

**Nom i Cognoms:** \_\_\_\_\_ **Una possible solució** \_\_\_\_\_

- 1) Volem transmetre 2048 Bytes d'un equip A a un equip B, a través d'una línia sèrie configurada a 19200 bps, 8bits, paritat senar, 1 stop bit. Cada cop que el receptor rep un byte correcte, retorna per la línia sèrie un símbol ***ack***=0xFF codificat en 8 bits sense paritat. Si ha detectat un error retorna un símbol ***nack***=0x00 codificat en 8 bits sense paritat.

Calculeu el temps de transmissió si (i indiqueu els càlculs fets):

- 1.1 La comunicació és *half-duplex* i no hi ha errors (1 punt).

Dades:      Start + 8 bits de dades + paritat + stop = 11 bits

Resposta: Start + 8 bits + stop = 10 bits

Total = 21 bits que circulen per cada Byte a enviar

$$\text{Temps} = (2048 \text{ Bytes} * 21 \text{ bits/Byte}) / 19200 \text{ bits/segon} = 2,24 \text{ segons}$$

- 1.2 La comunicació és *full-duplex* i no hi ha errors (1 punt).

Igual que abans, però al ser full-duplex, mentre circulen els 11 bits del Byte N+1, rebem els 10 bits de la resposta del Byte N.

En total tindrem:

$$1*11 \text{ (primer enviament)} + 2047*11 + 1*10 \text{ (la darrera resposta)} = 22549 \text{ bits}$$

$$\text{Temps} = 22538 \text{ bits} / 19200 \text{ bits/segon} = 1,17 \text{ segons}$$

- 1.3 La comunicació és *half-duplex* i en l'enviament d'informació falla un bit de cada 1000 enviats (1 punt).

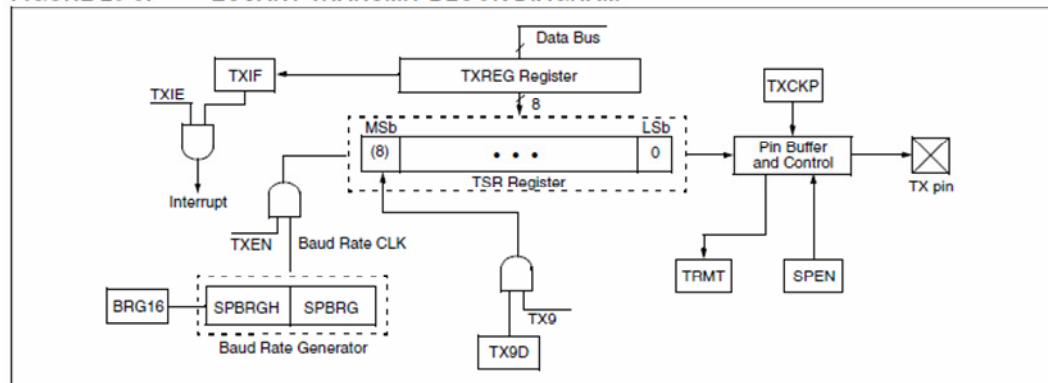
S'han d'enviar  $2048 \cdot 11 = 22528$  bits. En aquests hi haurà aproximadament 22 fallades, per tant el nombre total d'enviaments serà  $2048 + 22 = 2070$ .

Al ser halfduplex com en cas 1, tenim: 2070 enviaments i 21 bits per cadascun, per tant:

$$\text{Temps} = 2070 \text{ enviaments} * 21 \text{ bits/enviament} / 19200 \text{ bits/segon} = 2,26 \text{ segons.}$$

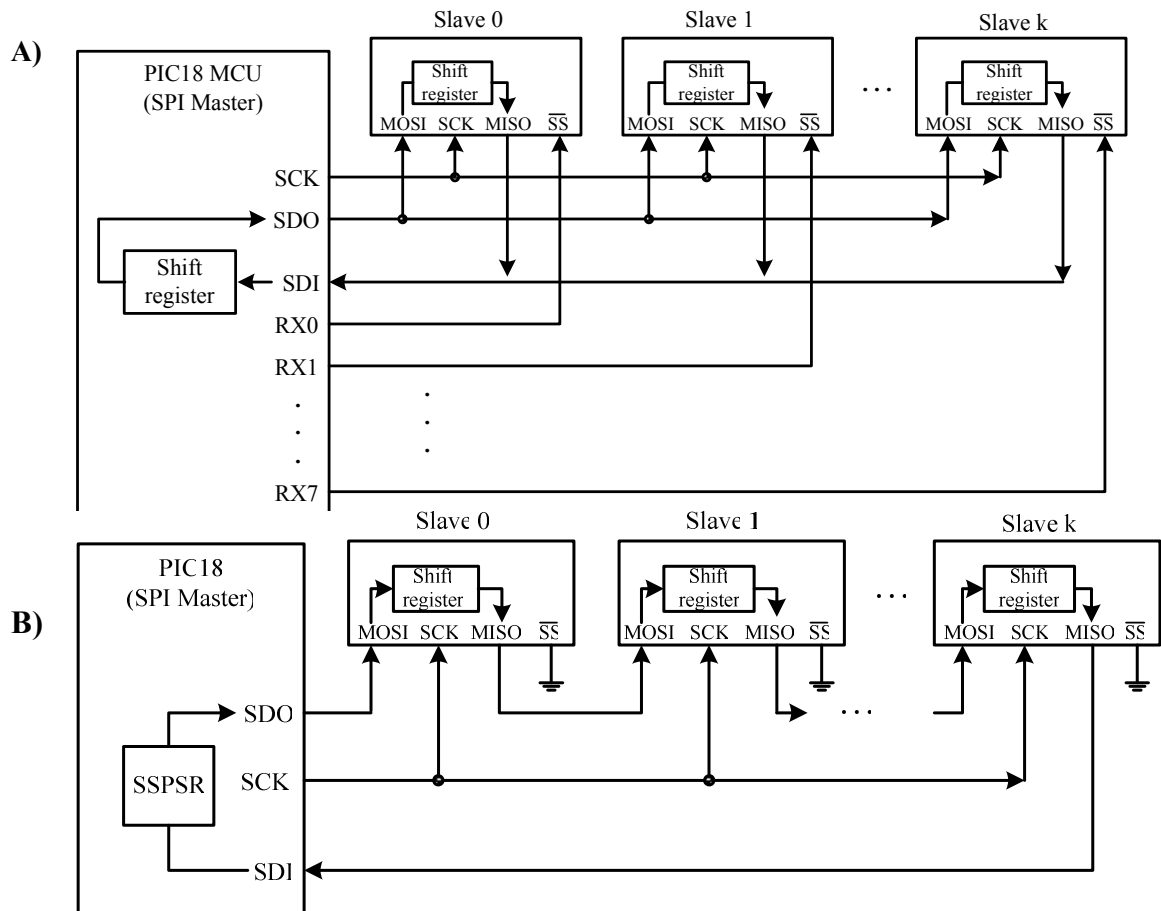
- 2) Per què el registre TSR de la figura té el bit de menys pes (lsb) a 0? (1 punt)

**FIGURE 20-3: EUSART TRANSMIT BLOCK DIAGRAM**



Aquest registre carrega automàticament el LSB a 0 per usar-lo com a bit d'Start en l'enviament. Veieu que al fer el desplaçament serà el primer bit a sortir.

3) Observeu les dues formes de connectar dispositius SPI a un microcontrolador:



Feu una rutina (en alt nivell, podeu usar pseudocodi) per llegir les dades del tercer esclau (suposant k=5) pel cas A i pel cas B (3 punts).

Cas A:

```
SS3 = 0; // seleccionem dispositiu
for (i=0; i<8; i++)
{
    SDO = DATAOut [i]; // pot ser tot zeros
    SCK=0; SCK =1; // provoquem clock
    DATA3[i] = SDI; // anem llegint els bits
}
```

Cas B:

```
for (i=0; i<8*6 ; i++) // Llegim totes les dades. Es pot
                        // fer amb menys clocks
{
    SDO = DATAOut [i]; // pot ser tot zeros
    SCK=0; SCK =1; // provoquem clock
    DATAIn[i] = SDI; // anem llegint els bits
}
for (i =0; i< 8; i++)
    DATA3[i] = DATAIn[24+i] ; // Triem les corresponents al 3
```

**Nom i Cognoms:** \_\_\_\_\_ *Una possible solució* \_\_\_\_\_

- 4) (1 Punt) Realitzeu el procés de DECODIFICACIÓ de les dades que es mostren a continuació rebudes via USB amb la codificació NRZI+Stuffed Bit:

Rx =	1	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	1
Data =																							

Nota: suposeu que el estat de repòs del bus té el valor 0.

Rx = 0	1	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	0	0	0	0	0	0	1
NRZI	0	0	1	1	1	1	1	1	0	0	0	1	1	1	1	1	0	1	1	1	1	1	0
Data =	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	0		

- 5) (1 Punt) En relació a USB, explicar el sistema utilitzar per els hubs per detectar i identificar que s'ha connectat un dispositiu.

Les resistències de pull down a l'entrada del hub provoquen que si no hi ha cap dispositiu connectat les tensions dels pins V+ i V- són zero. El hub detecta la presència d'un dispositiu degut a que connecta la seva resistència de pull-up o bé a V+ o bé a V- . Segons el tipus de dispositiu (LS o FS) el valor lògic dels pins V+ i V són (0,1) o bé (1,0).

- 6) (0.5 Punts) Indiqueu els paquets que s'envierien i qui els enviaria (el HOST o el DEVICE) en una transacció USB de sortida amb i sense errors.

El host envia primer un paquet OUT, i després el paquet amb les dades DATA. Si el dispositiu les rep correctament (i no és un dispositiu isòcron) enviarà un ACK. Si hi han errors a la recepció el dispositiu no enviarà cap paquet (NACK per Time-out). Si el dispositiu és isòcron no s'espera cap paquet de confirmació.

- 7) (0.5 Punts) Si en USB la longitud dels paquets enviats són de longitud variable, com s'indica la fi de transmissió.

Trencant la lògica diferencial durant el temps de dos bits (EOP).