Eric Díez Apolo

Exercicis Tema 10

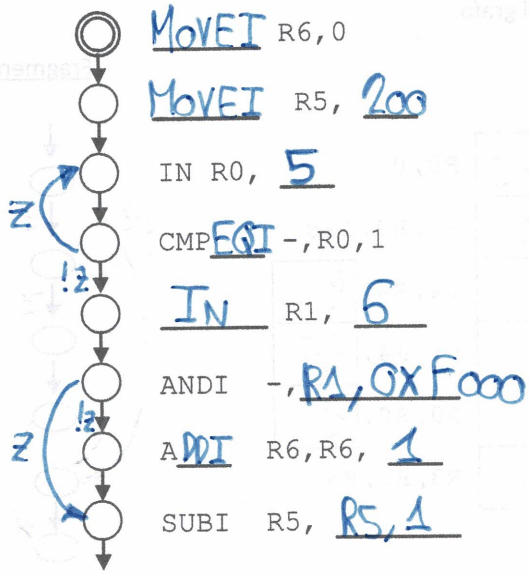
S'ha d'entregar un únic fitxer PDF que inclogui la solució que vosaltres proposeu als problemes plantejats. El fitxer PDF no te que ser necessàriament una solució feta per ordinador, pot ser una solució escrita a ma i digitalitzada. El PDF ha d'incloure una capçalera on s'indiqui el vostre nom i cognoms, i l'enunciat de cada pregunta abans de la vostra resposta.

Exercici 1

Se ha conectado a la UPG un dispositivo externo de entrada que nos envía valores **naturales 16 bits** y que tiene el registro de status en la dirección 5 del espacio de direccionamiento de entrada y el de datos en la 6. Este dispositivo tiene un efecto lateral en la lectura del dato sobre su registro de estado.

Este dispositivo de entrada nos envía de forma asíncrona 200 valores y se desea saber cuántas operaciones de multiplicación darían overflow si cada dato recibido lo multiplicásemos por el valor 16. El resultado de esta cuenta debe dejarse en el registro R6.

a) Completad el grafo de estados si estuviésemos utilizando una unidad de control específica (UCe) junto a la UPG para que realice la función anteriormente descrita. Indicad los arcos y las etiquetas de los arcos (z, !z, o nada) que falten (en caso que falten) y completad las casillas de cada palabra de control.



b) Usando el procesador SISC Harvard uniciclo, completad el fragmento de código ensamblador SISA para que realice la función anteriormente descrita. Una vez finalizada la operación, el programa deberá dejar el resultado de 16 bits en el registro R6.

```
MOVI   R6, 0
MOVI   R5, 0XC8
MOVHI  R5, 0X00
IN     R0, 5
BNZ    R0, -2
IN     R1, 6
MOVI   R2, 0X00
MOVHI  R2, 0XF0
ANDI   R3, R1, R2
BZ     R3, 2
ADDI   R6, R6, 1
ADDI   R5, R5, -1
BNZ    R5, -10
```

Exercici 2

Dados los dos siguientes fragmentos de código en C (el código no tiene que hacer algo útil), indicad como se implementarían cada uno en un procesador que use la UPG vista en clase, utilizando la UC de **propósito específico** (UCe) y la UP de **propósito general** (UPG). Todos los datos son **naturales**.

Fragmento 1

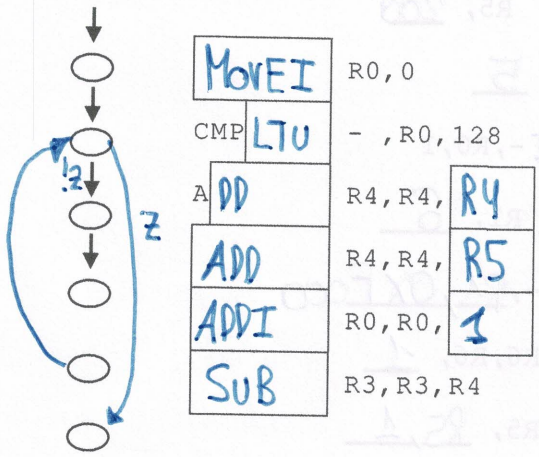
```
for (R0=0; R0<128; R0++) {
    R4=R4*2+R5;
}
R3=R3-R4;
```

Fragmento 2

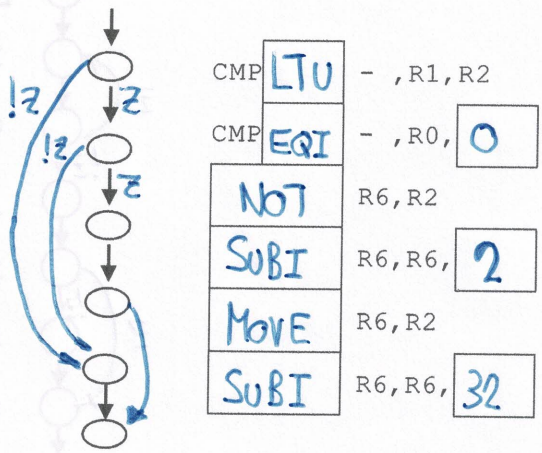
```
if ((R1>=R2) && (R0!=0)) {
    R6=not(R2)-2;
} else {
    R6=R2;
}
R6=R6-32;
```

a) Completad los dos fragmentos de grafo de estados de la UC de **propósito específico** para que junto con la UPG formen un procesador que realice la funcionalidad descrita en los fragmentos de código anteriores. Indicad los arcos que faltan, las etiquetas de los arcos (z, !z, o nada) y completad las casillas de cada palabra de control que se especifica con mnemotécnicos a la derecha de cada nodo del grafo.

Fragmento 1



Fragmento 2



b) Completad los fragmentos de programa en lenguaje ensamblador SISA para que el procesador formado por la unidad de control de propósito general (UCG) junto con la UPG realicen las funcionalidades descritas en los fragmentos de código en C (el código no tiene que hacer algo útil). El código SISA ya escrito siempre utiliza el registro R7 para valores temporales. En las comparaciones, hay que interpretar los datos como valores **naturales**. Rellenad la parte subrayada que falta.

Fragmento 1

@I-Mem	
0x0000	MovI R0, 0
0x0002	MOVI R7, 0x80
0x0004	CMPLTU R7, R0, R7
0x0006	BZ R7, 4
0x0008	ADD R4, R4, R4
0x000A	ADD R4, R4, R5
0x000C	ADDI R0, R0, 1
0x000E	BNZ R0, -6
0x0010	SUB R3, R3, R4

Fragmento 2

@I-Mem	
0x0000	CMPLEU R7, **R2, R1**
0x0002	BZ R7, 4
0x0004	BNZ R0, 3
0x0006	NOT R7, R2
0x0008	ADDI R6, R7, -2
0x000A	BZ R0, 1
0x000C	ST R6, R2, R2
0x000E	ADDI R6, R6, -32

Exercici 3

Dado el siguiente fragmento de código donde en el *ciclo i* se ejecuta la instrucción `MOVI R3, 0xFE`, rellenad el siguiente cronograma indicando el valor de las señales de la UCG o UPG y los valores de los registros. *(nota: para facilitar la lectura del cronograma no hace falta que pongáis el **0x** delante de los valores hexadecimales de los registros. Ya se sobreentienden)*

```
MOVI R3,0xFE
MOVHI R3,0x00
XOR R2, R2, R1
SHL R1, R1, R3
BNZ R1, -3
OUT 17,R2
```