

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

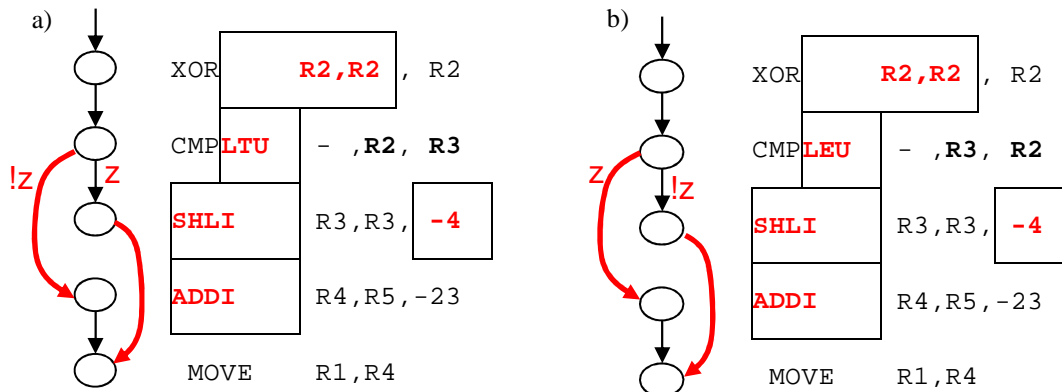
Examen 3. (Temas 8, 9, 10 y 11)

- 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 (Objetivos 8.x) (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 arcs (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;
```



Criterio de corrección: Ambos apartados se corrigen conjuntamente. -0.25 puntos por el primer nodo incorrecto y -0.50 puntos por cada uno de los siguientes nodos incorrectos.

Un nodo es erróneo si falta alguno de los arcos que salen de él, si alguna etiqueta es incorrecta o los destinos de alguno de sus arcos es incorrecto. También es incorrecto un nodo si la salida especificada mediante mnemotécnicos (operación, registros o valor inmediato) es incorrecta. Las funcionalidades que sean iguales en ambos apartados y estén mal en ambos, sólo contarán como un error.

Hacemos una excepción a esta regla: Si falta la U o la I se descuenta 0.1 por ese nodo si sólo tiene ese fallo.

Ejercicio 2 (Objetivo 8.x) (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	Wd	N (hexa)
XORI R0, R3, -6	011	xxx	0	00	010	0	000	1	F F F A
OUT R0 // MOVEI R2, 0x1234	000	xxx	0	10	001	0	010	1	1 2 3 4
ADDI -,R4, 0 // IN R3	100	xxx	0	00	100	1	011	1	0 0 0 0

Criterio de corrección: -0.25 puntos por cada fila y columna incorrecta, escogiendo el número mínimo de filas y/o columnas que cubren todos los errores.

- 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	Wd	N (hexa)
OUT R4	100	xxx	x	xx	xxx	x	xxx	0	X X X X
CMPLTUI -,R5,0x1234	101	xxx	0	01	100	x	xxx	0	1 2 3 4
SHL R2,R1,R3	001	011	1	00	111	0	010	1	X X X X

Criterio de corrección: -0.25 puntos por cada mnemotécnico incorrecto.

Ejercicio 3 (Objetivos 8.x) (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

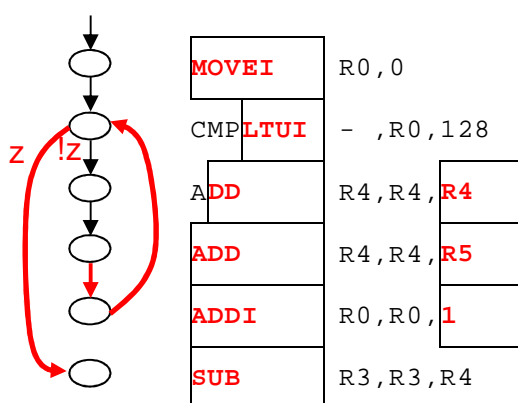
```
for (R0=0; R0<128; R0++) {
    R4=R4*2+R5;
}
R3=R3-R4;
```

Fragmento 2

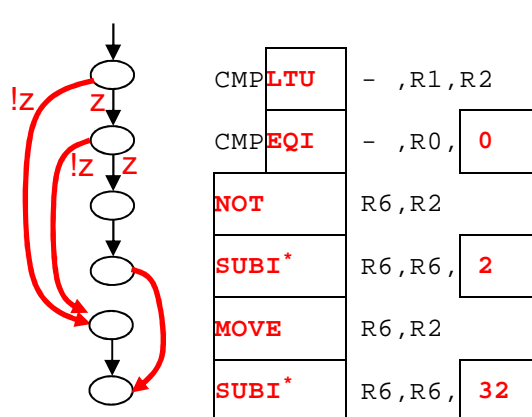
```
if ((R1>=R2) && (R0!=0)) {
    R6=not(R2)-2;
} else {
    R6=R2;
}
R6=R6-32;
```

- 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, !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 (0.5 puntos)



Fragmento 2 (0.5 puntos)



* También se puede hacer un ADDI con los valores negativos

Criterio de corrección: Cada fragmento vale 0.5 puntos y se corrigen por separado. -0.25 puntos por cada nodo incorrecto.

Un nodo es erróneo si falta alguno de los arcos que salen de él, si alguna etiqueta es incorrecta o los destinos de alguno de sus arcos es incorrecto. También es incorrecto un nodo si la salida especificada mediante mnemotécnicos (operación, registros o valor inmediato) es incorrecta. Las funcionalidades que sean iguales en ambos apartados y estén mal en ambos, sólo contarán como un error.

Hacemos una excepción a esta regla: Si falta la U o la I se descuenta 0.1 por ese nodo si sólo tiene ese fallo.

- 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	MOVI R0, 0
0x0002	MOV I R7, 127
0x0004	CMP LEU R7, R0, R7
0x0006	BZ R7, 4
0x0008	SHL/SHA R4, R4, R7
0x000A	ADD R4, R4, R5
0x000C	ADDI R0, R0, 1*
0x000E	BNZ R0, -7
0x0010	SUB R3, R3, R4

Fragmento 2 (0.5 puntos)

@I-Mem	
0x0000	CMP LEU R7, R2 , R1
0x0002	BZ R7, 4
0x0004	BZ R0, 3
0x0006	NOT R7, R2
0x0008	ADDI R6, R7, -2
0x000A	BNZ R0, 1
0x000C	AND/OR R6, R2, R2
0x000E	ADDI R6, R6, -32

* También se puede haber usado el registro R7

Criterio de corrección: Cada fragmento vale 0.5 puntos y se corrigen por separado. -0.1 puntos por la primera instrucción incorrecta y -0.2 puntos por cada una de las siguientes instrucciones incorrectas.

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

Ejercicio 4 (Objetivos X) (0.5 puntos)

Completa la siguiente tabla ensamblando las instrucciones en ensamblador SISA-I 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	SHL R3, R5, R0
0x5AFF	LDB R3, -1(R5)
0x9780	MOVHI R3, -128 / MOVHI R3, 0x80

Criterio de corrección: -0.25 puntos por cada fila incorrecta.

Ejercicio 5 (Objetivos X) (1.25 puntos)

Escribid sobre la siguiente tabla el valor de los bits que tiene la palabra de control del SISC-Harv 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	011	xxx	0	10	010	00	011	1	0	0	0	x	0	XXFC*	XX	
ST 0(R4), R0	100	000	0	00	100	xx	xxx	0	0	0	1	0	0	0000	XX	
BZ R1, -6	001	xxx	x	10	000	xx	xxx	0	0	0	0	x	1	FFF4	XX	
OUT 44, R1	001	xxx	x	xx	xxx	xx	xxx	0	1	0	0	x	0	XXXX	2C	
LDB R2,8(R0)	000	xxx	0	00	100	01	010	1	0	0	0	1	0	0008	XX	

* También se considerará correcto haber puesto FFFC

Criterio de corrección: -0.25 puntos por cada fila y columna incorrecta, escogiendo el número mínimo de filas y/o columnas que cubren todos los errores.

Ejercicio 6 (Objetivos X) (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 0xFFFF 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	R1=0x22FC	PC=0xA724
LDB R2, 4(R6)	R2=0x0013	PC=0xA724
BNZ R7, 3		PC=0xA72A
STB -3(R6), R2	MEM _b [0xFFFF9]=0xFC	PC=0xA724
LD R3, 2(R2)	R3=0x5713	PC=0xA724

Criterio de corrección: -0.25 puntos por cada fila incorrecta sin contar los PC que siguen el secuenciamiento implícito. Si los PC que siguen el secuenciamiento implícito están mal sólo descuentan -0.25 puntos adicionales.

Ejercicio 7 (Objetivos X) (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																				
I ₁₅	I ₁₄	I ₁₃	I ₁₂	I ₈	Bnz	Bz	Wr-Mem	Rd-In	Wr-Out	WrD	Byte	Rb/N	-i//a1	-i//ao	OP ₁	OP ₀	MxN1	MxN0	MxF	f2	f1	f0	MxD1	MxD0	
0	0	0	0	X			0				x	1							0						A / L
0	0	0	1	X			0				x	1							0						CMP
0	0	1	0	X	0	0	0	0	0	1	x	0	0	0	0	0	0	0	1	1	0	0	0	1	ADDI
0	0	1	1	X			0				0	0							1						LD
0	1	0	0	X			1				0	0							1						ST
0	1	0	1	X			0				1	0							1						LDB
0	1	1	0	X	0	0	1	0	0	0	1	0	x	x	0	0	0	0	1	1	0	0	x	x	STB
0	1	1	1	X			0				x	x							x						(NOP)
1	0	0	0	0	0	1	0	0	0	0	x	x	x	x	1	0	1	0	1	0	0	0	x	x	BZ
1	0	0	0	1			0				x	x							1						BNZ
1	0	0	1	0			0				x	0							1						MOVI
1	0	0	1	1	0	0	0	0	0	1	x	0	0	0	1	0	0	1	1	0	1	0	1	0	MOVHI
1	0	1	0	0			0				x	x							x						IN
1	0	1	0	1			0				x	x							x						OUT
1	0	1	1	X			0				x	x							x						(NOP)
1	1	X	X	X			0				x	x							x						(NOP)

Criterio de corrección: -0.25 puntos por cada fila y columna incorrecta, escogiendo el número mínimo de filas y/o columnas que cubren todos los errores.

Ejercicio 8 (Objetivos X) (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)

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

