Evamon	E2	122 40	naviambra	40	20101
Examen	⊏3	(ZZ UE	noviembre	ue	2010)

۸.	11:	and the second s	C	DNI:
A	pellidos y	nombre:	Grup:	DIN1:

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 uniciclo (UPG+I/O+MEM) para que guarde en el registro R0 el byte almacenado en la posición de memoria 0x3456.

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)
IN R1 // OUT R5 // ADDI -, R5, 0x1234									
ADD R0, R3, R7 // 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)

Mnemotécnico	@A	@B	Rb/N	ОР	F	In/Alu	@D	WrD	N (hexa)
	001	111	1	00	101	0	011	1	XXXX
	010	XXX	0	01	000	1	100	1	FFFC

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.

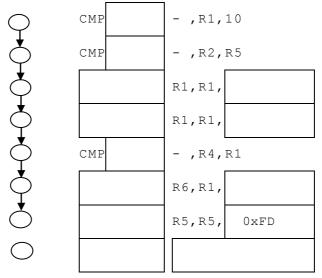
Lenguaje máquina SISA	Lenguaje ensamblador SISA
0x477D	
0xA1FC	
	MOVHI R4, -1
	BZ R3, -5

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 ((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)*



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, R1, R7
0x0004	В	R7,
0x0006	CMP	R7, R2, R5
0x0008	В	R7,
0x000A		R7,
0x000C		R1, R1,
0x000E	ADD	R1, R1,
0x0010	CMP	R7, R1, R4
0x0012	В	R7,
0x0014	MOVI	R7,
0x0016		R7,
0x0018		R6, R1, R7
0x001A		R5, R5,
0x001C	В	
0x001E		R3,,

F.,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	Γ	122	4-	noviembre	-1-	2010
Examen	E.3	(22	ue	noviembre	ue	2018

Apollidos y pombro:	Crun	DNI
Apellidos v nombre:	GIUD	DINI

Ejercicio 5 (1 punto)

Escribid sobre la siguiente tabla el valor de los bits que tiene la palabra de control del SISC-Harvard uniciclo (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 \mathbf{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.

		Palabra de Control del SISC Harvard uniciclo													
Instrucción SISA	@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(R6), R4															
BZ R0, -13															
OUT 0x23, R1															
CMPEQ R3, 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.

	Direc	cción	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	0P ₀	MxN ₁	MxN_0	MxF	f ₂	-	f ₀	MxD ₁	MxD_0	
0	0	0	1	Χ																					CMP
0	1	0	1	Х																					LDB
1	0	0	0	1																					BNZ
1	0	0	1	1																					мочні

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	
LDB R3·, 6(R5)	
IN R3, 11	
MOVHI R5,0x66	

Ejercicio 8 (1,5 puntos)

@7

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. Los datos recibidos son de 16 bits, y cada dato recibido se usará como una dirección de memoria. El sistema tiene que escribir un byte con valor 0x20 en cada dirección de memoria recibida cuando esta sea impar. Si la dirección de memoria recibida es par, entonces el sistema tendrá que escribir el word 0xFFFE en memoria en la dirección recibida. Cuando el dato recibido sea 0, el sistema debe quedarse en un bucle infinito sin hacer nada.

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 0xFFFE 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.

