

Dispositius Programables

Final - Gener 2014

1. Dissenya un programa en ensamblador que commuti l'estat del led cada vegada que rep pel port sèrie la seqüència "LED". Si en qualsevol moment es rep un "espai", s'ha de transmetre pel port sèrie una 'O' si el led està encès o una 'X' si està apagat. L'espai no interfereix en la seqüència de lletres que es vagi introduint. 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). Es recorda que el led es troba al bit 5 del Port B de l'avr i que escriure un "1" en el registre PIN commuta el valor del registre PORT.
 - a) Defineix les macros: `init_led` (inicialitza i configura l'AVR per fer servir el led i el deixa apagat), `led_on` (encendre led), `led_off` (apaga led), `get_led` (torna al bit T el valor de l'estat del led), `toggle_led` (canvia l'estat del led). S'ha de tenir en compte que han de ser transparents. Indica quins paràmetres es passen i quins resultats es tornen en cadascuna d'elles.
 - b) Justifica si és millor que les definicions de (a) siguin macros o subrutines. Una macro amb paràmetres és el mateix que una rutina amb paràmetres?
 - c) Descriu com funciona la interrupció de transmissió del port sèrie.
 - d) Dibuixa el graf de la màquina d'estats corresponent a aquest programa.
 - e) Presenta el codi font del programa complet, definint les constants necessàries i utilitzant les macros de (a) sempre que es pugui. (1)
2. Dissenya una subrutina `suma_parells` a la que se li passa com a paràmetre, a través de registre, l'adreça d'una taula i realitza la suma dels 100 primers valors **parells**. Considereu que el primer valor de la taula correspon a la posició 0. El resultat de la suma s'ha de retornar a través dels registres r1:r0. Aquesta subrutina ha de ser transparent.
 - a) Defineix prèviament una subrutina `suma` que sumi amb precisió de 2 bytes els words r1:r0 i r3:r2 i retorni el resultat a r1:r0. Aquesta subrutina ha de ser transparent.
 - b) Què s'ha de fer amb cadascun dels flags del registre d'estat de la CPU perquè tots indiquin el valor correcte després d'haver cridat a la subrutina `suma`? (0.25)
 - c) Escribe el codi font de la subrutina completa. (1)
 - d) Aprofitant la subrutina `suma_parells`, crea una subrutina `suma_parells_pila` on el pas de paràmetres i de resultats es fa a través de la pila. Indica que cal fer abans i després de la crida a aquesta subrutina. Dibuixa com queda la pila (bloc d'activació) amb aquesta subrutina. (1)
3. Dissenya una subrutina `canvia_pila` de manera que el valor que se li passa com a paràmetre a través de la pila el substitueixi pel valor que hi ha a la variable `var` situada en una posició de memòria de dades i retorni el valor corresponent al que hi havia originalment a `var`. La subrutina ha de ser transparent. Indica que cal fer abans i després de la crida a aquesta subrutina. Dibuixa com queda la pila (bloc d'activació) amb aquesta subrutina. (1.5)

4. Aquí teniu l'estructura d'un programa.

```
00 .global main
01 .section .text
02 .macro retco
03     pop r31
04     pop r30
05     icall
06 .endm

07 main:
08     /*inicialització del procés 1 */
09     call main2
10 bucle1:
11     retco
12     /* procés 1 */
13     rjmp     bucle1
14     ret

15 main2:
16     /*inicialització del procés 2 */
17 bucle2:
18     retco
19     /* procés 2 */
20     rjmp     bucle2
21     ret
```

- Descriu què li passa a la pila i al comptador de programa quan s'executa la macro *retco*.
- Elimina i substitueix tot el que processa l'assemblador deixant estrictament les instruccions que executarà la CPU de l'AVR i torna a numerar les línies. Considera els comentaris com a blocs de codi del programa. (0.25)
- Indica la seqüència d'execució del programa en forma de llista d'instruccions de cpu. Considera els comentaris com a blocs de codi del programa. Indica el contingut de la pila després de cada instrucció. Indica en quin moment es produeix una seqüència de repetició en l'execució. (1)
- Quantes rutines existeixen en aquest programa? Emmarca en un quadrat cadascuna de les rutines. Es pot considerar el programa principal com una rutina? (0.25)
- S'executa en algun moment algun dels *ret* que apareixen en el codi? Hi ha algun error en el codi? (0.25)
- Es pot considerar que aquest esquema correspon a una estructura d'execució de 2 processos en paral·lel? Justifica la resposta aportant pros i contres.