

INFORME PREVIO PRÁCTICA 6

Para poder realizar las 7 instrucciones Aritmético-Lógicas y las 5 nuevas instrucciones de comparación hacen falta una serie de cambios en nuestro ordenador SISC Von Neumann, cambios en el hardware y en su memoria.

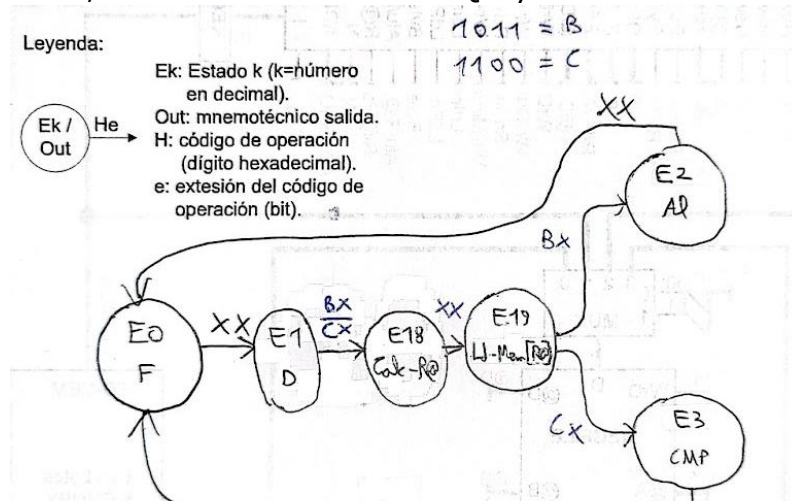
Al principio se podría plantear la posibilidad de añadir estas instrucciones solo añadiendo una instrucción más al computador, pero se observa que es algo inviable. Inviabile ya que en un solo ciclo no se puede saber el contenido de la dirección de memoria deseada, esto se debe por los registros que hay en el camino de la unidad aritmético-lógica y nuestra memoria.

Por lo tanto, la solución propuesta se basa en añadir dos estados más al computador, con el beneficio de que entonces solo tenemos que añadir un multiplexor al hardware, y entonces el tiempo de ciclo junto a otras instrucciones no se verían afectadas por la nueva implementación.

Una vez aclarada la solución final, hace falta explicarla. Primero de todo, se pasará por el estado Fetch y Decode como cualquier otra instrucción. Después pasará al estado 18 nombrado Calc-R@, donde guardaremos el valor que hay en la dirección de memoria guardada en Rb. Seguidamente, sin importar el valor del código de operación, se llega al estado 19 nombrado como Ld-Mem[R@], aquí se subtrae el valor deseado de la memoria y se guarda en Rb, luego se cargan los valores de Ra y Rb en Rx y Ry para luego realizar la operación. Finalmente, reutilizaremos los estados 2 (Al) y 3 (Cmp) para realizar la operación deseada según el código de operación.

De hardware solo hará falta utilizar un multiplexor 2-1, el cual se sitúa entre el regfile y el registro RY. Una de las entradas del multiplexor es Rb, y la otra va conectada directamente con la salida de la memoria con un cable nuevo.

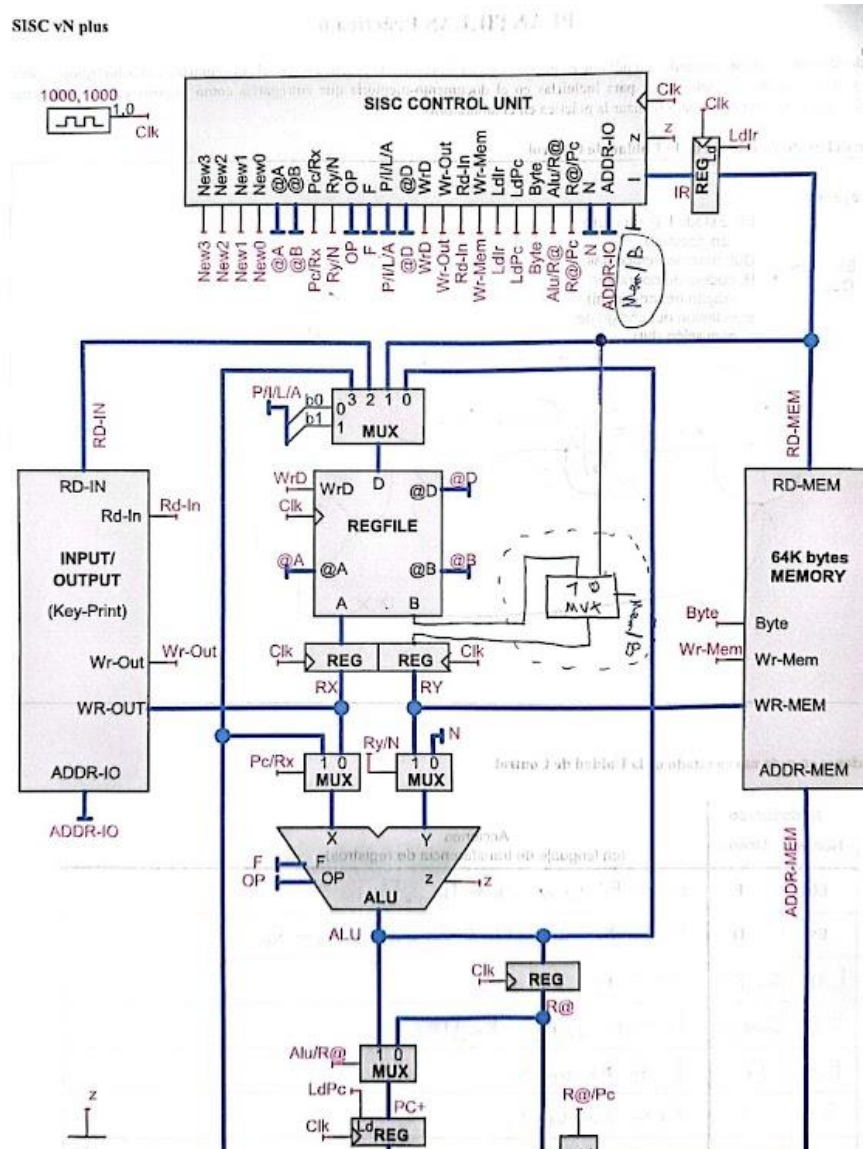
A continuación, fotografías que indican todos los cambios necesarios en el grafo de estados, el contenido de la ROM Q+ y de la ROM OUT:



1- Parte del grafo de estados modificado

Nodo/Estado		Acciones (en lenguaje de transferencia de registros)
Número	Mnem.	
E0	F	$PC \leftarrow PC + 2 // IR \leftarrow Mem_w[PC]$
E1	D	$R@ \leftarrow PC + 2 \cdot SE(NB) // Rx \leftarrow Ra // Ry \leftarrow Rb$
E18	Cal-R@	$R@ \leftarrow Ry$
E19	L-MEM[R@]	$Rx \leftarrow Ra // Ry \leftarrow Mem_w[R@]$
E2	AD	$Rd \leftarrow Rx \text{ op } Ry$
E3	CMP	$Rd \leftarrow Rx \text{ op } Ry$

2. Acciones en cada nuevo estado



3. Cambios en el hardware del SISC

Nuevo contenido de la ROM_OUT:

@ROM	Bz	Bz	WrMem	RdIn	WrOut	WrD	Ldr	Byte	R@Pc	AluR@	PcRx	RyN	PIL/A1	PIL/A0	OP1	OP0	MxN1	MxN0	MxF	F2	F1	F0	Mx@D1	Mx@D0	Node
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																									Nop
18																									Op-RR
19																									Op-RR
...																									
..31																									Nop

4. ROM OUT

Solución Práctica 6

Tabla ROM_Q+ (RQ+ori) del SISC vN plus sobre la que marcar los cambios

Q	I	Q*	Q* (Hexa)	# veces	Q* (Hexa)
F	x	D	00000 xxxxx 01	32	01
D	AL	Al	00001 0000x 02	2	02
D	CMP	Cmp	00001 0001x 03	2	03
D	ADDI	Addi	00001 0010x 04	2	04
D	LD	Addr	00001 0011x 05	2	05
D	ST	Addr	00001 0100x 05	2	05
D	LDB	Addr	00001 0101x 05	2	05
D	STB	Addr	00001 0110x 05	2	05
D	JALR	Jalr	00001 0111x 0A	2	0A
D	BZ	Bz	00001 10000 0B	1	0B
D	BNZ	Bnz	00001 10001 0C	1	0C
D	MOVI	Movi	00001 10010 0D	1	0D
D	MOVHI	Movhi	00001 10011 0E	1	0E
D	IN	In	00001 10100 0F	1	0F
D	OUT	Out	00001 10101 10	1	10
D	illegal	Nop	00001 1011x 11	2	11
			00001 1100x 12	8	11
Al	x	F	00010 xxxxx 00	32	00
Cmp	x	F	00011 xxxxx 00	32	00
Addi	x	F	00100 xxxxx 00	32	00
Addr	! (LD+ST+LDB+STB)	x	00101 0000x xx	2	00
			00101 0001x xx	2	00
			00101 0010x xx	2	00
Addr	LD	Ld	00101 0011x 06	2	06
Addr	ST	St	00101 0100x 07	2	07
Addr	LDB	Ldb	00101 0101x 08	2	08
Addr	STB	Stb	00101 0110x 09	2	09
Addr	! (LD+ST+LDB+STB)	x	00101 0111x xx	2	00
			00101 1xxxxx xx	16	00
Ld	x	F	00110 xxxxx 00	32	00
St	x	F	00111 xxxxx 00	32	00
Ldb	x	F	01000 xxxxx 00	32	00
Stb	x	F	01001 xxxxx 00	32	00
Jalr	x	F	01010 xxxxx 00	32	00
Bz	x	F	01011 xxxxx 00	32	00
Bnz	x	F	01100 xxxxx 00	32	00
Movi	x	F	01101 xxxxx 00	32	00
Movhi	x	F	01110 xxxxx 00	32	00
In	x	F	01111 xxxxx 00	32	00
Out	x	F	10000 xxxxx 00	32	00
			10001 xxxxx 00	32	00
Nop	x	F	10010 1111x 00	64	00
			101xx xxxxx 00	128	00
			11xxx xxxxx 00	256	00

Tabla 1.1. Contenido de la ROM Q+ en tres tablas con formatos diferentes pero la misma información

5. Cambios marcados de la tabla ROM Q+

Contenido de la RQ+ori del SISC vN plus sobre la que marcar los cambios

[illegible]

\leftarrow

(E11)	13	13	13	13
	\downarrow	\downarrow	\downarrow	\downarrow
	02	02	03	03

6. Cambios marcados sobre la RQ+ori

Q	I	Q*	94949494 1451451426	Q* (Hexa)	# veces	Q* (Hexa)
D	A9 (1011)	Col-RP	00000	x x x x x	D	
D	Car (1100)	Col-RP	00001	1 0 1 1 x	Col-RP	
Col-RP	x	U-Mem [RP]	00012	1 0 1 1 x	U-Mem [RP]	
			00013	1 0 1 1 x	A)	
			00002	x x x x x	F	
			00000	x x x x x	D	
			00001	1 1 0 0 x	Col-RP	
			00012	1 1 0 0 x	U-Mem [RP]	
			00013	1 1 0 0 x	Car	
			00003	x x x x x	F	

7. *RQ+new*

