E3 (7 de mayo de 2015) IC-14-15-Q2

### 

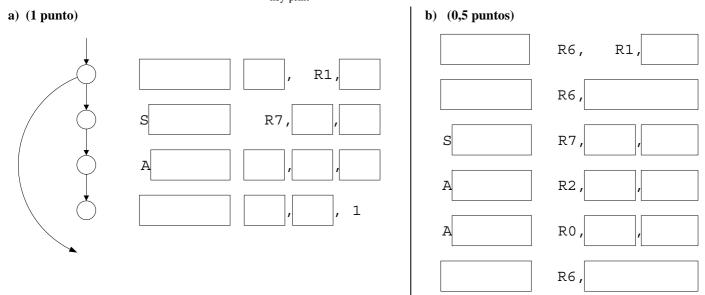
# Examen 3 (temas 8, 9, 10 y 11)

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)

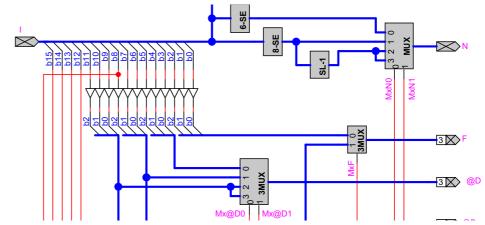
Para realizar la funcionalidad while (R0>R1) {R2=R2\*3; R0=R0-1;}, considerando que los registros contienen números naturales, completad:

- 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).



## 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 uniciclo (UCG+UPG+IO<sub>key-print</sub>+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.



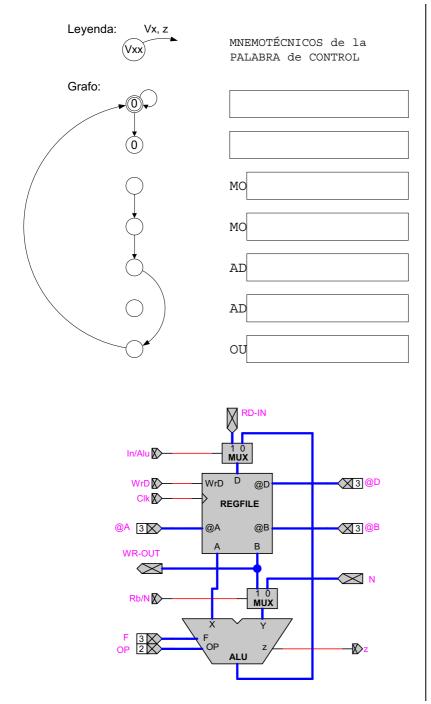
							C	Contenido																	
K15>	<b>1√4</b>	<u>√</u> 3	k12>	<b> &lt;8&gt;</b>	Bnz	Bz	Wr-Mem	RdIn	WrOut	WrD	Byte	Rb/N	-/i/l/a1	-/i/I/a0	0P1	OP0	MxN1	MxN0	MxF	F2	Ξ	P0	Mx@D1	Mx@D0	
0	0	1	0	Х																					ADDI
0	1	0	0	Х																					ST
0	1	0	1	Х																					LDB
1	0	0	0	1																					BNZ
1	0	0	1	0																					MOVI
1	0	0	1	1																					MOVHI
1	0	1	0	0																					IN

E3 (7 de mayo de 2015) IC-14-15-Q2

### Ejercicio 3 (2 puntos)

a) (1 punto) Completad el grafo de estados de la Unidad de Control Específica (UCE) de un PPE que junto a la UPG (sin espacio de E/S: con acciones IN Rd y OUT Rb) calcula indefinidamente el cuadrado de un número  $X_u$  (sin detectar si el resultado es no representable en 16 bits). Para calcular el cuadrado de  $X_u$  hace  $R2 = X_u$  y aplica  $X_u$ -1 veces la recurrencia  $R2 = R2 + X_u$ . El PPE se comunica con el exterior mediante dos buses de 16 bits, uno de entrada, RD-IN, y otro de salida, WR-OUT, y dos señales de validación de un bit, una de entrada, Vx, y otra de salida, Vxx. Si en el ciclo c la señal Vx vale 1 al ciclo siguiente, c+1, en la entrada RD-IN se encontrará el operando, Xu, de un nuevo cálculo. La señal Vx se debe ignorar desde el ciclo siguiente en el que Vx tomó el valor 1 hasta en ciclo anterior en el que Vxx tomará el valor 1, ambos incluidos.

El cálculo consiste en elevar al cuadrado  $X_u$ , módulo  $2^16$ , y mostrarlo por la salida WR-OUT. Para validar el resultado, durante un ciclo, el mismo en el que en WR-OUT se encuentre el valor  $(X_u^2)\%(2^16)$ , el PPE debe poner a 1 la señal Vxx. Suponed que  $X_u$  es siempre mayor que 0, de lo contrario no se puede realizar el cálculo correctamente con el grafo de la figura. En el grafo se debe usar R0 para mantener la constante Xu, R1 para contar (o descontar) las Xu-1 veces que hay que aplicar la recurrencia, y R2 como acumulador en la recurrencia (R2=R2+Xu). Indicad el valor de la salida Vxx en el interior de cada nodo del grafo.



b) (1 punto) Completad el código ensamblador SISA para que el sistema UCG+UPG+IO<sub>key-print</sub> haga lo que se hace en el apartado a: 1) entrar dato del teclado, 2) elevarlo al cuadrado y 3) imprimirlo. Se usan R0, R1 y R2 como antes (y cuando ya no hace falta la constante Xu, R0 se usa como registro temporal) En IN y OUT usad las direcciones simbólicas de los puertos.

IN	RO,
	R0,
	R1,
	R2,
	R1,
	R1,
	R2,
	R1,
	RO,
0	

<b>E3</b>	(7 de mayo de 2015)	IC-14-15-Q2

## 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<sub>key-print</sub>+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;  $MEM_{w}[@] = @+2 para @=0, 2, 4, 6... (2^16) -2$ 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<sub>w</sub> (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	4	13	12	Ξ	19	တ	∞	<b>6</b>	22	4 ო	0 1 2	Name	Mnemonic
0	0	0	0	а	a	a	b	b b	d d	d d	fff	Logic and Aritmetic Operations	AND, OR, XOR, NOT, ADD, SUB, SHA, SHL
0	0	0	1	а	a	a	b	b b	d o	d d	fff	Compare Signed and Unsigned	CMPLT, CMPLE, -, CMPEQ, CMPLTU, CMPLEU, -, -
0	0	1	0	а	а	a	d	d d	nı	n n	nnn	Add Immediate	ADDI
0	0	1	1	а	a	a	d	d d	nı	n n	nnn	Load	LD
0	1	0	0	а	a	a	b	b b	nı	n n	nnn	Store	ST
0	1	0	1	а	а	a	d	d d	nı	n n	nnn	Load Byte	LDB
0	1	1	0	а	a	a	b	b b	nı	n n	nnn	Store Byte	STB
0	1	1	1										Branch future extension
1	Λ	Λ	Λ	а	2	2	0	n n	n	n n	nnn	Branch on Zero	BZ
_	•	Ü	0	а	и	a	1	11 11	n n n n n n n n			Branch on Not Zero	BNZ
				d	d	d	0					Move Immediate	MOVI
1	0	0	1	а	a	a	1	n n	nı	n n	nnn	Move Immediate High	MOVHI
				d	d	d	_					move miniediate mgn	110 1112
1	0	1	0	d	d	d	0	n n	nı	n n	nnn	Input	IN
Ī	Ĭ	_	J	а	a	a	1					Output	OUT
1	0	1	1	х	x	x	x	хх	. x 2	хх	xxx		Future extensions
1	1	х	х										

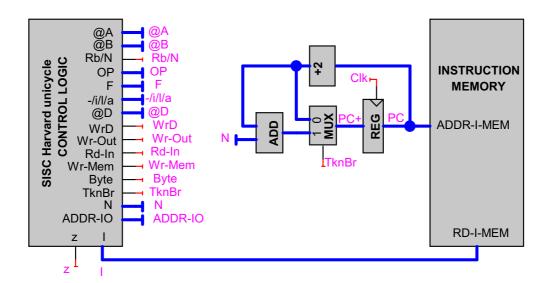
	1) Ensamblador	2) LM		,	3) B	its l	Pal.	Control	4) Estado después de su ejecución	
		(Hexa)	-/i/	/l/a	WrD	Byte	TknBr	N (hexa)		
a)	MOVI R3, 0x96									
<b>b</b> )	STB -12(R4), R0									
c)		392E								
<b>d</b> )		AA7F								
e)	BNZ R4, -8									
<b>f</b> )		9DB6								

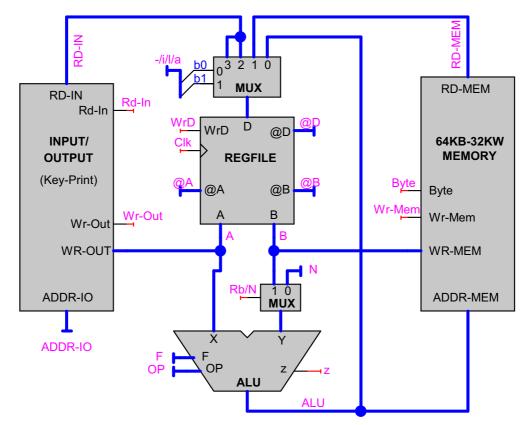
## Ejercicio 5 (1 punto)

Dado el computador SISC Harvard uniciclo (UCG+UPG+IO<sub>key-print</sub>+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)

E3 (7 de mayo de 2015) IC-14-15-Q2

# Estructura a bloques del SISC Harvard uniciclo (UCG+UPG+IO<sub>key-print</sub>+MEM)





#### Funcionalidad de la ALU

	F		OP									
b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>	11	1 0	0 1	00						
0	0	0		X	CMPLT (X, Y)	AND (X, Y)						
0	0	1		Y	CMPLE (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			CMPLEU (X, Y)	SUB (X, Y)						
1	1	0			10-006 - 21-0 1 0-00 - 10-00	SHA(X, Y)						
1	1	1				SHL(X, Y)						