

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 3 de diciembre.

Ejercicio 1 (0,5 puntos)

Completa el siguiente fragmento de código ensamblador SISA para el procesador SISC Harvard unicycle (UPG+I/O+MEM) para que guarde en el registro R0 el byte almacenado en la posición de memoria 0x3456.

```

...
MOVI  R1, 0x58
MOVHI R1, 0x34
LDB    R0, 0xFE (R1)
...

```

Criterio de corrección: Binario: 0.5 puntos si el fragmento de código es correcto

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. 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)

Mnemotécnico	@A	@B	Rb/N	OP	F	In/Alu	@D	WrD	N (hexa)
IN R1 // OUT R5 // ADDI -, R5, 0x1234	101	xxx	0	00	100	1	001	1	1 2 3 4
ADD R0, R3, R7 // OUT R3	011	111	1	00	100	0	000	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)
SUB R3, R1, R7	001	111	1	00	101	0	011	1	x x x x
CMPLTI -, R2, 0xFFFC // IN R4	010	xxx	0	01	000	1	100	1	F F F C

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
0x477D	ST -3(R3), R5
0xA1FC	OUT 252, R0
0x99FF	MOVHI R4, -1
0x86FB	BZ R3, -5

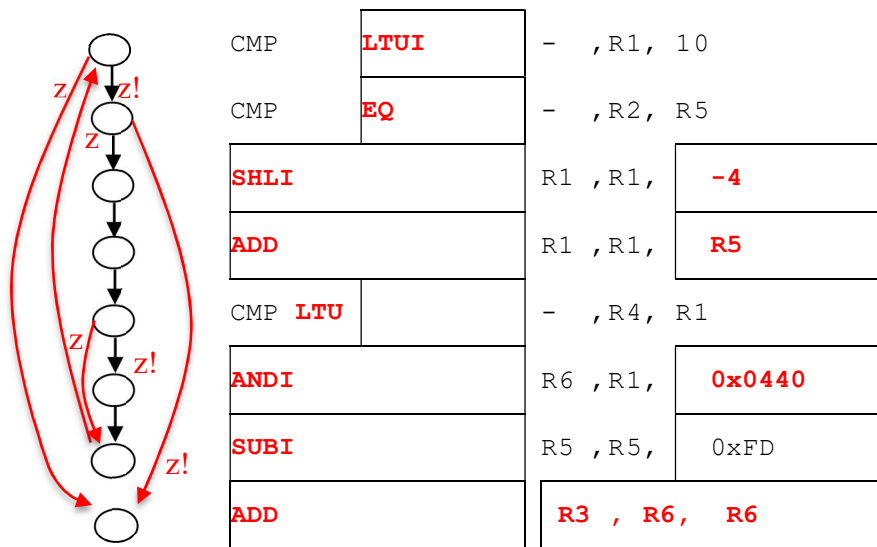
Criterio de corrección: +0.25 puntos por cada fila correcta.

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**.

```
while ((R1<10) && (R2!=R5)) {
    R1=R1/16+R5;
    if (R1>R4)
        R6=AND(R1,0x0440);
    R5 = R5+3;
}
R3= R6 + R6;
```

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. (1 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. Las funcionalidades que sean iguales en ambos apartados y estén mal en ambos, sólo contarán como un error.

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. (1 punto)

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

@I-Mem	
0x0000	MOVI R7, 10
0x0002	CMP <u>LTU</u> R7, R1, R7
0x0004	BZ R7, 12
0x0006	CMP <u>EQ</u> R7, R2, R5
0x0008	BNZ R7, 10
0x000A	MOVI R7, -4
0x000C	SHL R1, R1, R7
0x000E	ADD R1, R1, R5
0x0010	CMP <u>LEU</u> R7, R1, R4
0x0012	BNZ R7, 3
0x0014	MOVI R7, 0x40
0x0016	MOVHI R7, 0x04
0x0018	AND R6, R1, R7
0x001A	ADDI R5, R5, 3
0x001C	BNZ R7 , -15
0x001E	ADD R3, R6 , R6

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 uniclo (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 del SISC Harvard uniclo														
	@A	@B	Rb/N	OP	F	-/i//a	@D	WrD	Wr-Out	Rd-In	Wr-Mem	Byte	TknBr	N (hexa)	ADDR-IO (hexa)
STB -3(R6), R4	110	100	0	00	100	xx	xxx	0	0	0	1	1	0	FFFD	XX
BZ R0, -13	000	xxx	x	10	000	xx	xxx	0	0	0	0	x	1	FFE6	XX
OUT 0x23, R1	001	xxx	x	xx	xxx	xx	xxx	0	1	0	0	x	0	XXXX	23
CMPEQ R3, R3, R0	011	000	1	01	011	00	011	1	0	0	0	x	0	XXXX	XX

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 puntos)

Indica el contenido de la tabla de la ROM (sólo filas indicadas) 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//a ₁	-/i//a ₀	OP ₁	OP ₀	MxN ₁	MxN ₀	MxF	f ₂	f ₁	f ₀	MxD ₁	MxD ₀	
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	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
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	1	0	0	0	0	0	1	x	0	0	0	1	0	0	1	1	0	1	0	1	0	MOVHI

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 0x2A34, el contenido de todos los registros es 0xC328 y que el contenido de todas las posiciones pares de la memoria de datos es 0x15 y el de todas las posiciones impares de la memoria de datos es 0x73. Assumid todos los registros de E/S contienen el valor 0x0001. 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
XOR R2, R1, R5	R2=0x0000 PC=0x2A36
LDB R3, 6(R5)	R3=0x0015 PC=0x2A36
IN R3, 11	R3=0x0001 PC=0x2A36
MOVHI R5, 0x66	R5=0x6628 PC=0x2A36

Criterio de corrección: -0.25 puntos por cada fila incorrecta. Error en PC solo se cuenta una vez.

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 4 del espacio de direccionamiento de entrada y el de datos en la 8. Este dispositivo tiene un efecto lateral en la lectura/escritura del dato sobre su registro de estado.

Usando el procesador UCG+UPG+IO+MEM (visto en clase), escribid el programa en código ensamblador SISA para que realice la función anteriormente descrita. Usad el registro R1 para consultar el estado del dispositivo y el registro R0 para recibir datos. Usad los registros R3 y R4 para almacenar las constantes 0x20 y 0xFFFFE respectivamente. Usad el registro R5 para cualquier cálculo temporal. **Nota:** asumir las siguientes instrucciones como iniciales del programa, completad el programa a partir de ellas.

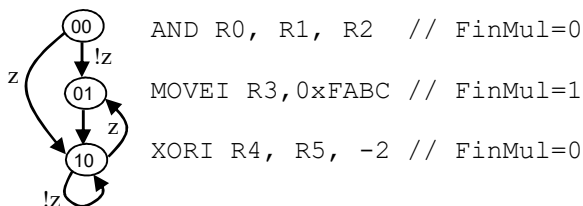
```

MOVI R3, 0x20
MOVI R4, 0xFE
MOVI R2, 0x01
IN R1, 4
BZ R1, -2
IN R0, 0x08
BZ R0, -1
AND R5, R0, R2
BZ R5, 2
STB 0(R0), R3
BNZ R5, -8
ST 0(R0), R4
BZ R5, -10

```

Criterio de corrección: 0,25 E/S correcta. 0,25 bucles correctos, 0,25 ST/STB correctos, 0,25 uso de la máscara correcto.

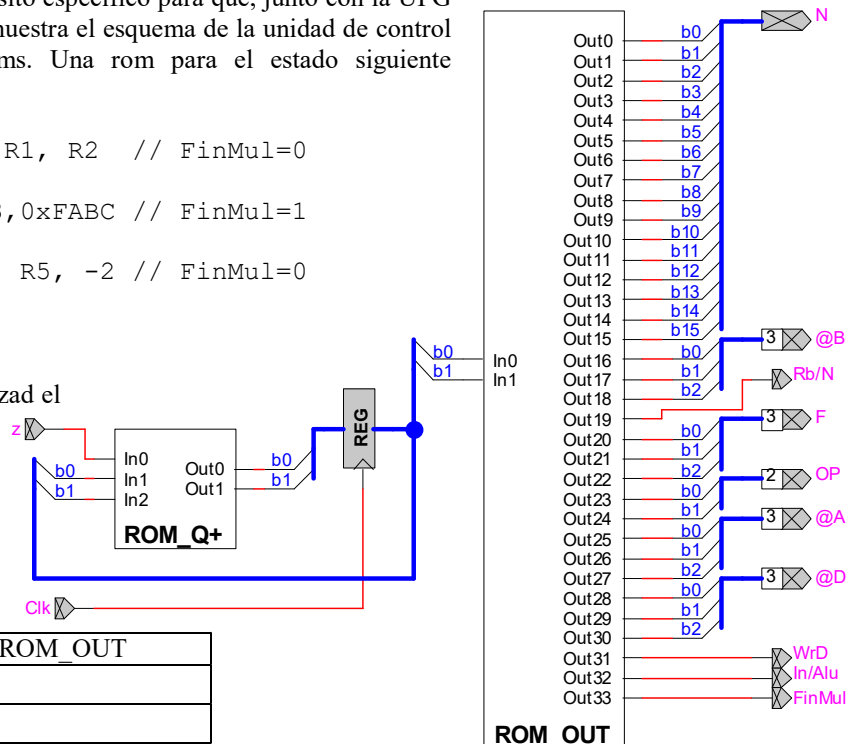
Dado el siguiente grafo de estados de una UC de propósito específico para que, junto con la UPG básica, formen un PPE que lo ejecute. En la figura se muestra el esquema de la unidad de control correspondiente a una implementación con dos roms. Una rom para el estado siguiente (ROM Q+) y otra rom para las salidas (ROM OUT).



Indica el contenido de las ROMS en hexadecimal. Utilizad el valor 0 para los bits que sean x. Si los contenidos no están en hexadecimal no se corregirá el ejercicio.

@ROM	Contenido ROM_Q+
@0	1
@1	2
@2	2
@3	2
@4	2
@5	1
@6	0
@7	0

@ROM	Contenido ROM_OUT
@0	0x0820A0000
@1	0x2B110FABC
@2	0x0CA20FFFE
@3	0x000000000



Criterio de corrección: 0.50 ROM Q+, 0,50 ROM_OUT. Para cada una, -0,25 por cada fila incorrecta.