

Examen 3 (temas 8, 9, 10 y 11)

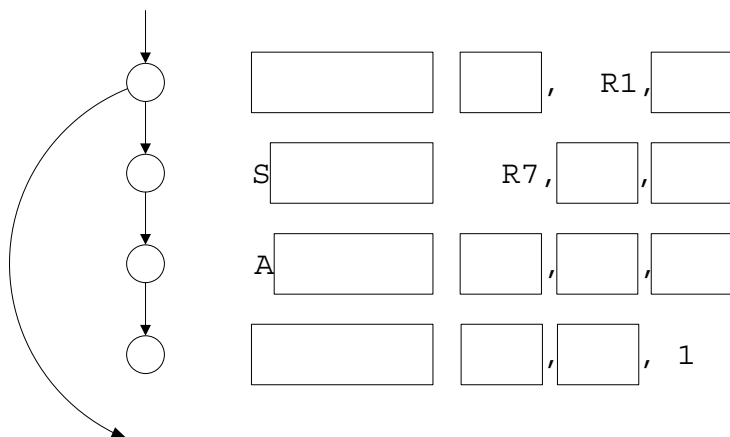
Duración del examen: 2 horas. Escribid la solución de cada ejercicio en el espacio reservado para ello en el enunciado. No podéis utilizar calculadora, móvil, apuntes, etc. La solución se publicará en Atenea mañana por la tarde y las notas antes de una semana.

Ejercicio 1 (1,5 puntos)

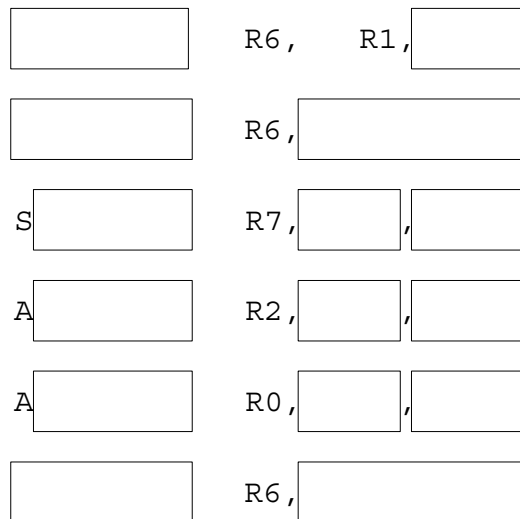
Para realizar la funcionalidad `while (R0>R1) {R2=R2*3; R0=R0-1;}`, considerando que los registros contienen números naturales, completad:

- a) El grafo de estados de la UC de **propósito específico** para que, junto con la UPG, formen un PPE que la realice. No faltan nodos, faltan arcos y sus etiquetas (0, 1, o x) y la palabra de control en mnemotécnicos de cada nodo. (R7 es un reg. temporal)
- b) El código en ensamblador SISA para realizarla en el sistema formado por la unidad de control general junto con la UPG y el subsistema de entrada/salida (UCG+UPG+IO_{key-print}). (Se usan R6 y R7 para almacenar valores temporales).

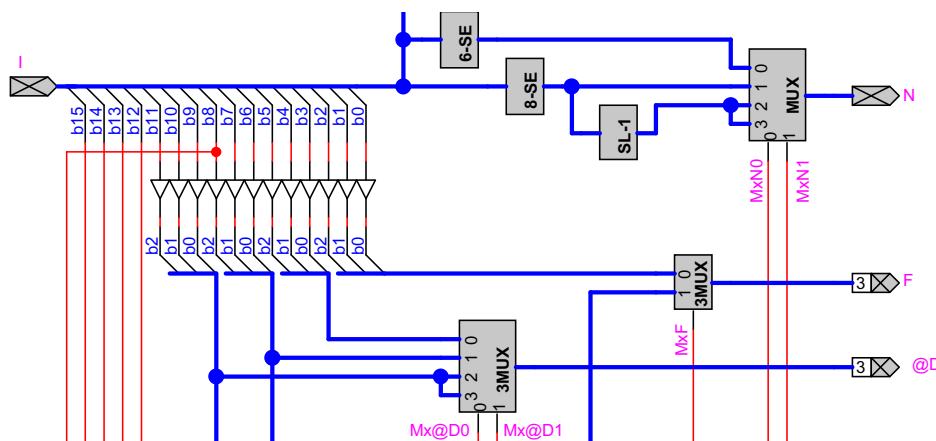
a) (1 punto)



b) (0,5 puntos)

**Ejercicio 2 (2 puntos)**

Escribid el contenido de las filas que se muestran de la tabla de la ROM del bloque ROM-CTRL-LOGIC del computador SISC Harvard uniclo (UCG+UPG+IO_{key-print}+MEM). La figura muestra parte de la lógica de control. Las señales de la palabra de control que no se muestran salen directamente de la ROM (excepto ADDR-IO que son los 8 bits de menor peso de la instrucción. Indicad con x los valores de los bits que pueden ser indistintamente 0 o 1.



Dirección					Contenido																				
I<15>	I<14>	I<13>	I<12>	I<8>	Bnz	Bz	Wr-Mem	RdIn	WrOut	WrD	Byte	Rb/N	-/I/a1	-/I/a0	OP1	OP0	MxN1	MxN0	MxF	F2	F1	F0	Mx@D1	Mx@D0	
0	0	1	0	x																					ADDI
0	1	0	0	x																					ST
0	1	0	1	x																					LDB
1	0	0	0	1																					BNZ
1	0	0	1	0																					MOVI
1	0	0	1	1																					MOVHI
1	0	1	0	0																					IN

ADDI

ST

LDB

BNZ

MOVI

MOVHI

IN

Ejercicio 4 (3,5 puntos)

Completad la tabla en la que cada fila es un apartado diferente que contiene 4 columnas para una misma instrucción: 1) Instrucción en ensamblador SISA, 2) Instrucción en lenguaje máquina (LM) en hexadecimal, 3) algunos bits de la palabra de control del SISC Harvard uniciclo (UCG+UPG+IO_{key-print}+MEM), ver pag. 4, (poned x siempre que no se sepa el valor del bit al no saber cómo se han implementado las x en la ROM de la Lógica de Control) y 4) estado del computador después de ejecutar la instrucción suponiendo que el estado antes de su ejecución es: PC=0x03DE; Ri=2*i para i=0,... 7; MEM_w[@]=@+2 para @=0, 2, 4, 6... (2¹⁶)-2 y IN[p]=OUT[p]=p%2 para p=0,... 255. Escribid solo el contenido, en hexadecimal, de los registros (incluido el PC), palabras de la memoria de datos, MEM_w (si se modifica un byte debe indicarse el valor de la palabra a la que pertenece) y puertos de entrada salida, IN[p] y OUT[p], que se modifican al ejecutar cada instrucción).

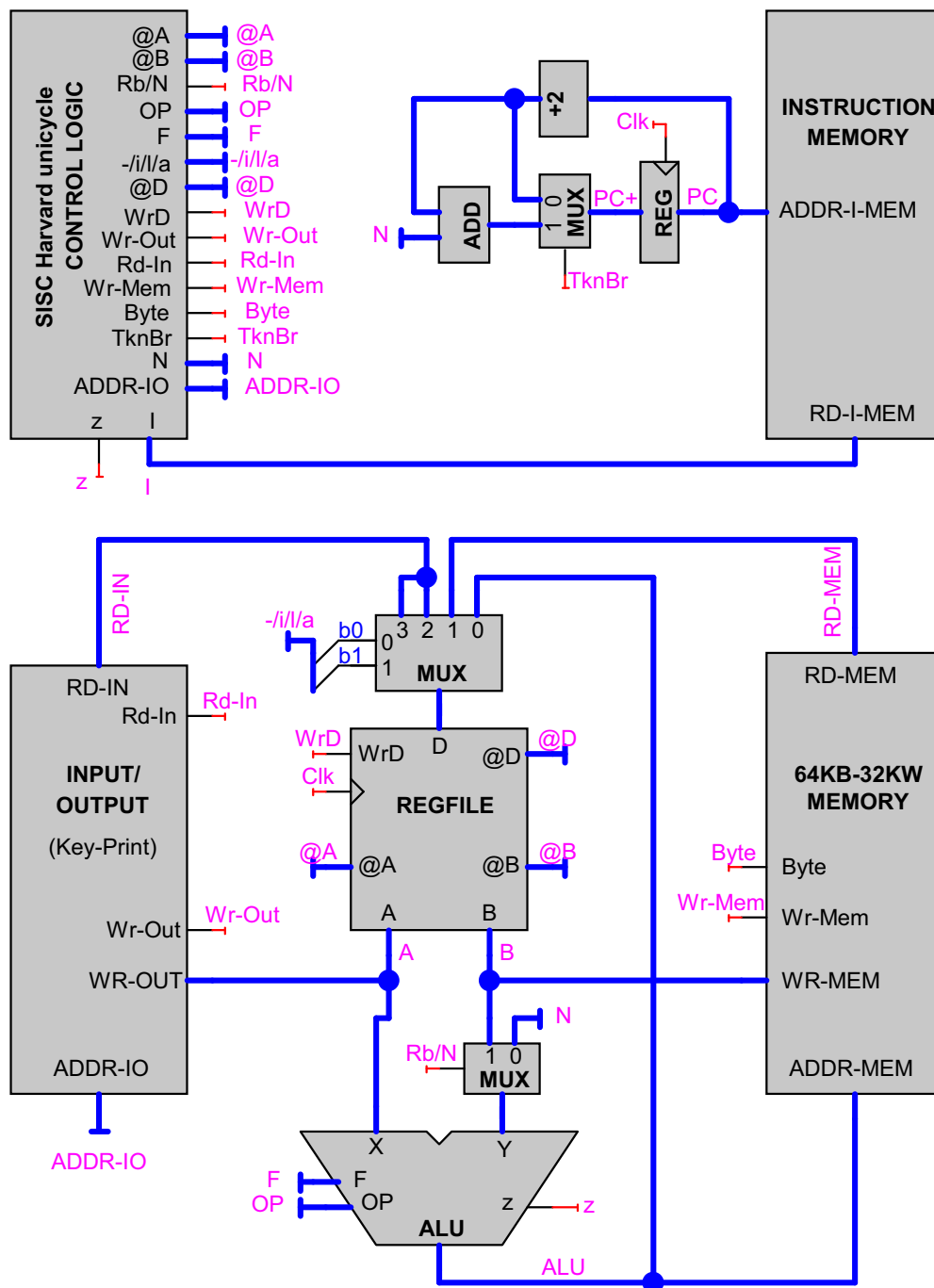
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Name	Mnemonic
0	0	0	0	a	a	a	b	b	b	d	d	d	f	f	f	Logic and Arithmetic Operations	AND, OR, XOR, NOT, ADD, SUB, SHA, SHL
0	0	0	1	a	a	a	b	b	b	d	d	d	f	f	f	Compare Signed and Unsigned	CMPLT, CMPLT, -, CMPEQ, CMPLTU, CMPLTU, -, -
0	0	1	0	a	a	a	d	d	d	n	n	n	n	n	n	Add Immediate	ADDI
0	0	1	1	a	a	a	d	d	d	n	n	n	n	n	n	Load	LD
0	1	0	0	a	a	a	b	b	b	n	n	n	n	n	n	Store	ST
0	1	0	1	a	a	a	d	d	d	n	n	n	n	n	n	Load Byte	LDB
0	1	1	0	a	a	a	b	b	b	n	n	n	n	n	n	Store Byte	STB
0	1	1	1													Branch future extension	BZ
1	0	0	0	a	a	a	0			n	n	n	n	n	n	Branch on Zero	BNZ
							1									Branch on Not Zero	BNZ
1	0	0	1	d	d	d	0			n	n	n	n	n	n	Move Immediate	MOVI
				a	a	a	1			n	n	n	n	n	n	Move Immediate High	MOVHI
				d	d	d				n	n	n	n	n	n	Input	IN
1	0	1	0	a	a	a	1			n	n	n	n	n	n	Output	OUT
1	0	1	1	x	x	x	x	x	x	x	x	x	x	x	x		Future extensions
1	1	x	x														

	1) Ensamblador	2) LM (Hexa)	3) Bits Pal. Control					4) Estado después de su ejecución
			-i/l/a	WrD	Byte	TknBr	N (hexa)	
a)	MOVI R3, 0x96							
b)	STB -12(R4), R0							
c)		392E						
d)		AA7F						
e)	BNZ R4, -8							
f)		9DB6						

Ejercicio 5 (1 punto)

Dado el computador SISC Harvard uniciclo (UCG+UPG+IO_{key-print}+MEM), escribid un código ensamblador SISA que, en solo 8 instrucciones, lea un dato de 16 bits del teclado y lo escriba en las tres palabras de memoria consecutivas a partir de la dirección 204 (incluida)

Estructura a bloques del SISC Harvard uniciclo (UCG+UPG+IO_{key-print}+MEM)



Funcionalidad de la ALU

F			OP			
b ₂	b ₁	b ₀	1 1	1 0	0 1	0 0
0	0	0	---	X	CMPLT (X, Y)	AND (X, Y)
0	0	1	---	Y	CMPLT (X, Y)	OR (X, Y)
0	1	0	---	MOVHI(X, Y)	---	XOR(X, Y)
0	1	1	---	---	CMPEQ (X, Y)	NOT (X)
1	0	0	---	---	CMPLTU (X, Y)	ADD (X, Y)
1	0	1	---	---	CMPLTU (X, Y)	SUB (X, Y)
1	1	0	---	---	---	SHA(X, Y)
1	1	1	---	---	---	SHL(X, Y)