Pràctica 6:

Per tal d'implementar les noves instruccions (7 aritmètico-lògiques i 5 de comparació) crearem tres nous estats (Fig. 1), Donat que el camí crític per accedir a memòria i fer la operació en un mateix cicle podria esdevenir en erros, haurem de crear estat intermig per carregar un valor de memòria a un registre (Ldd) i després realitzar la operació corresponent (Ald i Cmpd).

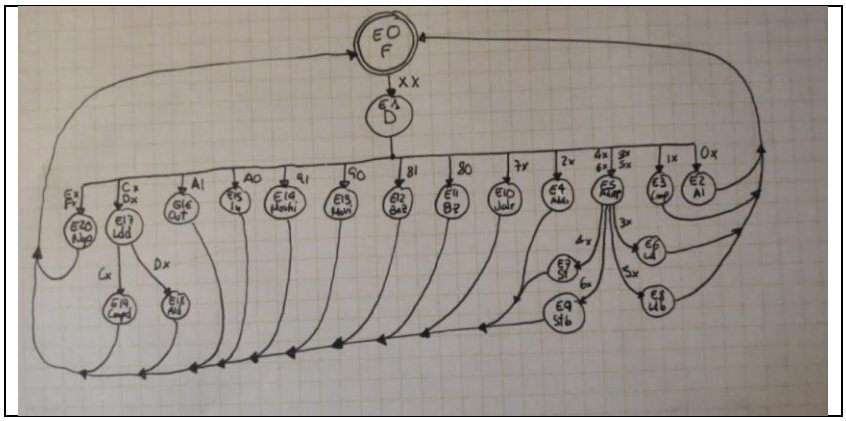

Fig. 1 Graf d'estats

Ara bé, tenint en compte que per accedir a la memòria no cal realitzar cap suma, sinó que s'accedeix directament a la adreça que hi ha al Rb podem col·locar un multiplexor com a drecera. A més a més també s'ha afegit un altre multiplexor per poder operar directament amb el valor de B sense necessitat de passar per un registre (Disseny a la Fig. 2).

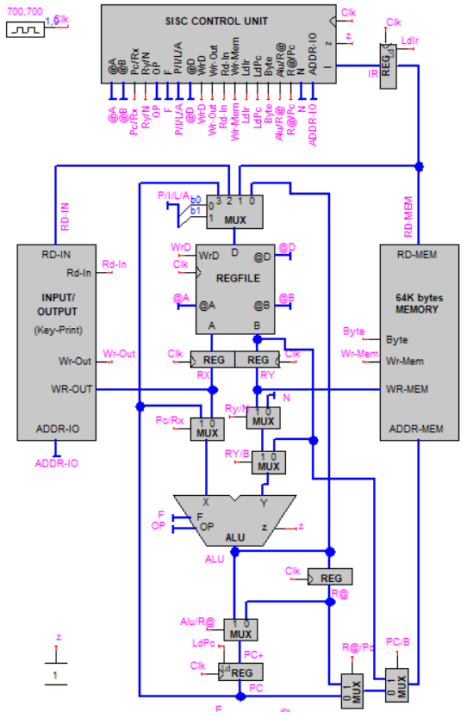

Fig. 2

Els 2 bits que necessitarem per seleccionar quin bus ha de passar el generarem amb la ROM_OUT (Fig. 4), per aconseguir que les instruccions originals haurem de decidir quin valor tindran aquests 2 bits per a cadascuna de les altres instruccions (es pot observar a la Fig. 3), a part del valor de tots els altres bits pels nous estats. El nou contingut de la ROM_OUT es pot observar a la (Fig. 5).

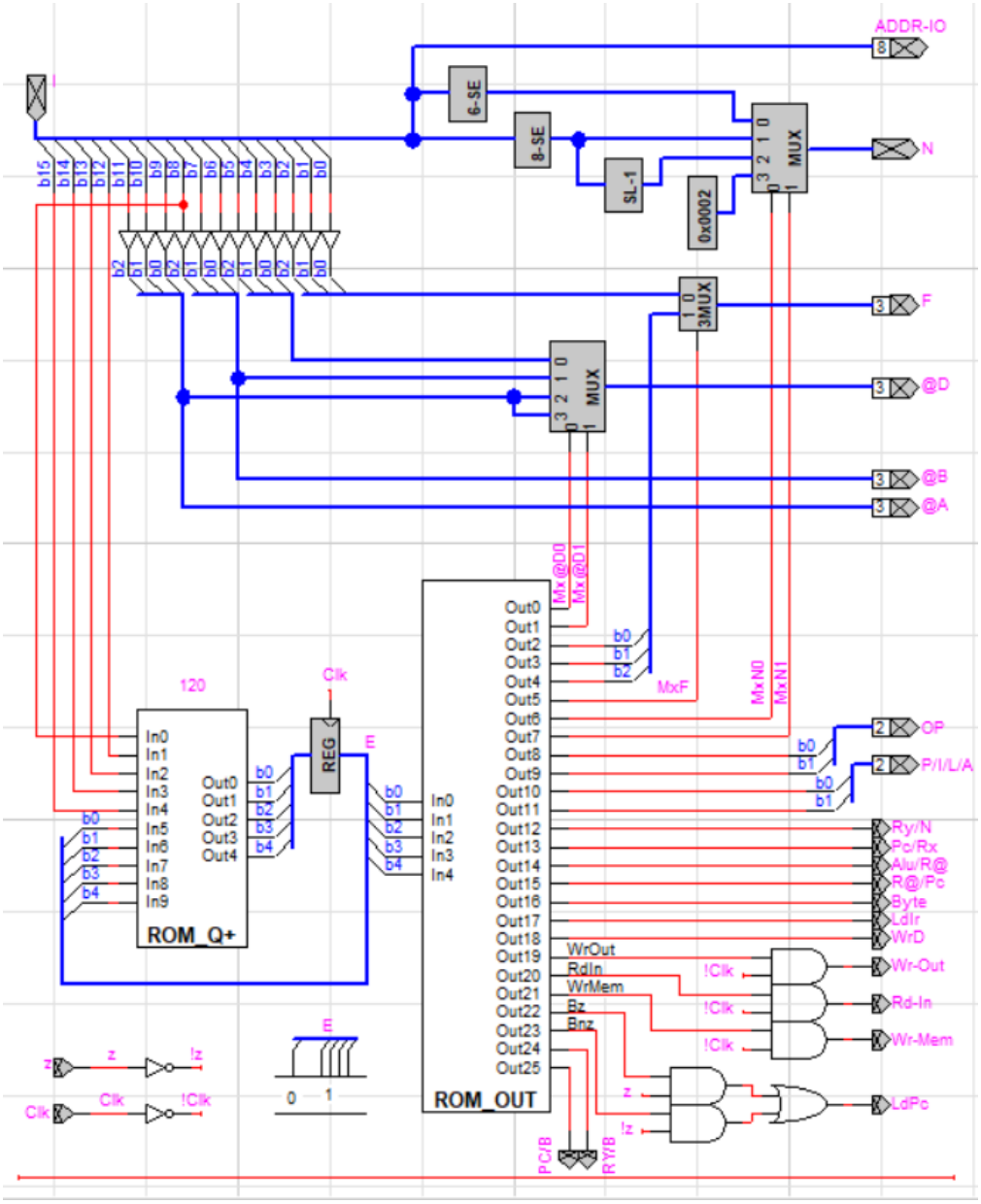

Fig. 4

@ROM	PC/B	RY/B	Bnz	Bz	WrMem	RdIn	WrOut	WrD	LdIr	Byte	R@/PC	Alu/R@	Pc/Rx	Ry/N	P/I/L/A 1	P/I/L/A 0	OP1	OP0	MxN1	MxN0	MxF	F2	F1	F0	Mx@D1	Mx@D0	
0	0	1																									F
1	0	1																									D
2	0	1																									Al
3	0	1																									Cmp
4	0	1																									Addi
5	0	1																									Addr
6	0	1																									Ld
7	0	1																									St
8	0	1																									Ldb
9	0	1																									Stb
10	0	1																									Jalr
11	0	1																									Bz
12	0	1																									Bnz
13	0	1																									Movi
14	0	1																									Movhi
15	0	1																									In
16	0	1																									Out
17	1	X	0	0	0	0	0	1	0	0	X	X	X	X	0	1	X	X	X	X	X	X	X	X	0	1	Ldd
18	X	0	0	0	0	0	0	1	0	X	X	X	0	X	0	0	0	0	X	X	0	X	X	X	0	0	Ald
19	X	0	0	0	0	0	0	1	0	X	X	X	0	X	0	0	0	1	X	X	0	X	X	X	0	0	Cmpd
20...31	X	X																									Nop

Fig. 3 Valors de la ROM_OUT (les caselles pintades mantenen el seu valor original)

F	0x1C260F0
D	0x10020B0
Al	0x1041000
Cmp	0x1041100
Addi	0x1040031
Addr	0x1000030
Ld	0x1048401
St	0x1208000
Ldb	0x1058401
Stb	0x1218000
Jalr	0x1C44E2D
Bz	0x1400220
Bnz	0x1800220
Movi	0x1040266
Movhi	0x104026A
In	0x1140802
Out	0x1080000
Ldd	0x2040401
Ald	0x0040000
Cmpd	0x0040100
Nop	0x0000000

Fig. 5 Contiguts de la ROM_OUT (en hexadecimal)

Tanmateix haurem de modificar la ROM_Q+ perquè quan rebi una de les noves instruccions el següent estat sigui el desitjat. La taula amb el contingut a actualitzar de la ROM_Q+ es pot veure a la Fig. 6.

Q	I	Q+	$q_4q_3q_2q_1q_0$	$I_{15}I_{14}I_{13}I_{12}I_8$	Q+ (Hexa)
D	ALD	LDD	00001	10001	11
D	CMPD	LDD	00001	10001	11
LDD	ALD	ALD	10001	10010	12
LDD	CMPD	CMPD	10001	10011	13
ALD	X	F	10010	00000	00
CMPD	X	F	10011	00000	00

Fig. 6 Contiguts de la ROM_Q+ (Pels nous estats)

A la Fig. 7 es pot veure la paraula de control compactada per a cadascun dels nous estats que hem creat per poder executar les noves instruccions.

Estado		Accions	Palabra de control compactada
17	Ldd	Rd ← Memw[Rb]	R@/Pc=1, Byte=0, P/I/L/A=01, WrD=1, @D=IR.
18	Ald	Rd ← RX Al B	Pc/Rx=0, Ry/N=1, OP=00, F=IR, P/I/L/A=00, WrD=1, @D=IR.
19	Cmpd	Rd ← RX Cmp B	Pc/Rx=0, Ry/N=1, OP=01, F=IR, P/I/L/A=00, WrD=1, @D=IR.

Fig. 7 Paraula de control compactada pels nous estats