

Dispositius Programables

Final - Gener 2015

1. Disseny un programa en ensamblador que commuti l'estat d'un oscil·lador implementat amb el Timer 0 de manera que quan rep del port sèrie la lletra "S" soni el to, quan rebi la lletra "A" apagi el to i quan rebi la lletra "T" commuti l'estat de l'oscil·lador. La recepció per port sèrie es fa per interrupció (`__vector_18`). Es disposa de les següents subrutines per facilitar la feina: *setup_serie* (configura el port sèrie per interrupció), *tx* (transmet pel port sèrie el contingut del registre r16), *setup_timer* (configura el Timer 0 per generar un to), *genera* (genera un to consistent en un senyal quadrat a la pota 6 del Port D), *apaga* (manté la pota 6 del Port D a un valor constant i per tant no es genera cap to).
 - a) Descriu de quina manera controlaries que la pota de sortida de l'oscil·lador generi el to o bé quedi sense oscil·lar. Descriu qualitativament com dissenyaries les subrutines *genera* i *apaga*. (0.5)
 - b) Dibuixa el graf de la màquina d'estats corresponent a aquest programa. (0.5)
 - c) Presenta el codi font del programa complet aprofitant les subrutines donades. (1)
2. Sobre el registre d'estat de la CPU de l'AVR respon a les següents preguntes: (1)
 - a) Suposant que r16=136 i r17=156, i després de la instrucció ADD r16, r17. Quins flags es veuran afectats per aquesta instrucció i quin serà el seu valor?
 - b) Suposant que r16=136 i r17=156, i després de la instrucció SUB r16, r17. Quins flags es veuran afectats per aquesta instrucció i quin serà el seu valor?
3. La subrutina *print* té com a paràmetre el registre r16 i treu pel port sèrie la representació decimal del valor que se li passa com a paràmetre. Disposes de la subrutina *tx* i de la subrutina *div10*. La subrutina *div10* té com a paràmetre el registre r16. Aquesta subrutina retorna a r16 el quocient de la divisió per 10 i a r17 el residu d'aquesta divisió.
 - a) Disseny una macro *val2dec* que actuï sobre el registre r16 transformant el seu valor numèric (en el marge 0:9) en la seva representació amb caràcter ASCII. Recordeu que el caràcter ASCII corresponent al '0' és el 0x30. (0.5)
 - b) Disseny la subrutina *print* de manera transparent. Si consideres oportú, es pot utilitzar la macro *val2dec*. (1)
 - c) Disseny de manera transparent la subrutina *div10*. (1)
4. Aprofitant *div10*, dissenya la subrutina *div10pila* on el pas de paràmetres i resultats es fa a través de la pila. Dibuixa com queda el bloc d'activació (pila) quan es crida a aquesta subrutina. (1.5)

5. Aquí teniu l'estructura d'un programa. Supposeu que disposem de les subrutines *tx* i *rx* utilitzades a les pràctiques. Supposeu que pel port sèrie s'està rebent un 3. Considereu que tant *tx* com *rx* tenen per paràmetre el registre r16.

```
1      .global main
2
3      main:   call rx
4             push r16
5             call txr
6             pop r16
7             ret
8
9      txr:   in r28, SPL
10          in r29, SPH
11          ldd r16, Y+3
12          tst r16
13          breq fi
14          dec r16
15          push r16
16          call tx
17          call txr
18          pop r16
19      fi:    ret
```

- Indica la seqüència d'execució d'aquest programa en forma de llista en funció del número de línia. (0.5)
- Podem considerar les subrutines *tx* i *rx* com una ampliació del conjunt d'instruccions de la CPU, considerant que tardaran més clocks que les instruccions originals? Si comparem la pila just abans de fer un *call tx* o *call rx* i just després d'aquest call, la pila queda exactament igual? Justifica les respostes. (0.25)
- Dibuixeu l'evolució que tindrà la pila fent referència a l'apartat a). Per simplificar considereu que *call tx* i *call rx* no afecten a la pila. (0.5)
- Existeix algun bucle en aquest codi? Es fa alguna acció repetitiva en funció del valor rebut pel port serie (*call rx*)? Justifica. (0.5)
- Es pot considerar que hi ha recursivitat en aquest codi? Justifica la resposta. (0.25)
- Proposa un altra codi diferent que faci el mateix, però amb una estructura totalment diferent. (1)