

Examen 4 (temas 12 y 13)

- 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 jueves 28 de diciembre.
- En este curso, por problemas de calendario, no se incluye el tema 14 (ni se hará en el examen final), que habitualmente se incluía en el E4.

Ejercicio 1 (2,5 puntos)

Para ser ejecutado en el SISC Von Neumann, el programa en ensamblador SISA, una vez completado, se deberá traducir a lenguaje máquina y ser cargado en la memoria del computador a partir de la dirección 0x0000.

El programa lee del teclado N datos (con N menor de 32), donde cada dato es un byte, los guarda en memoria, en un vector, y luego los imprime en el orden inverso a como han sido leídos y esto se repite indefinidamente. En esta implementación del código N vale 10. Se considera que N es un número natural y cada uno de los N datos son enteros codificados en 8 bits en complemento a dos. Al escribir el programa no podéis usar las típicas direcciones simbólicas de los puertos de entrada/salida del teclado ni de la impresora: debéis usar la dirección (en decimal o hexadecimal) de los puertos involucrados (Ver figura en página 4). En el display de la impresora se verá cada dato (byte) en 16 bits, previa extensión del bit de signo (bit 7). Se usa R7 como registro temporal siempre que se puede.

- Completad el código SISA para la funcionalidad descrita. (1,5 puntos)
- Una vez cargado el programa en memoria, ¿Cuál es la dirección de memoria y el contenido donde se encuentra la séptima instrucción? (0,25 puntos)

ADDI R0,R0,-1 => Mem_w[0x] = 0x

- Cuál es la dirección de memoria donde se almacena el primer dato leído y cuál la del último de cada grupo de N datos leídos por el teclado? (0,5 puntos.)

Dir. Primer = 0x Dir. Último = 0x

- ¿Cuántas instrucciones se ejecutan y en cuantos ciclos desde la primera vez que se ejecuta la primera instrucción del código hasta la primera vez que se ejecuta la última instrucción del código, para el primer grupo de N datos. Suponed que cada vez que se lee el registro de estado del teclado o de la impresora su contenido vale 1. Incluid en la cuenta la ejecución de la primera y la ultima instrucción del código. Hacedlo para el caso programado de $N=10$. (0,25 puntos)

Nº de instruc. ejecutadas = Nº de ciclos =

MOVI R0, 10

ADDI R1, R0, 0

I	R7,	
B	R7,	
I	R7,	
	0x20 (R0),	

ADDI R0, R0, -1

B	R0,	
I	R7,	
B	R7,	
L	R7,	
O		
A	R0,	
C	R7, R0,	
BNZ	R7,	
	R7,	

Ejercicio 2 (1,25 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 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). Para cada apartado/fila se indica el nodo/estado de la UC en ese ciclo y la instrucción (en ensamblador) que está almacenada en el IR en ese ciclo. Podéis ver el grafo de estados de Moore de la UC en el anexo. Suponed que el contenido de todos los registros, R_k para k=0,...,7, antes de ejecutarse cada instrucción es 0.

Apartado	Nodo (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	Stb	STB 0x20(R3),R1																		
b	Bz	BZ R5,-61																		
c	Movhi	MOVHI R4,-9																		

Ejercicio 3 (1,5 puntos)

Completad la tabla que representa en forma compacta parte del 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.

@ROM	Bnz	Ldlr	R@/Pc	Alu/R@	Pc/Rx	Ry/N	MxN1	MxN0	MxF	Mx@D1	Mx@D0	
4												Addi
9												Stb
11												Bz
14												Movhi

Ejercicio 4 (1,25 puntos)

Indicad qué cambios se producen en el estado del computador SISC Von Neumann (con el teclado e impresora de la figura de la pag. 4) después de ejecutar cada una de las instrucciones de la tabla suponiendo que antes de ejecutarse cada una de ellas el PC vale 0xF0F8, el contenido de todos los registros es 0x6789, el byte contenido en todas las direcciones pares de la memoria es 0x83 y el de todas las impares es 0x01 y el contenido de todos los puertos de entrada es 1 y el de los de salida es 2. Utilizad la notación $MEM_b[0x\dots]=0x\dots$ para indicar cualquier cambio en la memoria y $PORTIN[0x\dots]=0x\dots$ para los puertos de entrada y $PORTOUT[0x\dots]=0x\dots$ para los de salida.

Instrucción a ejecutar	Cambios en el estado del computador
ST -7(R3), R5	
LDB R7, 0x3F(R7)	
OUT PRINT-DATA, R2	

Ejercicio 5 (3,5 puntos)

Completad el diseño del SISC Von Neumann para que pueda ejecutar, además de las 25 instrucciones originales SISA, las dos nuevas instrucciones CLRIC y MOVIC. Estas dos instrucciones acceden a un nuevo registro denominado IC (Instruction Counter). El IC forma parte del nuevo estado del computador, ya que se puede leer y escribir usando las dos nuevas instrucciones. El IC no es ninguno de los registros R0..R7, es un registro especial nuevo, que se añade a la Unidad de Proceso del computador. Cuando se usa como operando fuente y/o destino en una de las dos nuevas instrucciones no se indica explícitamente en ningún campo de la sintaxis ensamblador de estas instrucciones (es lo que se denomina operando implícito, se sabe que es un operando por el mnemotécnico de la instrucción en ensamblador o por el código de operación (y el bit e) de la instrucción en lenguaje máquina.

El registro especial IC se incrementa en una unidad cada vez que se ejecuta una instrucción cualquiera de las 25 del SISA original. Cuando se ejecuta alguna de las dos nuevas instrucciones o alguna con código ilegal no se incrementa el IC. Podemos decir que el nuevo SISA añade a la semántica de las instrucciones SISA originales la acción $IC = IC + 1$, aunque esto no se especifica en la semántica de ninguna de las instrucciones originales SISA para no resultar pesado (al igual que no se indica el $PC = PC + 2$ en las instrucciones que no son de salto). El formato y codificación, la sintaxis ensamblador y la semántica de las dos nuevas instrucciones se define a continuación:

Codificación: 1110 xxx 0 xxxxxxxx

Sintaxis: CLRIC ;El registro IC es el operando destino implícito

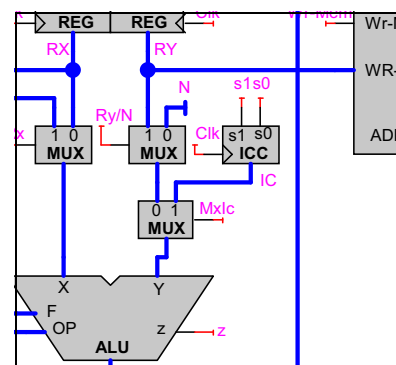
Semántica: $IC = 0$;Pone a 0 el registro IC

Codificación: 1110 ddd 1 xxxxxxxx

Sintaxis: MOVIC Rd ;El registro IC es el operando fuente implícito

Semántica: $Rd = IC$;Copia el contenido del registro IC en Rd.

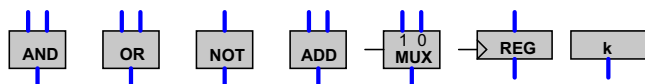
El registro IC junto con los circuitos combinacionales que lo controlan se han encapsulado en un bloque secuencial denominado ICC (Instruction Counter Circuit). El ICC tiene un bus de salida, IC, que codifica en binario un número natural de $n=16$ bits, y dos señales de entrada de un bit cada una, s1 y s0, que controlan su funcionamiento de la siguiente forma (interpretamos que $S=s1s0$ codifica en binario con dos bits un número natural, S_u): Si en el ciclo c la entrada S vale 0, $S(c)_u=0$, o vale 1, $S(c)_u=1$, al ciclo siguiente la salida IC tendrá el mismo valor que en el ciclo actual, $IC(c+1) = IC(c)$. Si $S(c)_u=2$ la salida al ciclo siguiente valdrá 0, $IC(c+1)_u = 0$, y si $S(c)_u=3$ la salida al ciclo siguiente valdrá una unidad más que en el ciclo actual, $IC(c+1)_u = IC(c)_u + 1$. Este tipo de contadores hardware (de instrucciones, de ciclos,...) y sus instrucciones de uso aparecen en muchos lenguajes comerciales para realizar evaluaciones del rendimiento de programas. Para poder ejecutar las nuevas instrucciones hay que añadir un nuevo multiplexor de buses, MUX-2-1, en la UP, con señal de selección Mxlc, como muestra la



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

figura. La Unidad de Control deberá generar la palabra de control original más las tres nuevas señales Mxlc y s1s0. La ROM_OUT deberá de generar estas tres nuevas señales de la Palabra de Control y la ROM_Q+ deberá modificarse para implementar el nuevo grafo de estados con los dos nuevos estados: Clrlc para la ejecución, propiamente dicha, de la instrucción CLRLIC y Movlc para la ejecución de la instrucción MOVLC. Se pide:

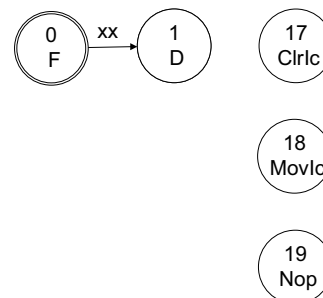
- a) Dibujad el circuito ICC usando unicamente el mínimo número de los siguientes bloques: AND, OR, NOT, ADD, MUX-2-1, REG y Constantes (en el bloque Constante de la figura se indica la constante k pero vosotros tendréis que indicar el valor en hexadecimal, con 0x delante, o en decimal de la constante que corresponda. **(0,5 puntos)**)



- b) Completad el fragmento del grafo de la unidad de control necesario para ejecutar las dos nuevas instrucciones y todas las que tienen un código ilegal (que se ejecutan en Nop). Se dan todos los nodos necesarios y su codificación, pero solo un arco con su etiqueta. Dibujad todos los arcos que faltan con sus etiquetas. **(0,25 puntos)**
- c) Completad, también, el contenido de la tabla que indica, mediante una fila para cada nodo, la acción o acciones en paralelo que se realiza en el computador en cada uno de los 4 ciclos/nodos que requiere la ejecución de las dos nuevas instrucciones (Fetch, Decode, un ciclo/nodo nuevo para CLRLIC y otro para MOVLC): F, D, Clrlc, Movlc. Para especificar las acciones usad el mismo lenguaje de transferencia de registros que en la documentación. **(0,5 = 0,25 (F, D y Nop) + 0,25 (Clrlc y Movlc) puntos)**

CO: Código de operación, $I_{15}I_{14}I_{13}I_{12}$ (en hexa)
e: Bit de extensión del c.o. (I_8)
Q: Estado (en decimal)
ROMout: Mnemotécnico de salida

Grafo:



Nodo/Estado		Acciones
Número	Mnem.	
0	F	
1	D	
17	Clrlc	
18	Movlc	

- d) Completad (poniendo 0, 1 o x en cada bit) las filas de las tablas que especifican parte del contenido de la ROM_OUT para que se ejecuten correctamente todas las instrucciones, poniendo el máximo número de x posibles. La dirección 0 de la ROM_OUT corresponde al estado 0 (F), la 1 al 1 (D)... la dirección 17 al estado 17 (Clrlc), la 18 al estado 18 (Movlc) y la 19 al estado Nop (No operación, que se usa para que las instrucciones con un código de operación ilegal no modifiquen el estado del computador y pasen a ejecutar la siguiente instrucción en secuencia). En la segunda tabla, completad las dos columnas de la ROM_OUT para las salidas Mxlc y S=s1s0, que forman parte de la nueva palabra de control. Estas dos columnas se han dibujado en horizontal para ocupar menos espacio (En la fila de cabecera se indica la dirección de la ROM_OUT y el mnemotécnico de salida. Se han sombreado las casillas que no tenéis que completar por haberlo hecho en la primera tabla). **(1,75 = 1,25 (primera tabla) + 0,5 (segunda tabla) puntos)**

@ROM	Mxlc	s1 s0	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	Mnemo
0															F
1															D
...
17															ClrIc
18															MovIc
19															Nop

	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	17 ClrIc	18 MovIc	19 Nop
Mxlc																				
s1 s0																				

- e) Indicad la dirección o las direcciones (en binario, usando x para representar varias direcciones cuyo contenido de la ROM es el mismo) de la memoria ROM_Q+ y su contenido o sus contenidos (en Hexa) para implementar correctamente el arco del nodo/ estado 1 al 17 y todos los arcos que salen del nodo 17. (0,5 puntos)

Apéndice: Teclado e Impresora del computador SISC Von Neumann

