RCSTA(**SPEN**) [**SPEN** ← 1] et la configuration des pin **TX** et **RX en entrée** dans le registre **TRIS** [REF : DATASHEET].*
- Vous remarquez que le circuit de transmission est activé indépendamment par rapport au circuit de réception par le positionnement de bit TXSTA(**TXEN**) [**TXEN** ← 1] et le circuit de réception par le positionnement de bit RCSTA(**CREN**) [**CREN** ← 1]

La communication (liaison) série

- Étapes à suivre pour configurer le module USART en mode asynchrone FULL-DUPLEX:
 - Calculer la valeur de registre **SPBRG** qui assure le minimum d'erreur entre le débit désiré et le débit calculé puis charger la dedans avec les bits **SYNC** et **BRGH**,
 - SPEN ← 1, (pour la mise en marche de l'USART)
 - Configurer les pins TX et RX en entrée,
 - CREN ← 1, (pour la mise en marche de réception)
 - TXEN ← 1. (pour la mise en marche de l'émission)

La communication (liaison) série

- Organigramme de transmission d'un mot:



- Organigramme de réception d'un mot:

