

Exercicis Tema 8

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.

Dels exercicis 5 i 6 només cal fer-ne un (a elecció de cadascú).

Exercici 1

Dissenyeu la implementació interna, similar a la que surt a la documentació del tema 8, d'un banc de registres de 4 registres de 64 bits amb 3 ports de lectura (ports A, B i C) i un d'escriptura (port D).

Exercici 2

Dissenyeu la implementació interna, similar a la que surt a la documentació del tema 8, d'una ALU que tingui com entrades de dades una senyal X i una senyal Y de 32 bits, com a sortida de dades una senyal W de 32 bits. Com a senyals de control: una senyal *oper* d'1 bit i una senyal *func* de dos bits, que en funció de les senyals faci les operacions indicades en la següent taula.

W	<i>oper</i> = 0	<i>oper</i> = 1
<i>func</i> = 00	ADD(X,Y)	Y
<i>func</i> = 01	SUB(X,Y)	LTU(X,Y)
<i>func</i> = 10	NAND(X,Y)	LEU(X,Y)
<i>func</i> = 11	SHA(X,Y<5..0>)	EQ(X;Y)

Exercici 3

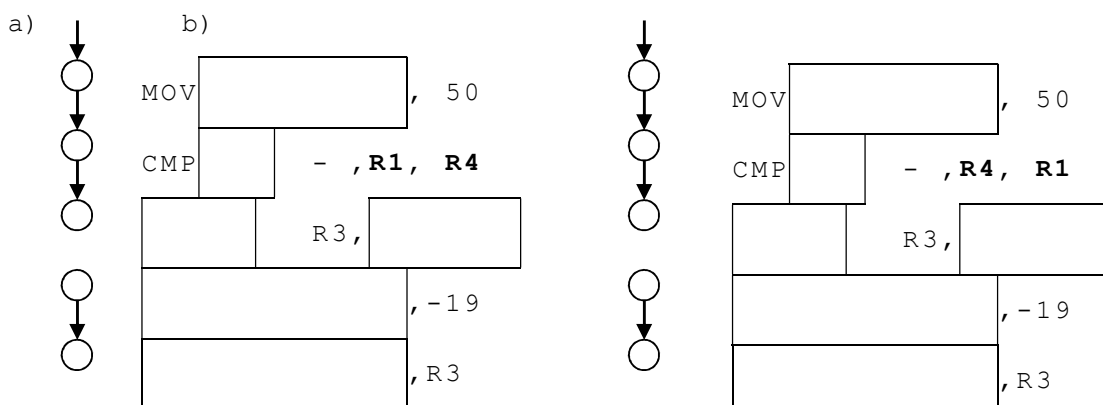
Construye el fragmento de grafo correspondiente a la Unidad de Control que ejecuta la funcionalidad que se indica. Tratad todos los valores como si fuesen números enteros.

Funcionalidad	Fragmento de grafo de la UC.
<pre>. . while (R5 != R4) { R2 = R2/4-1 ; if (R2 <= 3) { R1 = R1+R4; } R5++; } R3 = R3*2+1; .</pre>	

Exercici 4

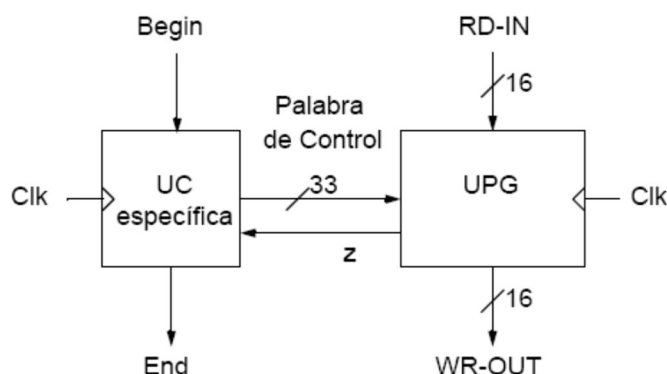
Completad los dos fragmentos de grafo de estados de la UC de **propósito específico** para que junto con la UPG formen un procesador que realice la funcionalidad descrita mediante el siguiente código en C. Indicad los arcos que faltan, las etiquetas de los arcos (z, !z, o nada) y completad las casillas de cada palabra de control que se especifica con mnemotécnicos a la derecha de cada nodo del grafo. Todos los datos son naturales. Ambos fragmentos ejecutan la misma funcionalidad pero de forma distinta.

```
R4 = 50;
if (R1<=R4) {
    R3 = R5 / 8;
} else {
    R3 = R5 - 19;
}
R6 = R3;
```



Exercici 5

Dibuixa el graf de Moore de la Unitat de Control específica que permet realitzar les funcionalitats descrites per a cadascuna dels següents apartats, utilitzant l'estructura del PPE següent, en el qual l'entrada/sortida de dades a través dels busos RD-IN i WR-OUT es sincronitza mitjançant els senyals Begin i End.



Esquema general del PPE amb unitat de control específica i unitat de procés general

Per resoldre aquest exercici és important observar l'esquema a blocs de la UPG, amb els senyals de la paraula de control, i la taula de funcionalitats de l'ALU de la documentació.

Dibuixeu la llegenda del graf i pels nodes del graf indiqueu les etiquetes dels arcs i els mnemotècnics de les paraules de control. **Si en algun cas no és possible resoldre algun dels problemes plantejats utilitzant la UPG explica'n el motiu.**

- El PPE rep n nombres de 16 bits pel bus RD-IN, en calcula el màxim, i l'envia a l'exterior pel bus WR-OUT. El primer valor que arriba és n (on $n \geq 1$) i estarà disponible en el mateix cicle que el senyal Begin valgui 1 (cicle k). El primer nombre pel càlcul arriba al cicle següent (cicle $k+1$). En el cas que $n > 1$, el segon nombre arriba al cicle $k+3$ i la resta de nombres arriben cada 4 cicles (és a dir, als cicles $k+7$, $k+11$, $k+15$, etc.). Una vegada s'ha iniciat una operació, s'ha d'ignorar el senyal Begin fins que aquesta acabi. En el mateix cicle en que el resultat està disponible a la sortida, i per tant el senyal End es posa a 1, Begin es pot posar a 1 i hem de començar un nou càlcul.
- El PPE ha de realitzar un senzill càlcul: canviar el signe d'un nombre enter codificat en Ca2 amb 16 bits que entra pel bus RD-IN i treure'l pel bus WR-OUT. El cicle en què Begin val 1, el nombre enter codificat en Ca2 està present al bus d'entrada RD-IN. Una vegada el senyal Begin es posa a 1, s'ignora fins al cicle en el qual el senyal End es posa a 1. Si en aquest últim cicle, mentre el resultat està disponible a la sortida WR-OUT, el senyal Begin es posa a 1 cal començar un nou càlcul.

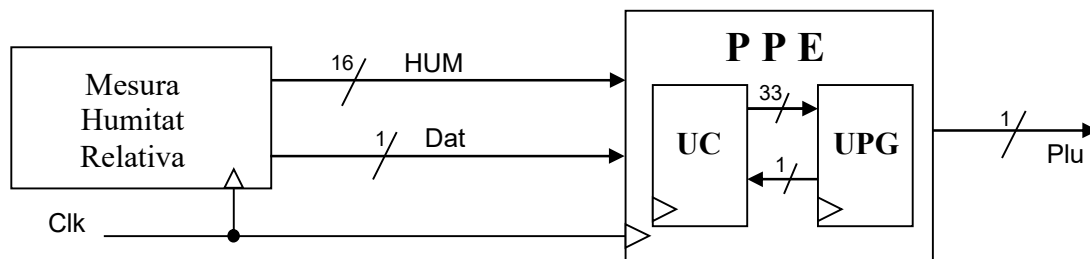
- c) El PPE ha de sumar dos nombres X i Y de 16 bits que arriben pel bus RD-IN al mateix cicle, quan Begin val 1, i retornar el resultat de la suma pel bus WR-OUT al cicle següent, posant el senyal de sortida End a 1. Cal poder començar un nou càlcul en el mateix cicle en el qual el resultat està disponible a la sortida WR-OUT.

Exercici 6

Dissenyem un PPE, fent servir la UPG, que predigui si plourà o no. A tal efecte, disposem d'un dispositiu que és capaç de mesurar la humitat relativa de l'aire i codificar-la en un número natural de 16 bits. Aquest dispositiu mesura la humitat un cop cada minut i posa el valor mesurat en el bus HUM de 16 bits i, simultàniament activa el senyal dat a 1 durant 1 cicle. El bus HUM i el senyal Dat estan sincronitzades amb el mateix rellotge que el PPE.

El PPE ha de generar un senyal Plu que valgui 1 durant 1 cicle si, durant les darreres 200 mesures (en els 200 darrers minuts) cada valor de la humitat llegida pel PPE ha estat sempre igual o superior a l'anterior lectura. En qualsevol altre cas el senyal Plu valdrà 0.

L'esquema

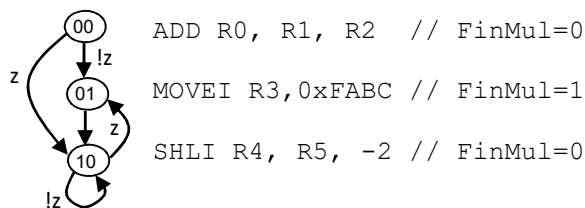


Un operador inicialitza tot el sistema un cop al mes (per tant no hi ha desbordaments).

Dibuixa el graf d'estats de la UC per a que el PPE tingui la funcionalitat descrita.

Exercici 7

Dado el siguiente grafo de estados de una UC de propósito específico para que, junto con la UPG básica, formen un PPE que lo ejecute. En la figura se muestra el esquema de la unidad de control correspondiente a una implementación con dos roms. Una rom para el estado siguiente (ROM_Q+) y otra rom para las salidas (ROM_OUT).



Indica el contenido de las ROMS en hexadecimal de igual modo como lo hicisteis en la práctica 4 (o haréis en un futuro próximo). Utilizad el valor 0 para los bits que sean x. Si los contenidos no están en hexadecimal no se corregirá el ejercicio.

@ROM	Contenido ROM_Q+
@0	
@1	
@2	
@3	
@4	
@5	
@6	
@7	

@ROM	Contenido ROM_OUT
@0	
@1	
@2	
@3	

