

Apellidos y nombre: Grup: DNI:.....

Examen 3. (Temas 8, 9, 10 y 11)

- Duración del examen: 2 horas.
- La solución de cada ejercicio se tiene que escribir en el espacio reservado para ello en el propio enunciado.
- No podéis utilizar calculadora, móvil, apuntes, etc.
- La solución del examen se publicará en Atenea mañana y las notas antes del 1 de diciembre.

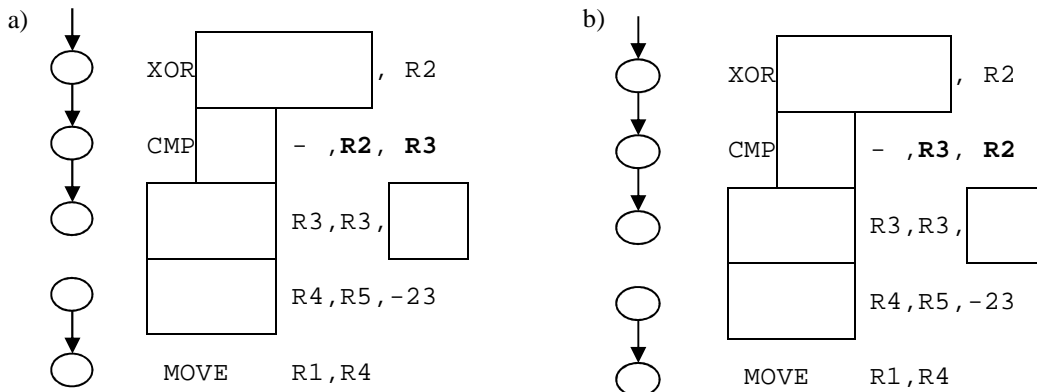
Ejercicio 1 (1 punto)

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.

```

R2 = 0;
if (R3 <= R2) {
    R3 = R3 / 16;
} else {
    R4 = R5 - 23;
}
R1 = R4;

```

**Ejercicio 2 (1 punto)**

- a) Indica el valor que debe tener cada uno de los bits de la palabra de control de la UPG (sin subsistema de I/O ni memoria) para que realice, durante un ciclo, la acción concreta especificada mediante el mnemotécnico. Indicad con **x** las casillas cuyo valor no importe para la ejecución de la instrucción. En caso de que no se pueda realizar la acción tachar **toda la línea de señales**. (0.5 puntos)

Mnemotécnico	@A	@B	Rb/N	OP	F	In/Alu	@D	WrD	N (hexa)
XORI R0, R3, -6									
OUT R0 // MOVEI R2, 0x1234									
ADDI -,R4, 0 // IN R3									

- b) Indica el mnemotécnico que corresponde a cada una de las siguientes palabras de control de la UPG (sin subsistema de I/O ni memoria). (0.5 puntos)

Mnemotécnico	@A	@B	Rb/N	OP	F	In/Alu	@D	WrD	N (hexa)
	100	xxx	x	xx	xxx	x	xxx	0	X X X X
	101	xxx	0	01	100	x	xxx	0	1 2 3 4
	001	011	1	00	111	0	010	1	X X X X

Ejercicio 3 (2 puntos)

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;
}
```

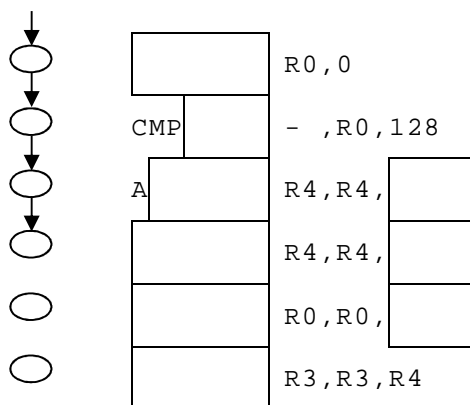
```
R3=R3-R4;
```

Fragmento 2

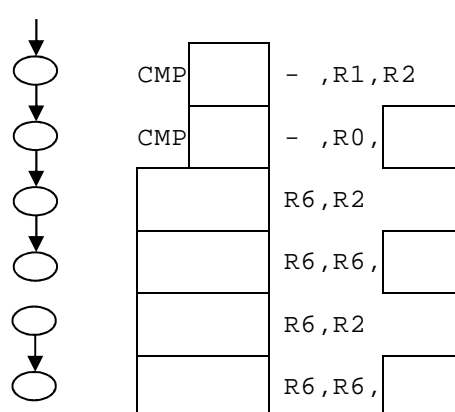
```
if ((R1>=R2) && (R0!=0)) {
    R6=not(R2)-2;
} else {
    R6=R2;
}
R6=R6-32;
```

- 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, lz, 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 (0.5 puntos)



Fragmento 2 (0.5 puntos)



- 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 (0.5 puntos)

@I-Mem	
0x0000	_____ R0, 0
0x0002	MOV_____ R7, _____
0x0004	CMP_____ R7, R0, R7
0x0006	B_____ R7, _____
0x0008	_____ R4, R4, R7
0x000A	ADD R4, R4, _____
0x000C	_____ R0, R0, _____
0x000E	B_____ R0, _____
0x0010	_____ R3, R3, R4

Fragmento 2 (0.5 puntos)

@I-Mem	
0x0000	CMP_____ R7, R2 , R1
0x0002	B_____ R7, _____
0x0004	B_____ R0, _____
0x0006	_____ R7, R2
0x0008	ADDI R6, R7, _____
0x000A	B_____ R0, _____
0x000C	_____ R6, R2, R2
0x000E	_____ R6, R6, -32

Ejercicio 4 (0.5 puntos)

Completa la siguiente tabla ensamblando las instrucciones en ensamblador SISA o desensamblando las instrucciones en lenguaje máquina según sea necesario. Indica poniendo NA en la casilla aquellos casos en los que la instrucción no corresponda al lenguaje SISA-I.

Lenguaje máquina SISA	Lenguaje ensamblador SISA
0x0A1F	
	LDB R3, -1(R5)
0x9780	

Apellidos y nombre: Grup: DNI:.....

Ejercicio 5 (1.25 puntos)

Escribid sobre la siguiente tabla el valor de los bits que tiene la palabra de control del SISC-Harvard uniclo y la señal *TknBr* durante el ciclo en que se ejecuta cada una de las instrucciones SISA que se indican. Poned x siempre no importe el valor de ese bit para la ejecución correcta de la instrucción (aunque se pueda saber el valor codificando la instrucción). Suponed que antes de ejecutar cada instrucción el contenido de los registros, de los puertos de entrada/salida y de la memoria de datos es cero.

Instrucción SISA	Palabra de Control del SISC Harvard uniciclo														N (hexa)	ADDR-IO (hexa)
	@A	@B	Rb/N	OP	F	-i/l/a	@D	WrD	Wr-Out	Rd-In	Wr-Mem	Byte	TknBr			
MOVHI R3, -4																
ST 0(R4), R0																
BZ R1, -6																
OUT 44, R1																
LDB R2,8(R0)																

Ejercicio 6 (1.25 puntos)

Indicad qué cambios hay en el estado del computador después de ejecutar cada una de las instrucciones de la tabla suponiendo que **antes de ejecutarse cada una** de ellas el PC vale 0xA722, el contenido de todos los registros es 0xFFFFC y que el contenido de todas las posiciones pares de la memoria de datos es 0x13 y el de todas las posiciones impares de la memoria de datos es 0x57. Utiliza el mnemotécnico MEM_b[...], MEM_w[...] y DataOut[...] para indicar los cambios en la memoria y los puertos de E/S respectivamente.

Instrucción a ejecutar	Cambios en el estado del computador
MOVHI R1, 34	
LDB R2, 4(R6)	
BNZ R7, 3	
STB -3(R6), R2	
LD R3, 2(R2)	

Ejercicio 7 (1.5 puntos)

Indica el contenido de la tabla de la ROM (sólo las celdas en blanco) correspondiente al bloque ROM_CTRL_LOGIC. Indica los valores que tomarían las señales para ejecutar correctamente las instrucciones. Indica con x los valores de los bits del contenido de la ROM que puedan valer 0 o 1.

Dirección ROM					Contenido de la ROM																					
					Bnz	Bz	Wr-Mem	Rd-In	Wr-Out	WrD	Byte	Rb/N	-i/l/a1	-i/l/a0	OP ₁	OP ₀	MxN1	MxN0	MxF	f2	f1	f0	MxD1	MxD0		
I ₁₅	I ₁₄	I ₁₃	I ₁₂	I ₈																						
0	0	0	0	X																						A / L
0	0	0	1	X																						CMP
0	0	1	0	X																						ADDI
0	0	1	1	X																						LD
0	1	0	0	X																						ST
0	1	0	1	X																						LDB
0	1	1	0	X																						STB
0	1	1	1	X																						(NOP)
1	0	0	0	0																						BZ
1	0	0	0	1																						BNZ
1	0	0	1	0																						MOVI
1	0	0	1	1																						MOVHI
1	0	1	0	0																						IN
1	0	1	0	1																						OUT
1	0	1	1	X																						(NOP)
1	1	X	X	X																						(NOP)

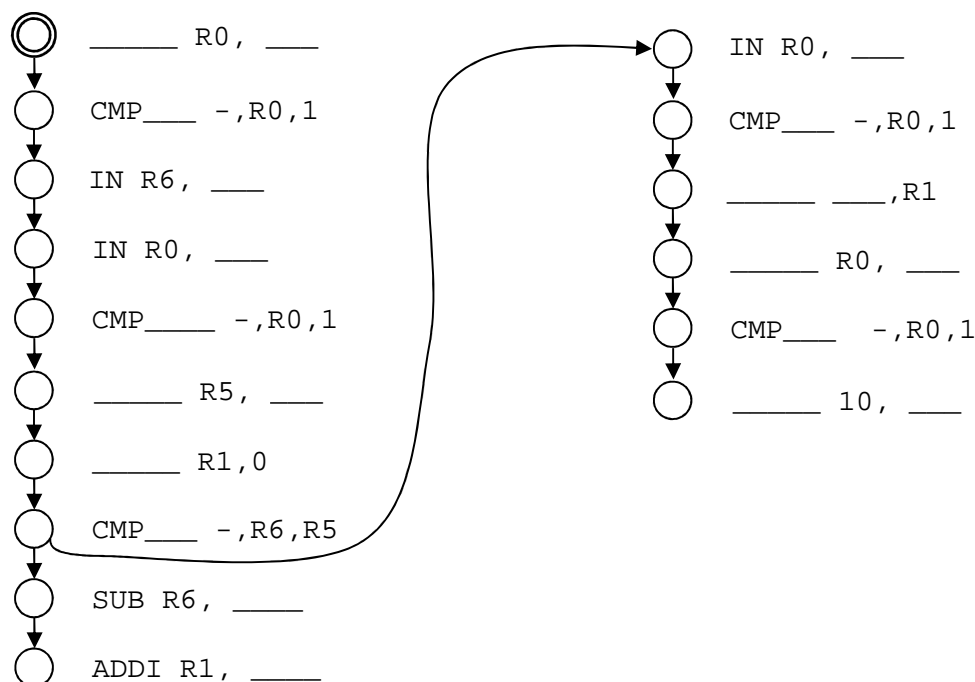
Ejercicio 8 (1.5 puntos)

Se desea resolver el mismo problema de la división que se resolvió en el examen E2 usando un PPE pero ahora usando la UPG. Se desea calcular el cociente y el resto de la división de dos naturales mayores que 0. Dados n (numerador) y d (denominador) tal que $n > 0$ y $d > 0$, calcula los naturales q y r que cumplen que $n = d * q + r$, y $0 \leq r < d$ mediante restas sucesivas.

Se ha conectado a la UPG un dispositivo externo de entrada que nos envía valores **naturales de 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 primero n (el numerador) y luego d (el denominador). Una vez recibidos los dos valores deberemos calcular q (el cociente) y r (el resto) de la división haciendo restas sucesivas.

a) Usando la UPG junto a una UCe, una vez realizada la división se deberán enviar los resultados q y r (primero q y seguidamente r) a un dispositivo externo de salida con efecto lateral en el puerto de datos. Este dispositivo tiene el registro de status en la dirección 9 de entrada y el de datos en la 10 del espacio de direccionamiento de salida. 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. (0.75 puntos)



b) Usando el procesador SISC Harvard unicycle, completad el fragmento de código ensamblador SISA para que realice la función de división anteriormente descrita. Una vez finalizada la operación de división, el programa deberá dejar el resultado en posiciones consecutivas en memoria de datos a partir de la dirección 0x1234 (0.75 puntos)

```

IN      R0, 5
B____
IN      R6, 6
IN      R0, 5
B____
IN      R5, 6
____    R1, 0
CMP____ R7, R6, R5
B____   R7, ____
SUB     R6, R6, R5
ADDI    R1, R1, ____
B____   R7, ____
MOVI    R7, 0x34
MOVHI   R7, 0x12
____    ____, R1
____    ____, R6
  
```