

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

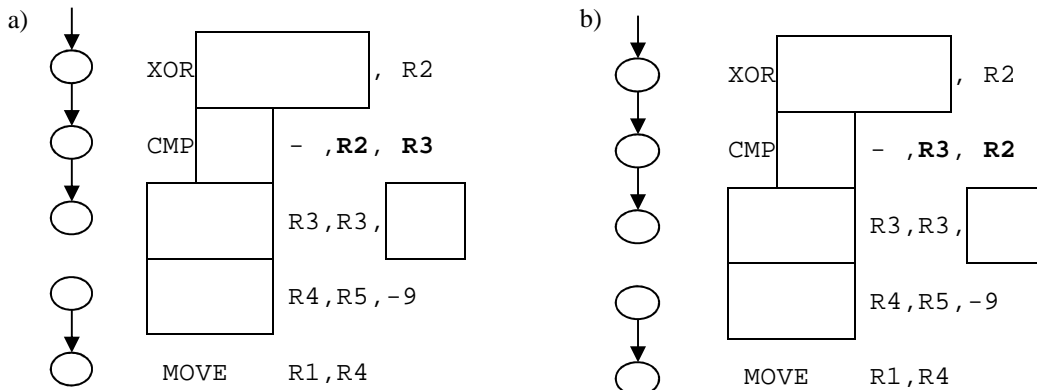
Examen 3. (Temas 8, 9, 10 y 11)

- 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 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 / 32;
} else {
    R4 = R5 - 9;
}
R1 = R4;
```

**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	W/D	N (hexa)
SUBI R1, R2, -3									
MOVEI R3, 0x00B7									
OUT R4 // IN R3 // XORI -, R4, 9									

- 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	W/D	N (hexa)
	xxx	xxx	x	xx	xxx	1	000	1	X X X X
	100	xxx	0	00	000	x	xxx	0	A B C D
	000	xxx	x	00	011	0	000	1	X X X X

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

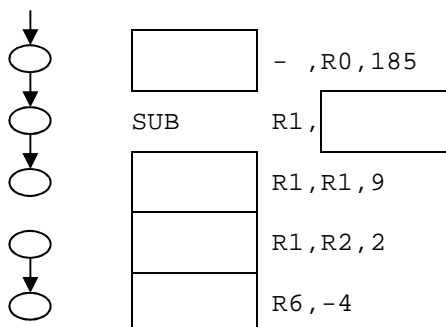
```
if (R0<=185) { R1=R0-R1+9; }
else { R1=R2-2; }
R6=-4;
```

Fragmento 2

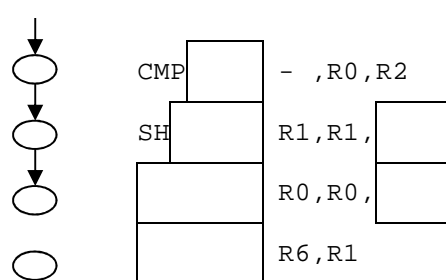
```
while (R0>R2) {
    R1=R1/4;
    R0=R0-3;
}
R6=not(R1);
```

- 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)



- b) Completad los fragmentos de programa en lenguaje ensamblador SISA-I 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-I 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	_____ R7, 0xB9
0x0001	MOV_____ R7, _____
0x0002	CMP_____ R7, R0, R7
0x0003	B_____ R7, _____
0x0004	SUB R1, _____
0x0005	_____ R1, R1, 9
0x0006	B_____ R7, _____
0x0007	_____ R1, R2, _____
0x0008	_____ R6, -4

Fragmento 2 (0.5 puntos)

@I-Mem	
0x0000	CMP_____ R7, R0, R2
0x0001	B_____ R7, _____
0x0002	MOVI R7, _____
0x0003	S_____ R1, R1, _____
0x0004	ADDI R0, R0, _____
0x0005	B_____ R7, _____
0x0006	_____ R6, R1

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-I	Lenguaje ensamblador SISA-I
0x616C	
	IN R6, 128
0x3DFE	

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

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 (UCG+UPG+subsistema IO+Memoria) durante el ciclo en que se ejecuta cada una de las instrucciones SISA-I 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 y salida y de la memoria de datos es cero.

Instrucción SISA-I	Palabra de Control del SISC Harvard uniciclo														
	@A	@B	Rb/N	OP	F	-i/l/a	@D	WrD	Wr-Out	Rd-In	Wr-Mem	Byte	TknBr	N (hexa)	ADDR-IO (hexa)
MOVHI R3, -2															
STB 6(R2), R0															
BNZ R1, -4															
IN R4, 198															
LD R2,3(R0)															

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 0x34 y el de todas las posiciones impares de la memoria de datos es 0x56. 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
LDB R5, -1(R4)	
STB 6(R0), R3	
BNZ R7, 6	
SHL R4, R2, R0	
LD R4, 0(R2)	

Ejercicio 7 (Objetivos X) (1.5 puntos)

Indica el contenido de la tabla de la ROM (sólo las celdas sombreadas) 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/l/a1	-i/l/a0	OP ₁	OP ₀	MxN1	MxN0	MxF	f2	f1	f0	MxD1	MxD0	
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																					-
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																					-
1	1	X	X	X																					-

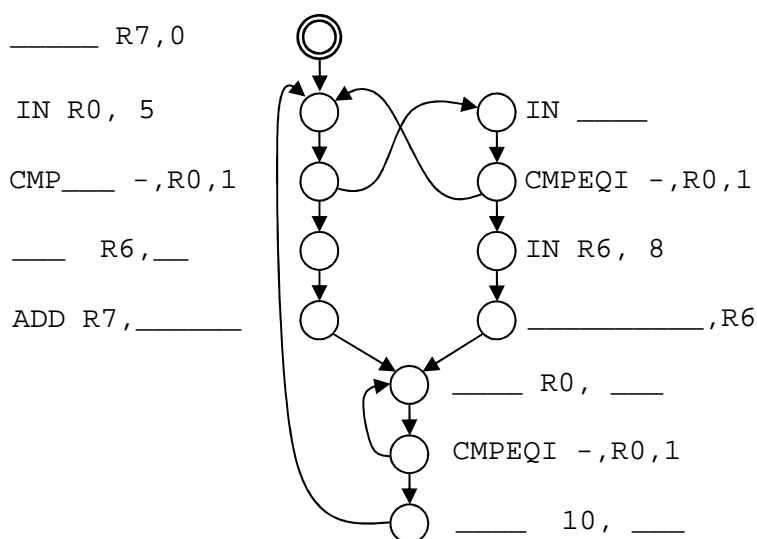
Ejercicio 8 (Objetivos X) (1.5 puntos)

Se han conectado a la UPG dos dispositivos externos de entrada (*dispA* y *dispB*) que nos envían valores enteros y un dispositivo externo de salida (*dispIMP*) al cual le podemos enviar datos. Todos estos dispositivos tienen un efecto lateral sobre su registro de estado.

Se desea realizar una función que vaya leyendo los datos de ambos dispositivos en el orden en el que se vayan detectando. Partiremos de un registro inicializado a cero, cada vez que llegue un valor por el *dispA* se le sumará a este registro y cada vez que llegue un valor por el *dispB* se le restará. A cada actualización del valor de este registro se enviará al *dispIMP*.

Suponiendo que el dispositivo *dispA* tiene el registro de status en la dirección 5 del espacio de direccionamiento de entrada y el de datos en la 6; que el *dispB* tiene el registro de status en la dirección 7 y el de datos en la 8; y que el *dispIMP* 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:

a) 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) el fragmento de código ensamblador SISA-I para que realice la función anteriormente descrita. (0.75 puntos)

```

MOVI R7, 0
IN R0, 5
__ R0, __
IN R6, 6
ADD R7, R7, R6
B__ R0, __
IN R0, 7
B__ R0, __
IN R6, 8
SUB R7, R7, R6
__ R0, 9
B__ R0, __
__ 10, R7
B__ R0, __

```