

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 XX de diciembre.

Ejercicio 1 (1 punto)

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, _____
MOVHI R1, _____
LDB   R0, _____
...

```

En el procesador SISC Harvard unicycle se puede acceder a una misma posición de memoria de varias maneras. ¿De cuantas maneras distintas hubiésemos podido completar el código anterior (teniendo en cuenta que todos los valores numéricos siempre están escritos en hexadecimal) para que realizase el acceso al byte de la posición 0x3456? Indicad el número total y justificadlo.

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)

<i>Mnemotécnico</i>	@A	@B	Rb/N	OP	F	In/Alu	@D	WrD	N (hexa)
IN R1 // OUT R5 // MOVEI -, 0x1234									
SHL R2, R3, R1 // OUT 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)

<i>Mnemotécnico</i>	@A	@B	Rb/N	OP	F	In/Alu	@D	WrD	N (hexa)
	001	111	1	00	101	0	011	1	X X X X
	010	xxx	0	01	000	1	100	1	F F F C

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.

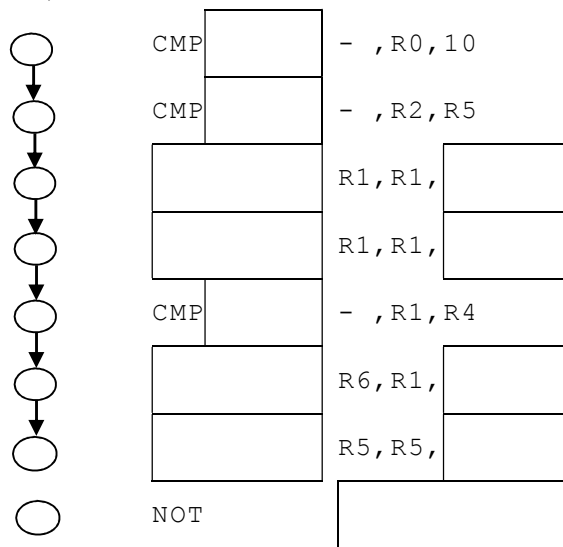
Lenguaje máquina SISA	Lenguaje ensamblador SISA
0x677D	
0xA0FC	
	MOVHI R4, 24
	BNZ R2, -5

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 ((R0<10) || (R2!=R5)) {
    R1=R1/16+R5;
    if (R1>R4)
        R6=XOR(R1,0x0440);
    R5--;
}
R3=not(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)



- 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)

@I-Mem	
0x0000	_____ R7, 10
0x0002	CMP_____ R7, R0, R7
0x0004	B_____ R7, _____
0x0006	CMP_____ R7, R2, R5
0x0008	B_____ R7, _____
0x000A	_____ R7, _____
0x000C	_____ R1, R1, _____
0x000E	ADD R1, R1, _____
0x0010	CMP_____ R7, R1, R4
0x0012	B_____ R7, _____
0x0014	MOVI R7, _____
0x0016	S_____ R7, R7, R7
0x0018	_____ R6, R1, R7
0x001A	_____ R5, R5, _____
0x001C	B_____ , _____
0x001E	_____ R3, _____

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, 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/l/a	@D	WrD	Wr-Out	Rd-In	Wr-Mem	Byte	TknBr	N (hexa)	ADDR-IO (hexa)
LDB R6, 0(R4)															
BNZ R4, 7															
IN R1, 5															
CMPEQ R2, R3, R0															

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	0	X																				A / L
0	1	0	0	X																				ST
1	0	0	0	0																				BZ
1	0	0	1	1																				MOVHI

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. 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
SHL R2, R1, R5	
ST 6(R6), R3	
OUT 11, R3	
MOVI R5, 0x66	

Ejercicio 8 (1 punto)

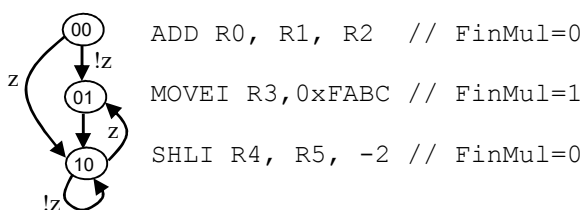
Se ha conectado a la UPG un dispositivo externo de entrada que nos envía valores y que tiene el registro de status en la dirección 4 del espacio de direccionamiento de entrada y el de datos en la 8. Este dispositivo tiene un efecto lateral en la lectura/escritura del dato sobre su registro de estado.

Se desea que este sistema vaya leyendo indefinidamente los datos que se reciban por el dispositivo de entrada mientras el valor de estos datos sea distinto de 0, y los vaya almacenando consecutivamente en la memoria de datos a partir de la posición 0x0064. Los datos recibidos son de 16 bits y primero se almacenará el valor recibido y luego su valor negado bit a bit. Por ejemplo, si el primer valor recibido es el 0xABCD, este valor se almacenará en la posición 0x0064 y en la posición 0x0066 se almacenará su valor negado (0x5432), cuando llegue el segundo valor por el dispositivo de entrada, por ejemplo el valor 0x1234, este valor se almacenará en la posición de memoria 0x0068 y su valor negado (0xEDCB) en la posición 0x006A, etc. Cuando el dato recibido sea 0, el sistema debe quedarse en un bucle infinito sin hacer nada.

Usando el procesador SISC Harvard uniciclo, escribid el programa en código ensamblador SISA para que realice la función anteriormente descrita.

Ejercicio 9 (1 punto)

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 de igual modo como lo hicisteis en la práctica 4. 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	
@2	
@3	
@4	
@5	
@6	
@7	

@ROM	Contenido ROM_OUT
@0	
@1	
@2	
@3	

