

Examen Final, parte 2

- Duración de esta parte: 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 la hoja anexa se os da una “chuleta” con información útil para realizar los ejercicios.
- La solución se publicará mañana y las notas se publicaran el 28/06/2018. La revisión será el 29/06/2018 a las 12:00 en el aula D6-114.

Ejercicio 1 (0,4 puntos)

El programa se ha traducido a lenguaje máquina para ser ejecutado en el SISC Von Neumann, situando la sección `.data` a partir de la dirección 0x80A0 de memoria y a continuación la sección `.text`.

Una vez ensamblado y cargado el programa en memoria:

- a) ¿A qué direcciones de memoria corresponden las etiquetas, o direcciones simbólicas siguientes?

C=0x	E=0x	L1=0x
------	------	-------

- b) ¿Cuál es la dirección de memoria y su contenido donde han quedado almacenadas las siguientes instrucciones una vez cargado el programa?

MOVI	R2, LO (C)	=> Mem _w [0x]= 0x
BZ	R0, L1	=> Mem _w [0x]= 0x
LD	R5, 0 (R2)	=> Mem _w [0x]= 0x

```
.data
    N = 5
A:   .space 1
B:   .byte -27
     .even
C:   .word 2,-5,264,-63,23
D:   .byte 58,-64,32,0,-7
     .even
E:   .space 20
.text
L1:  IN      R0, KEY-STATUS
     BZ      R0, L1
     IN      R0, KEY-DATA
     MOVI    R2, LO (C)
     MOVHI   R2, HI (C)
     MOVI    R1, N
     MOVI    R4, 1
L2:  LD      R5, 0 (R2)
     ...
```

Ejercicio 2 (0,6 puntos)

Indicad qué cambios hay en el estado del computador SISC Von Neumann después de ejecutar cada una de las instrucciones de la tabla suponiendo que antes de ejecutarse cada una de ellas el PC vale 0x3C18, el contenido de todos los registros es 0x9ABC y que el contenido de todas las direcciones pares de la memoria es 0xA5 y el de todas las impares es 0x00. Utiliza el mnemotécnico MEM_b[...]=... y/o MEM_w[...]=... para indicar los cambios en la memoria.

Instrucción a ejecutar	Cambios en el estado del computador
LDB R2, -1 (R6)	
ST 6 (R0), R3	
BZ R4, 2	

Ejercicio 3 (0,5 puntos)

Cada uno de los apartados pregunta sobre un ciclo concreto de la ejecución de una instrucción en el SISC Von Neumann. Escribid el valor de los bits de la palabra de control que genera el bloque **SISC Von Neumann 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). La segunda y tercera columna definen la situación en la que se encuentra la Unidad de Control (UC) para cada apartado/fila: nodo/estado de la UC en ese ciclo e instrucción (en ensamblador) que está almacenada en el IR en ese ciclo. Suponed que el contenido de R4 antes de ejecutarse cada instrucción es 0x0000.

Apartado	Nodo/Estado (Mnemo Salida)	Instrucción en IR (en ensamblador)	Palabra de Control																	
			@A	@B	Pc/Rx	Ry/N	OP	F	P/I/L/A	@D	WrD	Wr-Out	Rd-In	Wr-Mem	LdIr	LdPc	Byte	Alu/R@	R@/Pc	N (hexa)
a	Addi	ADDI R3, R1, -8																		
b	Bnz	BNZ R4, -16																		
c	Addr	LD R2, -3 (R5)																		

Ejercicio 4 (1 punto)

Completad el fragmento de código SISA para que al ejecutarse se realice la siguiente funcionalidad. Antes de comenzar la ejecución de la primera instrucción del fragmento de código el registro R6 contiene la dirección de memoria donde se encuentra el primer elemento de un vector. Cada elemento es un número entero codificado en complemento a dos en 8 bits (cada elemento del vector ocupa un byte de memoria. Los elementos del vector se almacenan en bytes consecutivos de memoria.

Si el primer elemento del vector es negativo termina la ejecución del fragmento de código sin imprimir nada. Si no es así, imprime los primeros elementos positivos y consecutivos del vector cambiados de signo. Cuando encuentra el primer elemento negativo termina la ejecución del fragmento de código sin imprimir su valor cambiado de signo. Por lo tanto, imprime una secuencia de valores negativos. Cada valor impreso es un número entero codificado en complemento a dos con 16 bits (2 bytes, una palabra). Por ejemplo, si los tres primeros elementos/bytes del vector codifican en complemento a dos el 3, 5 y -8 al ejecutarse el fragmento de programa se imprimirán dos palabras/words que codifican el -3 y el -5.

No se pueden añadir etiquetas al código por lo que las instrucciones de salto condicional no pueden usar etiquetas. Para acceder a los puertos de entrada/salida sí que se pueden usar sus direcciones simbólicas.

Ejercicio 5 (2,5 puntos)

En el diseño del lenguaje máquina SISA se tomaron decisiones que son bastante arbitrarias. Una de ellas es el código de operación que se asignó a cada instrucción (como por ejemplo que a la instrucción STB se le asignara el código de operación, bits 15...12 de la instrucción, 0110) o a cada tipo de instrucciones (como que a las instrucciones de movimiento MOVI y MOVHI se les asignara el código 1001).

Una empresa de computadores ha creado un lenguaje máquina exactamente igual al SISA (las mismas instrucciones con la misma semántica, los mismos formatos de las instrucciones, etc.) excepto que ha asignado de forma diferente los códigos de operación de las instrucciones. El lenguaje máquina resultante se denomina SISAn. La tabla del Formato y Codificación de las instrucciones SISAn se muestra a la derecha.

También a la hora de implementar el SISC Von Neumann se tomaron decisiones bastante arbitrarias. Una de ellas es la codificación de cada uno de los 18 estados del grafo de la UC (El código 0 se asignó al estado de búsqueda, Fetch, el 1 al de decodificación, D, el 2 al de ejecución, propiamente dicha, de las instrucciones Aritmético lógicas, Al, etc.).

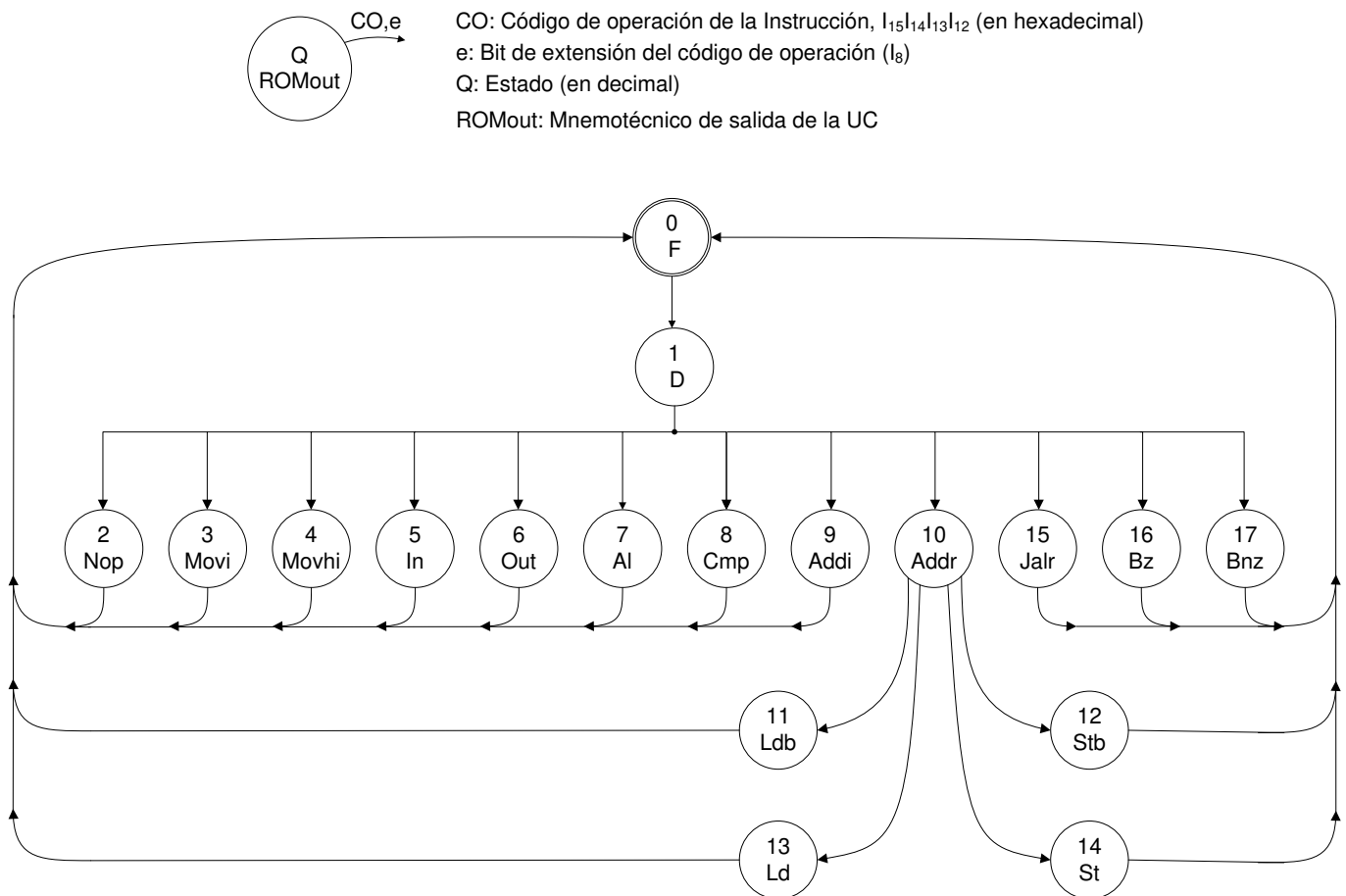
La empresa que diseñó el SISAn ha implementado un computador, que llama SISCn, que ejecuta código SISAn y que es exactamente igual al SISC Von Neumann, excepto que tuvieron que cambiar el grafo de estados de la UC para adaptarse a los cambios en los códigos de operación y además también cambiaron, de forma arbitraria, la codificación de los nodos del grafo. Esto hace que el hardware del SISCn sea exactamente igual al del SISC Von Neumann, que se muestra en el anexo, excepto el contenido de las dos ROMs que implementan el grafo de estados de la UC.

El grafo incompleto de la UC del SISCn se muestra a continuación, donde se ven la leyenda del grafo, los 18 nodos cada uno con su código y todos los arcos del grafo a los que les faltan sus etiquetas. Observad que el código 0 se ha asignado al nodo F, el 1 al D, como en el SISC Von Neumann, pero el resto de nodos tienen códigos diferentes: por ejemplo, el nodo Al tiene el código 7 en vez del 2. Responded a las siguientes preguntas:

M		R0,	
L		R1,	
C		R7, R1, R0	
		R7,	
A		R6,	
		R1,	
A		R1,	
		R2,	
B			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Mnemonic
0	0	0	0	a	a	a	0	n	n	n	n	n	n	n	n	BZ
							1									BNZ
0	0	0	1	a	a	a	d	d	d	x	x	x	x	x	x	JALR
0	0	1	0	a	a	a	d	d	d	n	n	n	n	n	n	LD
0	0	1	1	a	a	a	b	b	b	n	n	n	n	n	n	ST
0	1	0	0	a	a	a	d	d	d	n	n	n	n	n	n	LDB
0	1	0	1	a	a	a	b	b	b	n	n	n	n	n	n	STB
0	1	1	0	a	a	a	d	d	d	n	n	n	n	n	n	ADDI
0	1	1	1	a	a	a	b	b	b	d	d	d	f	f	f	AND, OR, XOR, NOT, ADD, SUB, SHA, SHL
1	0	0	0	a	a	a	b	b	b	d	d	d	f	f	f	CMPLT, CMPLT, -, CMPEQ, CMPLTU, CMPEU, -, -
1	0	0	1	d	d	d	0	n	n	n	n	n	n	n	n	IN
				a	a	a	1									OUT
1	0	1	0	d	d	d	0	n	n	n	n	n	n	n	n	MOVI
							1									MOVHI
1	0	1	1	x	x	x	x	x	x	x	x	x	x	x	x	Future Extensions
1	1	x	x													

a) Completad el dibujo del grafo de la UC del SISCn añadiendo al grafo las etiquetas (con el formato indicado en la leyenda del grafo) de todos los arcos que entran y salen de los nodos 2 (Nop), 8 (Cmp) y 10 (Addr). (0,5 puntos)



- ROM_Q+[330] = , ROM_Q+[59] = ,
ROM_Q+[.....] = 16 , ROM_Q+[.....] = 6

- ROM_Q+[.....] = ,

- ROM_OUT [4] = 0x....., ROM_OUT [.....] = 0x040031,
- ROM_OUT [12] = 0x....., ROM_OUT [28] = 0x.....,