

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 (0.5 puntos)

Raona els avantatges i inconvenients de disposar d'una unitat de control amb seqüenciament implícit davant d'una unitat de control amb seqüenciament explícit. Marca amb una creu totes les afirmacions correctes (nota: les respostes incorrectes restaran nota sobre les correctes)

- ☐ Les unitats de control amb seqüenciament implícit no poden saltar a la mateixa instrucció que estan executant com si poden fer-ho les unitats de control amb seqüenciament explícit.
- ☐ Els programes/grafs amb seqüenciament implícit tenen menys instruccions/estats que amb seqüenciament explícit.
- ☒ Per fer programes/grafs amb seqüenciament implícit necessitem noves instruccions especials.
- ☒ En general, els programes/grafs amb seqüenciament implícit ocupen menys espai a la memòria ROM.
- ☒ Les unitats de control amb seqüenciament explícit necessiten disposar d'un registre que contingui l'adreça de la ROM on està ubicada la instrucció que s'està executant.
- ☐ Les unitats de control amb seqüenciament explícit executen les instruccions en l'ordre que estan escrites a memòria.
- ☒ Les unitats de control amb seqüenciament implícit incrementen automàticament el registre PC amb el nombre de posicions de memòria que ocupa una instrucció quan executen una instrucció que no trenqui la seqüència.
- ☐ Les unitats de control amb seqüenciament implícit son capaces de fer més tasques que les unitats amb seqüenciament explícit.
- ☐ Les unitats de control amb seqüenciament explícit son capaces de fer més tasques que les unitats amb seqüenciament implícit.
- ☒ Les unitats de control amb seqüenciament implícit necessiten d'un bloc que incrementi el valor PC.

Criterio de corrección: +0.05 puntos por cada respuesta correcta y -0.05 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 (sin subsistema de I/O ni memoria) para que realice, durante un ciclo, la acción concreta especificada mediante el mnemotécnico. Indica 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)

<i>Mnemotécnico</i>	@A	@B	Rb/N	OP	F	In/Alu	@D	WrD	N (hexa)
MOVEI R0, 0xABCD	xxx	xxx	0	10	001	0	000	1	A B C D
IN R7 // XORI -, R4, -6	100	xxx	0	00	010	1	111	1	F F F A
CMPLEU R3, R1, R0	001	000	1	01	101	0	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 (sin subsistema de I/O ni memoria). (0.5 puntos)

<i>Mnemotécnico</i>	@A	@B	Rb/N	OP	F	In/Alu	@D	WrD	N (hexa)
ADD R3, R4, R1	100	001	1	00	100	0	011	1	X X X X
CMPEQI -, R7, 10	111	xxx	0	01	011	x	xxx	0	0 0 0 A
IN R0 // NOT -, R5	101	xxx	x	00	011	1	000	1	X X X X

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

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

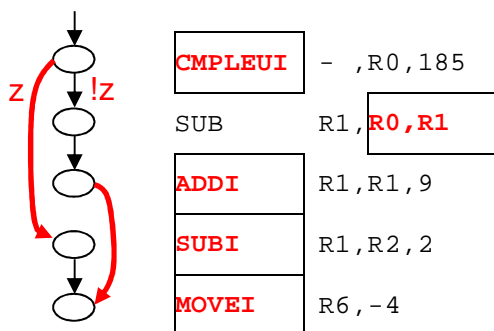
```
if (R0<=185) { R1=R0-R1+9; }
else { R1=R2-2; }
R6=-4;
```

Fragmento 2

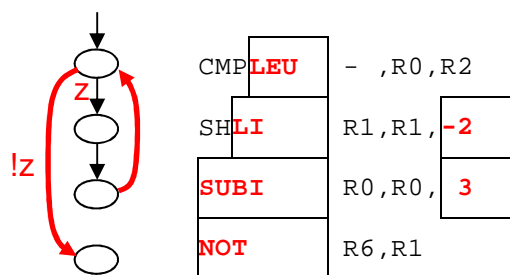
```
while (R0>R2) {
    R1=R1/4;
    R0=R0-3;
}
R6=not(R1);
```

- 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)



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 R7, 0xB9
0x0002	MOV HI R7, 0x00
0x0004	CMP LEU R7, R0, R7
0x0006	B Z R7, 3
0x0008	SUB R1, R0, R1
0x000A	ADDI R1, R1, 9
0x000C	BNZ R7, 1
0x000E	ADDI R1, R2, -2
0x0010	MOVI R6, -4

Fragmento 2 (0.5 puntos)

@I-Mem	
0x0000	CMP LEU R7, R0, R2
0x0002	BNZ R7, 4
0x0004	MOVI R7, -2
0x0006	SHL R1, R1, R7
0x0008	ADDI R0, R0, -3
0x000A	BNZ R7, -6
0x000C	NOT R6, R1

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.

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.

Lenguaje máquina SISA	Lenguaje ensamblador SISA
0x022E	SHA R5, R1, R0
0x8AFD	BZ R5, -3
0x3877	LD R1, -9(R4)

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

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

Ejercicio 5 (0.75 puntos)

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, 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														
	@A	@B	Rb/N	OP	F	-i/l/a	@D	WrD	Wr-Out	Rd-In	Wr-Mem	Byte	TknBr	N (hexa)	ADDR-IO (hexa)
STB 3(R2), R5	010	101	0	00	100	xx	xxx	0	0	0	1	1	0	0003	XX
BNZ R7, 11	111	xxx	x	10	000	xx	xxx	0	0	0	0	x	0	0016	XX
MOVHI R1, -2	001	xxx	0	10	010	00	001	1	0	0	0	x	0	XXFE*	XX

* También se considerará correcto haber puesto FFFE

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 (0.75 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 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_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
BNZ R5, 6	PC=0x8C44
LDB R4, 8(R5)	R4=0xFF83 PC=0x8C38
MOVHI R7, 3	R7=0x03DA 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 7 (1.5 puntos)

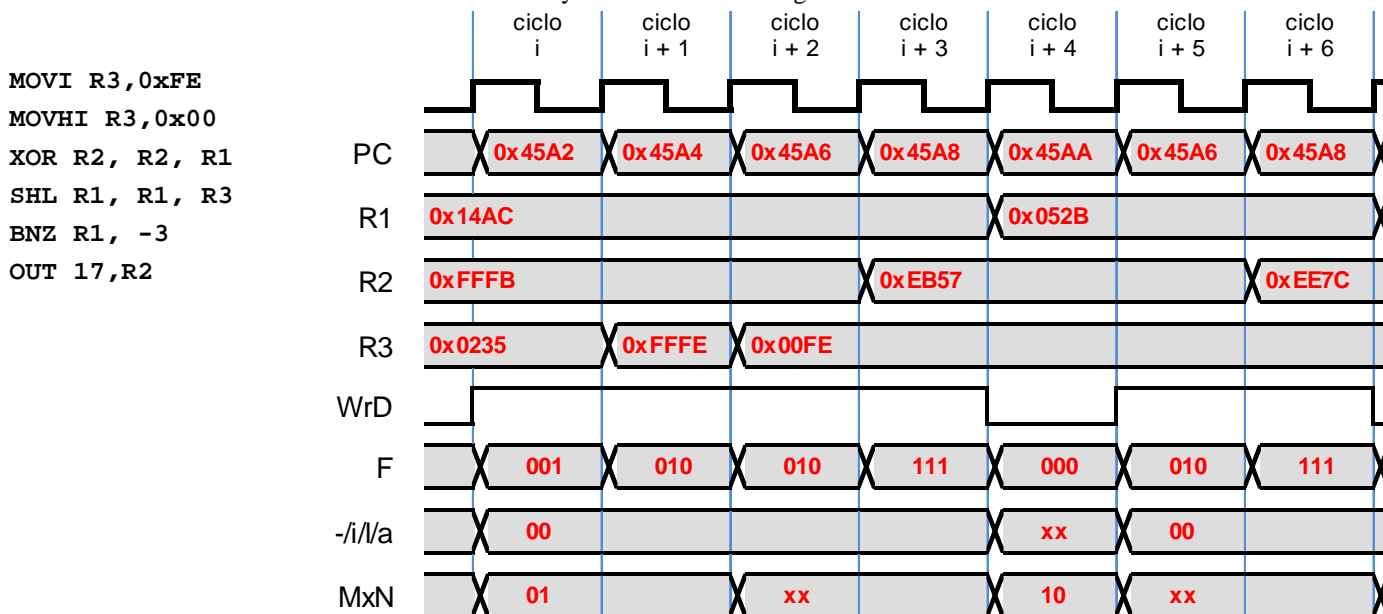
Indica el contenido de la tabla de la ROM (sólo las celdas en blanco) correspondiente al bloque ROM_CRTL_LOGIC. Indica los valores que tomarían las señales para ejecutar correctamente las instrucciones. Indica con x los valores de los bits del contenido de la ROM que puedan valer 0 o 1.

Dirección ROM					Contenido de la ROM																				
I ₁₅	I ₁₄	I ₁₃	I ₁₂	I ₈	Bnz	Bz	Wr-Mem	Rd-In	Wr-Out	WrD	Byte	Rb/N	-i/l/a ₁	-i/l/a ₀	OP ₁	OP ₀	MxN ₁	MxN ₀	MxF	f ₂	f ₁	f ₀	MxD ₁	MxD ₀	
0	0	0	0	X							x	1							0				0	0	A / L
0	0	0	1	X							x	1							0				0	0	CMP
0	0	1	0	X							x	0							1				0	1	ADDI
0	0	1	1	X							0	0							1				0	1	LD
0	1	0	0	X	0	0	1	0	0	0	0	0	x	x	0	0	0	0	1	1	0	0	x	x	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							1	0							1				x	x	STB
0	1	1	1	X							x	x							x				x	x	(NOP)
1	0	0	0	0							x	x							1				x	x	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	0	0	0	0	0	1	x	0	0	0	1	0	0	1	1	0	0	1	1	0	MOVI
1	0	0	1	1							x	0							1				1	0	MOVHI
1	0	1	0	0							x	x							x				1	0	IN
1	0	1	0	1							x	x							x				x	x	OUT
1	0	1	1	X							x	x							x				x	x	(NOP)
1	1	X	X	X							x	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 8 (1 punto)

Dado el siguiente fragmento de código donde en el *ciclo i* se ejecuta la instrucción `MOVI R3, 0xFE`, rellena el siguiente cronograma indicando el valor de las señales de la UCG o UPG y los valores de los registros.



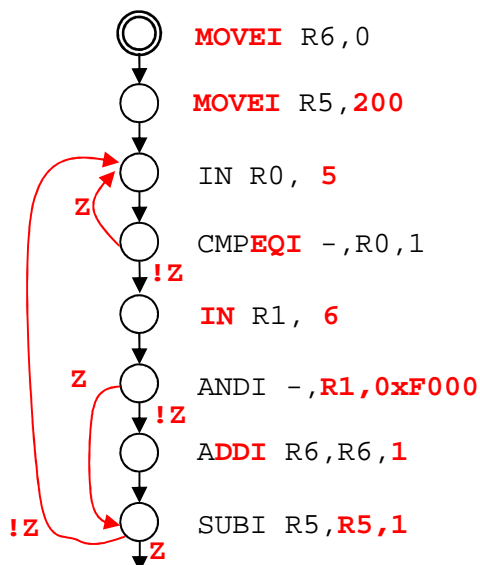
Criterio de corrección: -0.25 puntos por cada fila incorrecta.

Ejercicio 9 (2 puntos)

Se ha conectado a la UPG un dispositivo externo de entrada que nos envía valores **naturales 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 200 valores y se desea saber cuántas operaciones de multiplicación darían overflow si cada dato recibido lo multiplicásemos por el valor 16. El resultado de esta cuenta debe dejarse en el registro R6.

- a) 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. (1 punto)



- b) Usando el procesador SISC Harvard uniciclo, completad el fragmento de código ensamblador SISA para que realice la función anteriormente descrita. Una vez finalizada la operación, el programa deberá dejar el resultado de 16 bits en la posición 0xF006 de la memoria de datos (1 punto)

```

MOVI R6, 0
MOVI R5, 0xC8
MOVHI R5, 0x00
IN R0, 5
BZ R0, -2
IN R1, 6
MOVI R2, 0x00
MOVHI R2, 0xF0
AND R3, R1, R2
BZ R3, 1
ADDI R6, R6, 1
ADDI R5, R5, -1
BNZ R5, -10
ST 6(R2), R6
  
```

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.