

Apellidos y Nombre:Grupo:.....DNI:

Examen 4 (temas 12, 13 y 14)

- 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. En hoja aparte se os da una “chuleta” con información útil para realizar los ejercicios.
- La solución del examen se publicará en Atenea mañana por la tarde y las notas antes del próximo lunes a la noche. En general, para revisar la nota de este examen debéis enviar un email a vuestro profesor de teoría el próximo martes, excepto que el profesor indique otra cosa.

Ejercicio 1 (3 puntos)

El programa ensamblador se ha traducido a lenguaje máquina para ser ejecutado en el SISC Von Neumann, situando la sección `.text` a partir de la dirección `0x0000` de memoria y a continuación la sección `.data`. Suponed que justo antes de ejecutarse el programa los registros/puertos de entrada `KEY-STATUS` y `KEY-DATA` contienen `0x0001` y `0x0005` respectivamente.

a) Una vez cargado el programa en memoria:

- ¿A qué dirección de memoria corresponde la etiqueta, o dirección simbólica, `V`? (0,25 puntos)

V = 0x

- ¿Cuál es la dirección de memoria y su contenido donde han quedado almacenadas las siguientes instrucciones? (0,75 puntos)

MOVI R3, LO(F)	=> Mem _w [0x] = 0x
BZ R6, L3	=> Mem _w [0x] = 0x
ST 20(R2), R0	=> Mem _w [0x] = 0x

- b) Una vez ejecutado el programa en el computador Von Neumann ¿Cuál es la dirección de memoria donde ha escrito por última vez la instrucción (ST 20(R2), R0) y cuál es su contenido? (1 punto)

Mem_w[0x] = 0x

- c) ¿Cuánto tarda en ejecutarse el código en las tres versiones de los computadores SISC? Suponed que el tiempo de ciclo del Harvard unicycle, el Harvard multiciclo y el Von Neumann es de 4.000, 1.000 y 1.400 u.t. respectivamente.

¿Cuánto vale x para que sea cierta la siguiente afirmación? “El computador Harvard unicycle tarda, en ejecutar este código, un $x\%$ más que el Harvard multiciclo.” (1 punto).

Tejec(Harv.uni.) = ; Tejec(Harv.multi.) = ; Tejec(V.Neumann) = ; x =

```
.data
N = 7; tiene que ser <= 10
F: .space 1
   .even
V: .word 2,-5,264,-63,23
   .word 58,-64,32,0,-7
W: .space 20
.text
L1: IN      R0, KEY-STATUS
    BZ      R0, L1
    IN      R0, KEY-DATA
    MOVI     R2, LO(V)
    MOVHI    R2, HI(V)
    MOVI     R3, LO(F)
    MOVHI    R3, HI(F)
    MOVI     R1, N
    MOVI     R4, 1
L2: LD      R5, 0(R2)
    CMPLT    R6, R5, R0
    BZ      R6, L3
    ST      20(R2), R0
    STB      0(R3), R4; Flag=1
L3: ADDI     R2,R2,2
    ADDI     R1,R1,-1
    BNZ      R1, L2
.end
```

Ejercicio 2 (1 punto)

Completad la sección `.text` para que escriba todos los bits a 1 en todo el espacio reservado por la sentencia `W: .space 20`, suponiendo que la sección `.data` de este programa es igual a la del *Ejercicio 1*

`.text`

L1: R0,

L2: R0,

L3: R1,

L4: R2,

L5: 0(R0),

L6: R0, R0,

L7: R2, R2,

L8: R2,

`.end`

Ejercicio 3 (2 puntos)

Cada uno de los apartados pregunta sobre un ciclo concreto de la ejecución de una instrucción en el SISC Von Neumann (No se trata de la ejecución ciclo a ciclo de un programa, si no de ciclos sueltos de instrucciones sueltas). La siguiente tabla define la situación en la que se encuentra la Unidad de Control (UC) en cada apartado indicando el nodo/estado de la UC en ese ciclo y la instrucción (en ensamblador) que está almacenada en el IR en ese ciclo. Por claridad, nos referimos a cada nodo/estado por el mnemotécnico de su salida (Así, el estado E0 lo denotaríamos como F). Podéis ver el grafo de estados de Moore del circuito secuencial de la UC en el anexo. Suponed, para responder al apartado d, que el contenido de R4 antes de ejecutarse la instrucción BZ R4, -5 era 0xFFFF.

Además, se aprovecha esta primera tabla para que podáis responder, para cada apartado, a la siguiente pregunta: **¿Cuál es el nodo/estado que se ejecutará al ciclo siguiente?**

Apartado	Nodo/Estado (Mnemo Salida)	Instrucción en el IR (en ensamblador)	Nodo/Estado Siguiente (Mnemo Salida)
a	F	(no se sabe)	
b	D	STB -3(R1), R7	
c	Movhi	MOVHI R1, 0xA3	
d	Bz	BZ R4, -5	
e	Cmp	CMPLT R6, R2, R3	

En la siguiente tabla debéis **escribir el valor de los bits de la palabra de control** que genera el bloque SISC Von Neuman CONTROL UNIT durante el ciclo a que hace referencia cada apartado. Poned x siempre que no se pueda saber el valor de un bit (ya que no sabemos cómo se han implementado las x en la ROM_OUT).

Apartado	@A	@B	Pc/Rx	Ry/N	OP	F	P//L/A	@D	WrD	Wr-Out	Rd-In	Wr-Mem	LdIr	LdPc	Byte	Alu/R@	R@/Pc	N (hexa)	ADDR-IO (hexa)
a																			
b																			
c																			
d																			
e																			

Ejercicio 4 (1,5 puntos)

Completad las filas y columnas sombreadas en la siguiente tabla que representa en forma compacta el contenido de la ROM_OUT de la unidad de control del SISC Von Neumann. Poned x siempre que un bit pueda valer tanto 0 como 1. Tened en cuenta que cuando se ejecuta una instrucción con un código de operación que no coincide con ninguno del SISA no se debe modificar el estado del computador, excepto el PC que se debe incrementar para pasar a ejecutar la siguiente instrucción (podemos decir que estos códigos de operación no usados en ninguna de las 25 instrucciones SISA codifican instrucciones NOP).

Apellidos y Nombre:Grupo:.....DNI:

@ROM	Bnz	Bz	WrMem	RdIn	WrOut	WrD	Ldr	Byte	R@/Pc	Alu/R@	Pc/Rx	Ry/N	P/IL/A1	P/IL/A0	OP1	OP0	MxN1	MxN0	MxF	F2	F1	F0	Mx@D1	Mx@D0	
0																									F
1																									D
2																									Al
3																									Cmp
4																									Addi
5																									Addr
6																									Ld
7																									St
8																									Ldb
9																									Stb
10																									Jalr
11																									Bz
12																									Bnz
13																									Movi
14																									Movhi
15																									In
16																									Out
17..31																									Nop

Ejercicio 5 (2,5 puntos)

Completad el diseño del SISC Von Neumann para que pueda ejecutar, además de las 25 instrucciones originales SISA, la nueva instrucción LD+, que tiene el formato y codificación, la sintaxis ensamblador y la semántica siguientes:

Codificación: 1011 aaa ddd nnnnnn

Sintaxis: LD+ Rd, 0xN6(Ra)

Semántica: $Rd = Mem_w[(Ra + SE(N6)) \& (\sim 1)]$; $Ra = Ra + 2$;

Para ello, solo debéis modificar el contenido de las dos memorias ROM de la unidad de control. Se pide:

- a) Dibujad el fragmento del grafo de estados del circuito secuencial de la unidad de control necesario para ejecutar completamente la nueva instrucción LD+ (por lo que en el fragmento de grafo se debe incluir el estado F y el D). Etiquetad todos los arcos que salen de cada nodo. Usad la misma leyenda que la del grafo original que podéis ver en el Anexo. Aprovechad al máximo los nodos existentes en el grafo original para que el grafo final completo (que no hay que dibujar) tenga el menor número de nodos posible. Numerad los estados y poned el mnemotécnico de la salida para cada estado. Para los estados del grafo original poned el número y el mnemotécnico originales (por ejemplo E0 / F para el estado de búsqueda) y para los nuevos nodos usad números a partir del E17 (incluido) y mnemotécnicos no usados en el grafo original. (0,5 puntos)

- b) Completad la siguiente tabla que indica, para cada nodo del fragmento de grafo, la acción (o acciones en paralelo) que se realiza la UP en ese ciclo, usando un lenguaje de transferencia de registros como el usado en la documentación (y con el que ya hemos completado las acciones del nodo E0/F. Usad las filas de la tabla que necesitéis. (0,5 puntos)

Nodo		Acciones
Número	Mnemo	
E0	F	$IR \leftarrow Mem_w[PC] \ // \ PC \leftarrow PC + 2$

- c) Indicad los bits (0, 1 o x) de las palabras de la ROM_OUT para cada estado del fragmento del grafo que habéis dibujado en el apartado anterior. Hemos marcado las palabras de la ROM con direcciones 0 y 1 para los estados E0 y E1 que corresponden a los Mnemotécnicos de salida F y D ya que, como hemos dicho deben estar incluidos en el fragmento de grafo necesario para ejecutar completamente la nueva instrucción LD+. Al final hemos marcado varias filas para los nuevos estados. Si se usa algún otro estado original distinto de F y D también debéis completar su fila en la tabla, así como las filas completas de los estados originales que deban modificar su comportamiento, por pequeña que sea la modificación. (1,5 puntos)

@ROM	Bnz	Bz	WrMem	RdIn	WrOut	WrD	Ldlr	Byte	R@/Pc	Alu/R@	Pc/Rx	Ry/N	P/I/L/A1	P/I/L/A0	OP1	OP0	MxN1	MxN0	MxF	F2	F1	F0	Mx@D1	Mx@D0		
0																								F		
1																								D		
2																									AI	
3																										Cmp
4																								Addi		
5																								Addr		
6																									Ld	
7																										St
8																										Ldb
9																										Stb
10																								Jalr		
11																									Bz	
12																										Bnz
13																									Movi	
14																										Movhi
15																									In	
16																										Out
..31																									Nop	