

Dispositius Programables

Final - Gener 2017

durada 3h

1. Es vol dissenyar un programa que treballarà amb matrius. Aquestes matrius estaran definides a memòria de dades en temps de compilació. La disposició de les matrius a memòria serà com s'indica en el següent esquema:

Disposició de la memòria física:

f_0				f_1				f_2				...	
0	1	2	...	C	C+1	C+2	...	2C	2C+1	2C+2	

Disposició de la matriu:

...	2	1	0	f_0
...	C+2	C+1	C	f_1
...	2C+2	2C+1	2C	f_2
...				...

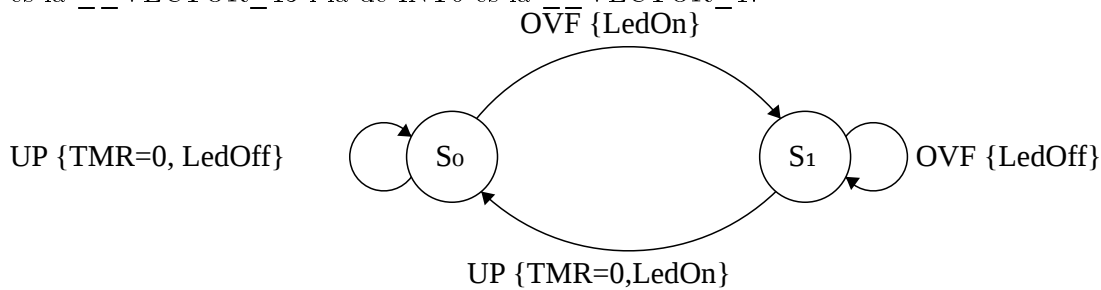
on C és el nombre de columnes i F és el nombre de files que es defineixen en temps de compilació.

- a) Quin tipus de disposició és aquesta? Big-endian, Little-endian o cap de les anteriors. Justifica avantatges o inconvenients.
 - b) Es vol disposar de 2 matrius $M1$ i $M2$ que estan inicialitzades a 0. Defineix en ensamblador aquestes dues matrius a la secció que correspongui.
2. Per fer referència a un element de la matriu, es fa servir la notació tradicional, on l'element (0,0) fa referència al que està situat a la part superior esquerra de la matriu. El primer element fa referència a la fila i el segon element fa referència a la columna. Les files s'enumeren de dalt a baix començant per 0 i les columnes s'enumeren d'esquerra a dreta començant també per 0.
 - a) Defineix una subrutina "escriu" fent servir codi d'alt nivell (python) que té com a paràmetres: la matriu sobre la que es vol escriure, l'element escollit en format fila i columna, i el valor que es vol escriure. Considereu que l'estructura de dades que disposeu és la definida anteriorment. Quins recursos fareu servir de la CPU per passar cadascun d'aquests paràmetres a la subrutina "escriu" en ensamblador?
 - b) Defineix la subrutina "escriu" amb codi ensamblador de manera transparent.
 - c) Defineix la subrutina "escriu" a partir de "escriu" amb pas de paràmetres per pila.
 - d) La subrutina "llegeix" fa el cas contrari a "escriu". Té com a paràmetres la matriu que es vol llegir i l'element escollit en format fila i columna. A diferència, retorna el resultat d'aquest element. Podeu reestructurar les subrutines "escriu" i "llegeix" de manera que totes les accions que siguin comunes es puguin separar en un bloc de codi apart? Aquests blocs de codi apart poden ser macros o subrutines? En cas afirmatiu redefiniu "escriu" i "llegeix" fent servir (sense definir) aquests blocs de codi comuns.

3. Supposeu que disposeu de la subrutina “TX” i de “llegeix”.

- Definiu una subrutina “print_fil” que envii per port sèrie els elements de la fila que se li passa com a paràmetre. La matriu també se li passa com a paràmetre. Els elements s’imprimeixen d’esquerra a dreta.
- Definiu una subrutina “print_columna” que envii per port sèrie els elements de la columna que se li passa com a paràmetre. La matriu també se li passa com a paràmetre. Els elements s’imprimeixen de dalt a baix.
- Definiu “print_fil” sense fer servir “llegeix”. Compara les dues implementacions indicant avantatges i inconvenients.
- Definiu la subrutina “suma” que té com a paràmetres l’adreça de les dues matrius i retorna la suma de matrius substituint una de les dues matrius. Justifiqueu la conveniència de fer servir o no les subrutines “llegeix” i “escriu”.

4. Es vol dissenyar un programa que controli el LED de l’arduino segons indica el diagrama d’estats de la següent figura. Per fer-ho, es farà servir el Timer1 i un polsador extern. Supposeu que disposeu d’una subrutina “setup_timer” que configura el Timer1 en mode normal de manera que es genera la interrupció d’overflow cada segon. El polsador està connectat a la pota PD2 de manera que es fa servir la interrupció INT0 i també es disposa de la subrutina “setup_int0” que configura aquesta interrupció per flanc de pujada. Recordeu que el Led està a la pota PB5 i que la interrupció d’overflow de Timer1 és la `__VECTOR_13` i la de INT0 és la `__VECTOR_1`.



- Dissenyau el circuit analògic corresponent a la interconnexió entre un polsador i la pota d’entrada de l’arduino de manera que quan sigui polsat el polsador es generi un flanc de pujada a la pota.
- Definiu macros/subrutines (segons considereu el més adient) per les accions: LedOn i LedOff.
- Implementeu de manera sistemàtica un programa en ensamblador que correspongui a aquest graf donat.
- Describeu què ha de fer l’usuari per a que el Led sempre es vegi encès.