

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

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

- Duración del examen: 1 hora y 45 minutos.
- 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 6 de diciembre.

**Ejercicio 1 (0.5 puntos)**

Indicad para si es cierta o falsa cada una de las siguientes afirmaciones. (nota: Las respuestas incorrectas restaran sobre las correctas).

	Verdadero	Falso
1 - Los programas/grafos con secuenciamiento implícito tienen menos instrucciones/estados que con secuenciamiento explícito	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2 - Las unidades de control con secuenciamiento implícito necesitan un bloque que incremente el valor del PC	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3 - Las unidades de control con secuenciamiento implícito no pueden saltar a la misma instrucción que están ejecutando como si lo pueden hacer las unidades de control con secuenciamiento explícito	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4 - Las unidades de control con secuenciamiento implícito absoluto necesitan más espacio de memoria para codificar las instrucciones de salto que las unidades de control con secuenciamiento implícito relativo	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5 - La codificación en formato compacto sirve para que los programas ocupen menos espacio en la memoria	<input checked="" type="checkbox"/>	<input type="checkbox"/>

**Criterio de corrección: +0.1 puntos por cada respuesta correcta y -0.1 por cada respuesta incorrecta.****Ejercicio 2 (1 punto)**

- a) Indica el valor que debe tener cada uno de los bits de la palabra de control de la UPG básica (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)
OUT R5 // MOVEI R1, 0x1234	101	xxx	0	10	001	0	001	1	1 2 3 4
CMPLTU -,R4, R1 // IN R3	100	001	1	01	100	1	011	1	x x x x

**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 básica (sin subsistema de I/O ni memoria). (0.5 puntos)

Mnemotécnico	@A	@B	Rb/N	OP	F	In/Alu	@D	WrD	N (hexa)
CMPLTUI -,R2, 0x5678	010	xxx	0	01	101	x	xxx	0	5 6 7 8
SHA R2, R1, R3	001	011	1	00	110	0	010	1	x x x x

**Criterio de corrección: -0.25 puntos por cada mnemotécnico incorrecto.****Ejercicio 3 (1 punto)**

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.

Lenguaje máquina SISA	Lenguaje ensamblador SISA
0x01CA	XOR R1, R0, R7
0x9780	MOVHI R3, -128 / MOVHI R3, 0x80
0x0AD5	SUB R2, R5, R3
0x4AFE	ST -2 (R5), R3

**Criterio de corrección: +0.25 puntos por cada fila correcta.**



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

**Ejercicio 5 (1 punto)**

Escribid sobre la siguiente tabla el valor de los bits que tiene la palabra de control del SISC-Harvard unicycle (incluyendo la señal *TknBr*) durante el ciclo en que se ejecuta cada una de las instrucciones SISA. Indicad únicamente el valor (0 o 1) de los bits que son estrictamente necesarios para ejecutar correctamente cada instrucción. Para el resto de bits de la palabra de control, que pueden valer 0 o 1 indistintamente para la ejecución correcta de la instrucción, ponad x (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 unicio														N (hexa)	ADDR-IO (hexa)
	@A	@B	Rb/N	OP	F	-/i//a	@D	WrD	Wr-Out	Rd-In	Wr-Mem	Byte	TknBr			
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	
ADDI R1, R2, 5	010	xxx	0	00	100	00	001	1	0	0	0	x	0	0005	XX	
OUT 44, R1	001	xxx	x	xx	xxx	xx	xxx	0	1	0	0	x	0	XXXX	2C	

**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 (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 <sub>15</sub>	I <sub>14</sub>	I <sub>13</sub>	I <sub>12</sub>	I <sub>8</sub>	Bnz	Bz	Wr-Mem	Rd-In	Wr-Out	WrD	Byte	Rb/N	-i//a <sub>1</sub>	-i//a <sub>0</sub>	OP <sub>1</sub>	OP <sub>0</sub>	MxN <sub>1</sub>	MxN <sub>0</sub>	MxF	f <sub>2</sub>	f <sub>1</sub>	f <sub>0</sub>	MxD <sub>1</sub>	MxD <sub>0</sub>	
0	0	0	0	X						1	X		0	0					0						A / L
0	0	0	1	X	0	0	0	0	0	1	X	1	0	0	0	1	X	X	0	X	X	X	0	0	CMP
0	0	1	0	X						1	X		0	0					1						ADDI
0	0	1	1	X						1	0		0	1					1						LD
0	1	0	0	X						0	0		X	X					1						ST
0	1	0	1	X	0	0	0	0	0	1	1	0	0	1	0	0	0	0	1	1	0	0	0	1	LDB
0	1	1	0	X						0	1		X	X					1						STB
0	1	1	1	X						0	X		X	X					X						(NOP)
1	0	0	0	0						0	X		X	X					1						BZ
1	0	0	0	1	1	0	0	0	0	0	X	X	X	X	1	0	1	0	1	0	0	0	X	X	BNZ
1	0	0	1	0						1	X		0	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						1	X		1	0					X						IN
1	0	1	0	1						0	X		X	X					X						OUT
1	0	1	1	X						0	X		X	X					X						(NOP)
1	1	X	X	X						0	X		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 7 (1 punto)**

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 0x8C36, el contenido de todos los registros es 0xFFDA y que el contenido de todas las posiciones pares de la memoria de datos es 0x83 y el de todas las posiciones impares de la memoria de datos es 0x24. Utiliza el mnemotécnico MEM<sub>b</sub>[...], MEM<sub>w</sub>[...] 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=0x22DA	PC=0x8C38
LDB R2, 4 (R6)	R2=0xFF83	PC=0x8C38
SHA R3, R3, R3	R3=0xFFFF	PC=0x8C38
STB -3 (R6), R2	MEM <sub>b</sub> [0xFFD7]=0xDA	PC=0x8C38

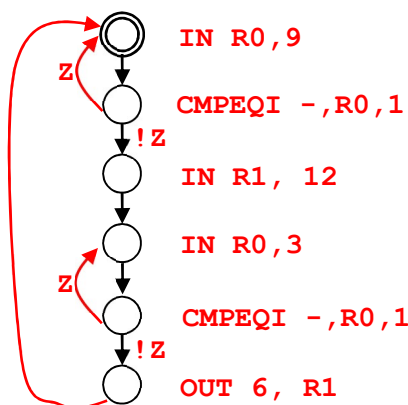
**Criterio de corrección:** -0.25 puntos por cada fila incorrecta sin contar los PC que siguen el secuenciamiento implícito. Si los PC que siguen el secuenciamiento implícito están mal sólo descuentan -0.25 puntos adicionales.

**Ejercicio 8 (1 punto)**

Se ha conectado a la UPG un dispositivo externo de entrada que nos envía valores y que tiene el registro de status en la dirección 9 del espacio de direccionamiento de entrada y el de datos en la 12. También se ha conectado un dispositivo externo de salida que tiene el registro de status en la dirección 3 del espacio de direccionamiento de entrada y el de datos en la dirección 6 del espacio de direccionamiento de salida. Ambos dispositivos tienen un efecto lateral en la lectura/escritura del dato sobre su registro de estado.

Se desea que nuestro sistema vaya leyendo indefinidamente los datos que se reciban por el dispositivo de entrada y se los envíe al dispositivo de salida. Nota: No hay que hacer ninguna operación con el valor recibido.

- a) Dibuja el grafo de estados si estuviésemos utilizando una unidad de control específica (UCe) junto a la UPG (con entrada/salida y memoria) para que realice la función anteriormente descrita. (0.5 punto)



- b) Usando el procesador SISC Harvard unificado, escribid el programa en código ensamblador SISA para que realice la función anteriormente descrita. (0.5 puntos)

```
IN  R0, 9
BZ  R0, -2
IN  R1, 12
IN  R0, 3
BZ  R0, -2
OUT 6, R1
BNZ R0, -7
```

**Criterio de corrección:** -0.25 puntos por cada instrucción incorrecta. La primera instrucción incorrecta no descontará ningún punto.

**Criterio de corrección:** -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.

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 9 (1 punto)**

Se ha encargado fabricar una remesa de procesadores SISC Harvard unificado (UPG+I/O+MEM) y ha salido defectuosa. El multiplexor controlado por la señal F del bloque MISC que está en la ALU no funciona correctamente para las entradas 1 y 2. Ello imposibilita que puedan ejecutar correctamente las instrucciones MOVI y MOVHI. Deseamos ejecutar este código en estos procesadores:

```
...
MOVI  R0, 0xCD
MOVHI R0, 0xAB
SUB   R2, R0, R1
...
```

Escribid un fragmento de código ensamblador SISA equivalente que ejecute la misma tarea, pero sin usar las instrucciones MOVI y MOVHI que no funcionan correctamente. Podéis usar el resto de registros para almacenar valores temporales si los necesitáis.

Hay varias soluciones correctas. Aquí tenéis algunas como ejemplo.

Solución 1:

```
XOR  R0, R0, R0
ADDI R7, R0, 4
ADDI R0, R0, 0x0A
SHL  R0, R0, R7
ADDI R0, R0, 0x0B
SHL  R0, R0, R7
ADDI R0, R0, 0x0C
SHL  R0, R0, R7
ADDI R0, R0, 0x0D
SUB  R2, R0, R1
```

Solución 2:

```
XOR  R0, R0, R0
ADDI R7, R0, 5
ADDI R0, R0, 0x2A
SHL  R0, R0, R7
ADDI R0, R0, 0x1E
SHL  R0, R0, R7
ADDI R0, R0, 0x0D
SUB  R2, R0, R1
```

Solución 3:

```
XOR  R0, R0, R0
ADDI R7, R0, 6
ADDI R0, R0, 0x0B
SHL  R0, R0, R7
ADDI R0, R0, 0x2F
SHL  R0, R0, R7
ADDI R0, R0, 0x0D
SUB  R2, R0, R1
```

**Criterio de corrección:** binario. Sólo es correcto si el fragmento de código escrito realiza la función deseada. Excepción: Si todo el fragmento de código es correcto, pero falta la última instrucción (SUB) se contarán la mitad de puntos