

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 restarán sobre las correctas).

	Verdadero	Falso
1 - Los programas/grafos con secuenciamiento implícito tienen menos instrucciones/estados que con secuenciamiento explícito	<input type="checkbox"/>	<input type="checkbox"/>
2 - Las unidades de control con secuenciamiento implícito necesitan un bloque que incremente el valor del PC	<input 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 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 type="checkbox"/>	<input type="checkbox"/>
5 - La codificación en formato compacto sirve para que los programas ocupen menos espacio en la memoria	<input type="checkbox"/>	<input type="checkbox"/>

**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)

<b>Mnemotécnico</b>	@A	@B	Rb/N	OP	F	In/Alu	@D	WrD	N (hexa)
OUT R5 // MOVEI R1, 0x1234									
CMPLTU -,R4, R1 // IN R3									

- 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)

<b>Mnemotécnico</b>	@A	@B	Rb/N	OP	F	In/Alu	@D	WrD	N (hexa)
	010	xxx	0	01	101	x	xxx	0	5 6 7 8
	001	011	1	00	110	0	010	1	X X X X

**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.

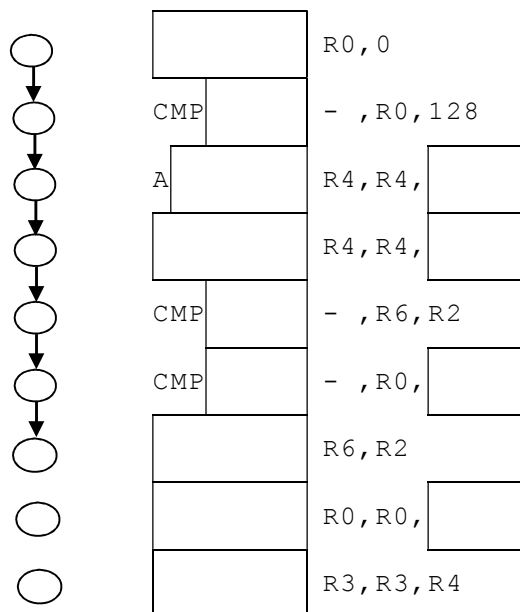
Lenguaje máquina SISA	Lenguaje ensamblador SISA
0x01CA	
0x9780	
	SUB R2, R5, R3
	ST -2 (R5), R3

**Ejercicio 4 (2 puntos)**

Dado el siguiente fragmento de código en C (el código no tiene que hacer algo útil), indicad como se implementarían 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**.

```
for (R0=0; R0<128; R0++) {
    R4=R4*2+R5;
    if ((R6>=R2) && (R0!=0))
        R6=not (R2)
}
R3=R3-R4;
```

- a) Completad el fragmento 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.



- b) Completad el fragmento 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.

@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	CMP_____ R7, <b>R2, R6</b>
0x000E	B_____ R7, _____
0x0010	B_____ R0, _____
0x0012	_____ R6, R2
0x0014	_____ R0, R0, _____
0x0016	_____ R0, _____
0x0018	_____ R3, R3, R4

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. Indica 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, poned 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 unicycle														
	@A	@B	Rb/N	OP	F	-i/i/a	@D	WrD	Wr-Out	Rd-In	Wr-Mem	Byte	TknBr	N (hexa)	ADDR-IO (hexa)
ST 0(R4), R0															
BZ R1, -6															
ADDI R1, R2, 5															
OUT 44, R1															

**Ejercicio 6 (1.5 puntos)**

Indica el contenido de la tabla de la ROM (sólo las celdas en blanco) correspondiente al bloque ROM\_CRTL\_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/l/a <sub>1</sub>	-i/l/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																					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 7 (1 punto)**

Indica 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	
LDB R2, 4 (R6)	
SHA R3, R3, R3	
STB -3 (R6), R2	

**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.

- |   |  |
|---|--|
| <p>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 puntos)</p> | <p>b) Usando el procesador SISC Harvard unicycle, escribid el programa en código ensamblador SISA para que realice la función anteriormente descrita. (0.5 puntos)</p> |
|---|--|

**Ejercicio 9 (1 punto)**

Se ha encargado fabricar una remesa de procesadores SISC Harvard unicycle (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 se 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.