Dispositius Programables

Final - Gener 2013

1. La següent funció de Python fa el sumatori des del valor que es passa com a paràmetre fins a 1 decrementant en 1 unitat. És una funció recursiva que té un paràmetre i retorna un resultat. Es demana:

```
def suma (valor):
        if valor == 1:
                return valor
        else:
                return valor + suma(valor-1)
```

   a) Justifica quin mecanisme de pas de paràmetre / resultat es pot fer servir per implementar aquesta funció en assemblador. (0.5)

   b) Quina diferència hi ha entre el mecanisme de pas de paràmetre per valor o per referència? (0.5)

   c) Implementa una rutina en assemblador que faci el mateix que la funció de Python. Fes servir un pas de paràmetres per valor utilitzant la pila. Defineix què cal fer en el programa principal abans i després de cridar a la rutina. (1.5)

2. Defineix el concepte d'ortogonalitat i dóna un exemple detallat on NO es compleix en el cas de l'AVR. (1)

3. Dissenya una subrutina que faci el quadrat del número que se li passa com a paràmetre a través del registre r16. El resultat el retorna al parell de registres r17:r16. Suposeu que no disposeu de la instrucció de multiplicació. (1)

4. Aquí teniu un programa complert. Suposeu que les constants corresponents a les adreces dels registres es troben correctament definides. L'objectiu d'aquest programa és rebre pel port sèrie un *índex* (byte) i treure pel port sèrie el caràcter en codi ASCII segons la posició definida per *índex* en la taula de caràcters definida per *text*.

```
EXTRA = 0

.global main
.global __vector_18
.global __do_copy_data
.global __do_clear_bss

.section .text
/* rutina de transmissió de byte, el valor a transmetre està al registre r16 */
tx:
        lds     r17,UCSR0A
        sbrs    r17,5
        rjmp    tx
        sts     UDR0,r16
        ret

/* rutina d'interrupció per recepció de byte a la USART */
__vector_18:
        lds     r16,UDR0
        sts     index,r16
        rcall   processa
        reti

processa:
        ldi     r26,lo8(text+index)
        ldi     r27,hi8(text+index)
        ld      r16,X
        call tx
        reti

main:   /* En aquesta part es troba el codi necessari per
        configurar correctament el port serie de la USART.*/
        /*habilitem interrupcions */
        sei
loop:   rjmp    loop
        ret

.section .data
text:   .ascii "Hola que tal"
text_fi:

.section .bss
index:  .byte 0
```

a) Si detectes errors en el codi, enumera'ls i reescriu un nou codi sense errors. (1)

b) Quantes subrutines existeixen en aquest programa? Quantes rutines d'interrupció? Justifica si es pot considerar la rutina d'interrupció com una subrutina més? Emmarca en un quadrat cadascuna de les subrutines / rutines d'interrupció. (0.5)

c) Millora les rutines que hi hagi de tal manera que siguin transparents al programa principal i no afectin al seu funcionament. (1)

d) Enumera cadascuna de les rutines i indica si es fa algun tipus de pas de paràmetres i/o resultats. En cas afirmatiu indica quin són. (0.5)

e) Modifica el programa de manera que si EXTRA = 1 el programa retorni pel port sèrie el caràcter definit per índex i el caràcter anterior. A més ho ha de fer en aquest ordre. Però igualment si EXTRA = 0 continua funcionant com està indicat a l'enunciat. (1)

f) Quan EXTRA canvia de valor entre 0 i 1, cal que es torni a compilar el programa? Per què? Descriu amb paraules com es podria fer sense necessitat de tornar a compilar. (0.5)

g) Defineix una rutina *exam* a la que se li passa com a paràmetre la taula *text* i treu el primer element de la taula com a resultat. Quin mecanisme de pas de paràmetre faries servir? El passaries per valor o per referència? Defineix igualment el que s'hagi de fer en el programa principal abans i després de cridar a la rutina. (1)