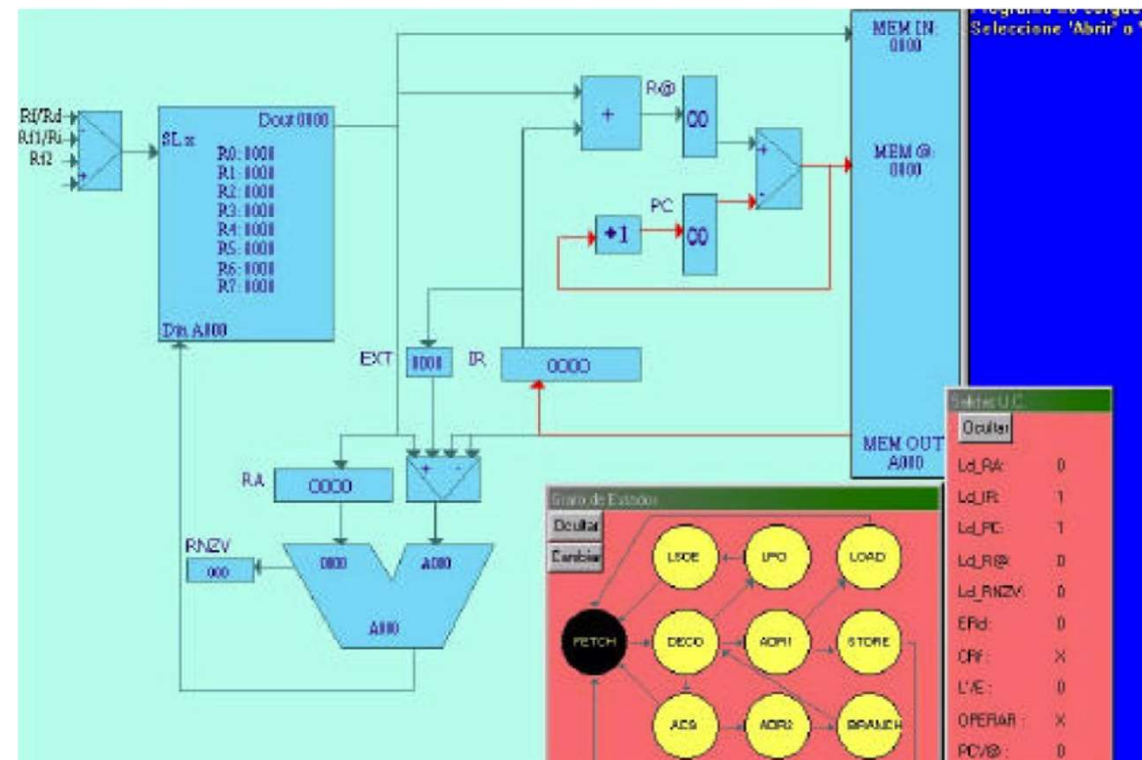


Teoricopràctic 4

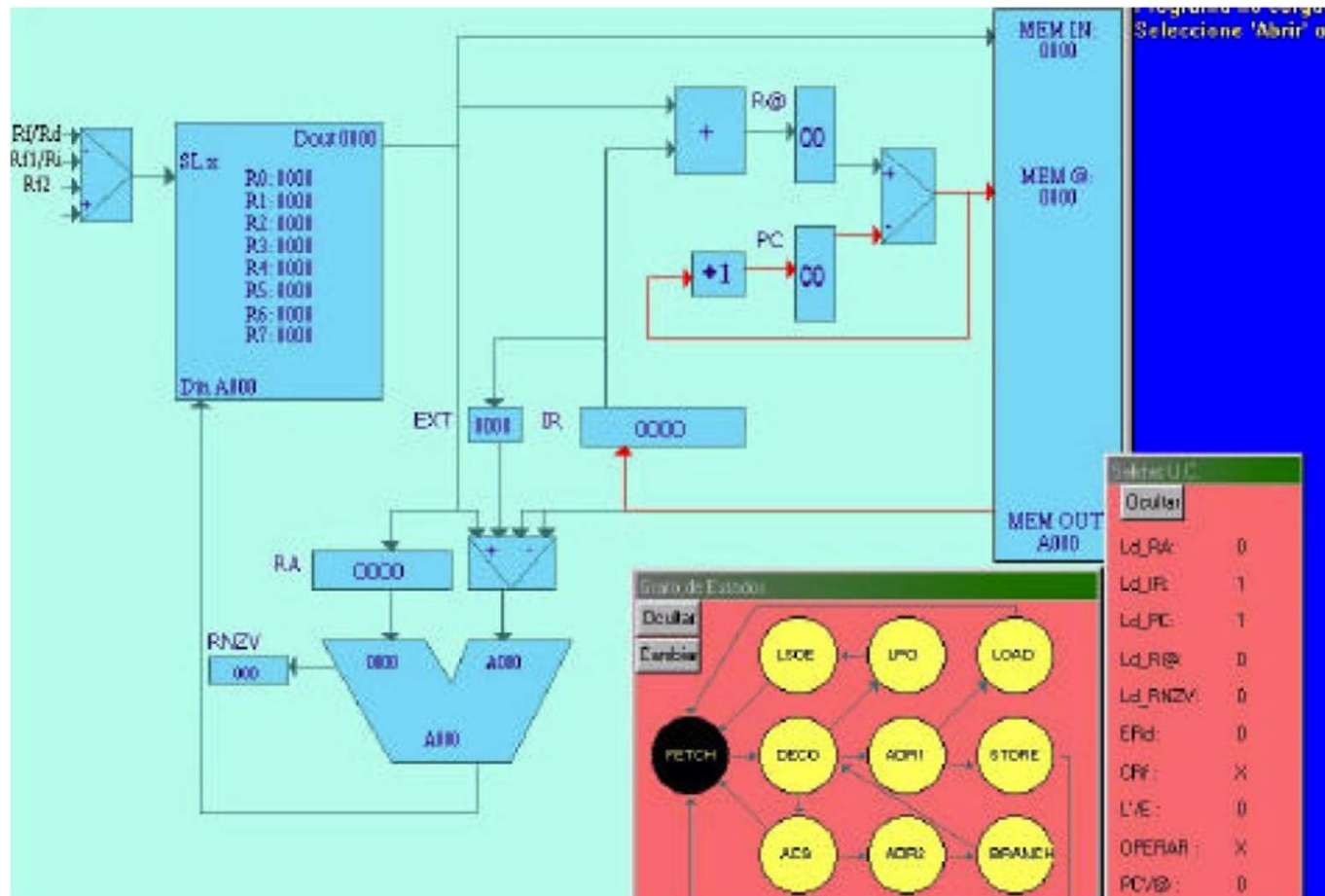
El camí de dades

Exemple de CPU senzilla

- Aquesta és una CPU de 16 bits amb 8 registres. Les instruccions que anem a estudiar són
 - LOAD R1, imm(R2)
 - ADD R3, R2, R3
 - BEQ etiqueta
- L'objectiu és entendre el disseny del camí de dades



Configuració



OP Code

N

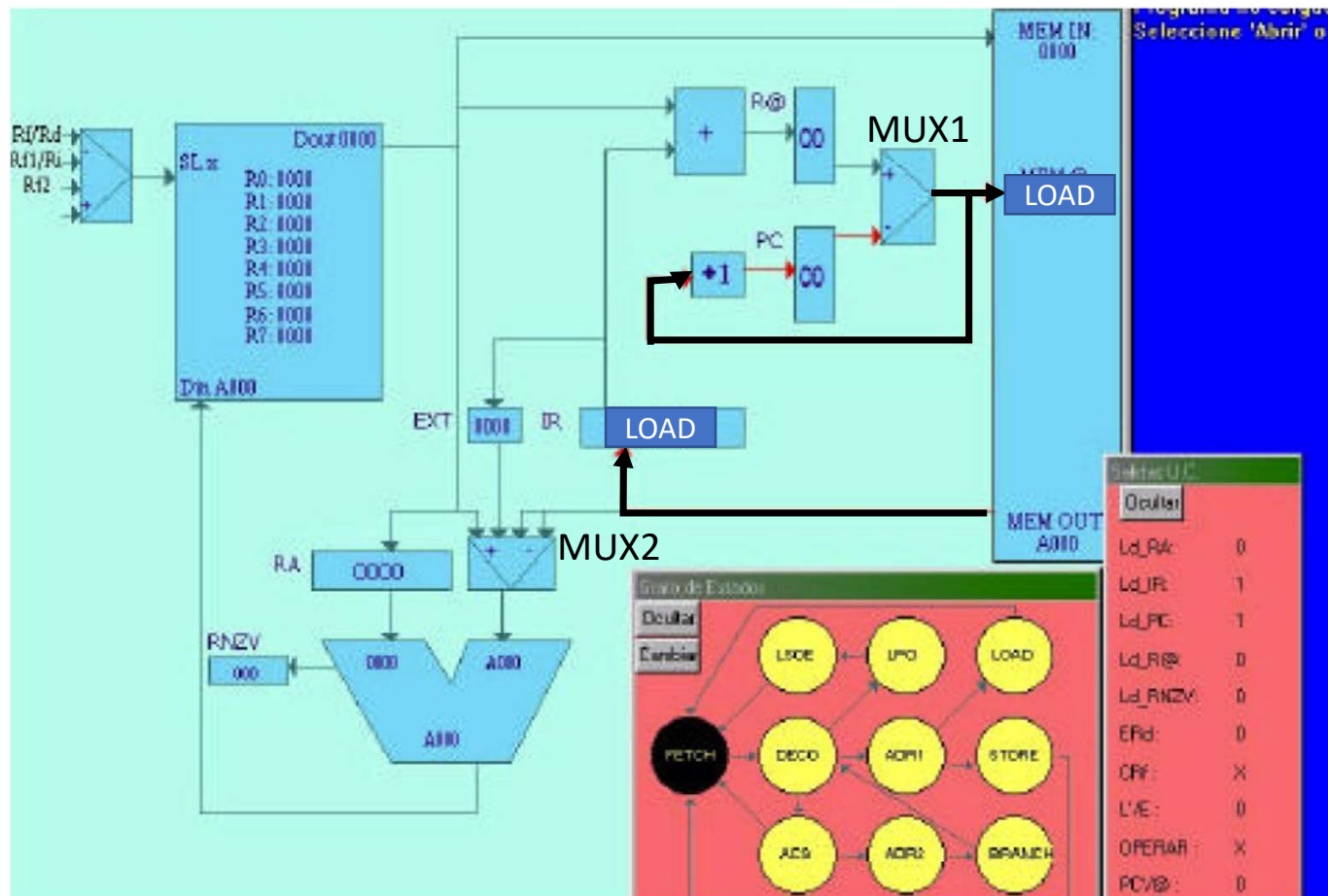
Z

V

W_Mem
MUX1
MUX2
ALU
SelR0
SelR1
SelR2
W_RF
W_PC
W_IR
W_R@
W_RA
W_RNZV
W_Ext

Subtítol: U.C.	
Ocular	
Ld_RA:	0
Ld_RB:	1
Ld_PC:	1
Ld_RS@:	0
Ld_RN@M:	0
EPd:	0
OPF:	X
L'VE:	0
OPERAR:	X
PCV@:	0

LOAD R1, imm(R2)



Fase FETCH o càrrega de la Instrucció.

Registres implicats:

PC, IR

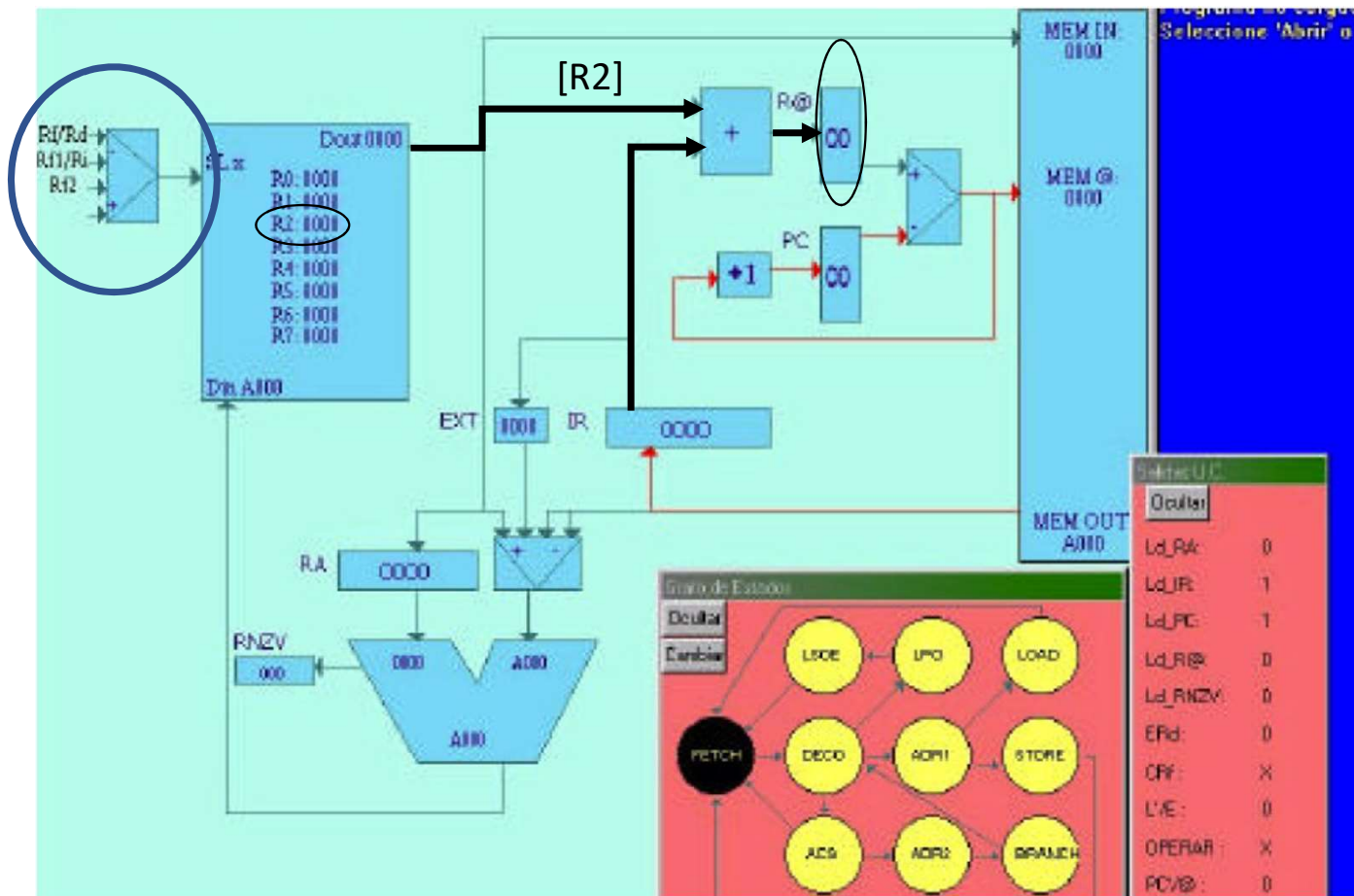
-Accés a memòria (LECTURA)

-Escriptura de PC

-Escriptura de IR

(1 cycle)

LOAD R1, imm(R2)



Fase ACCÉS previ a memòria

Registres implicats:

R@, R2, IR

-Accés al banc de registres

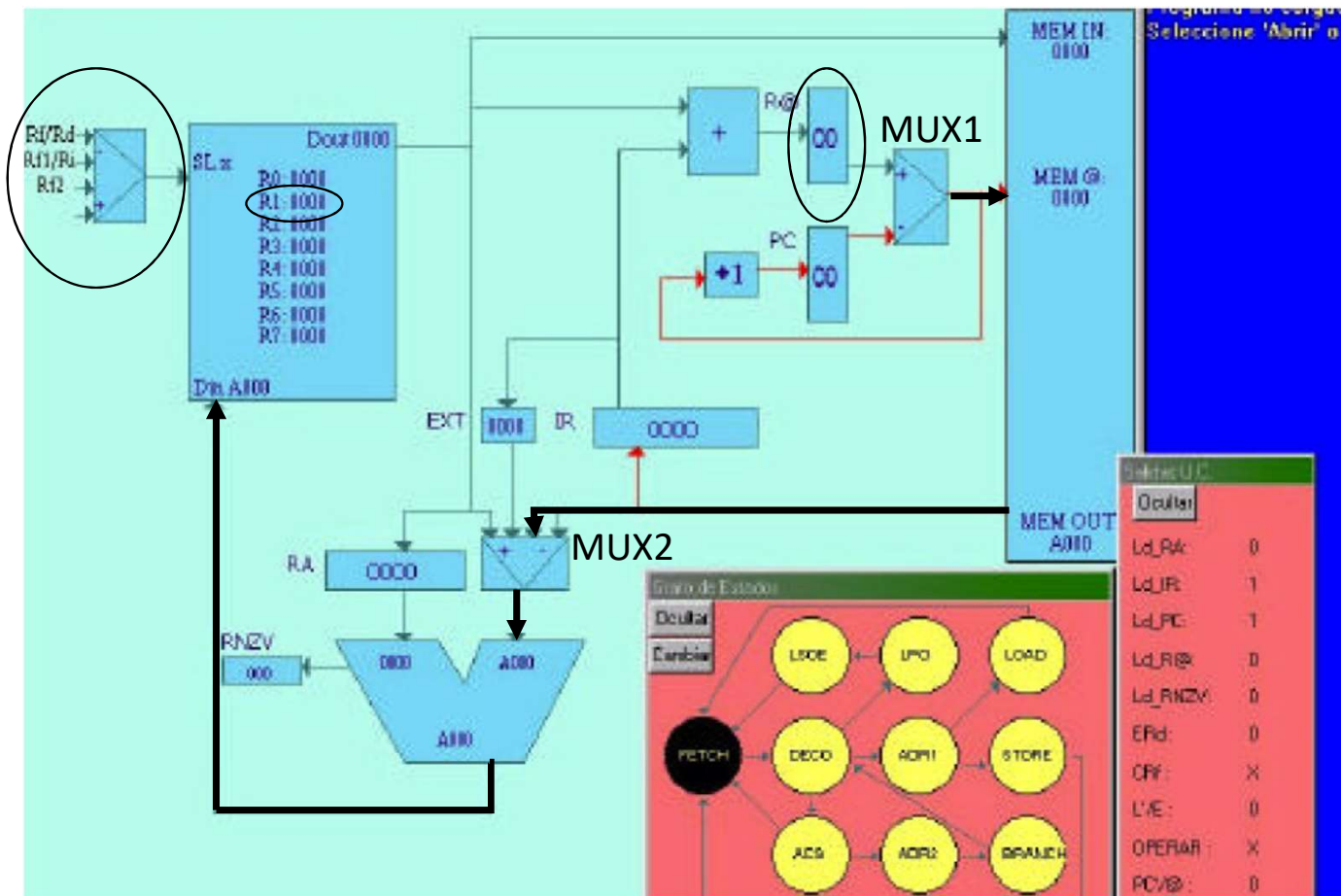
-Escriptura de R@

-Lectura R2

-L'imm ve del IR

(1 cicle)

LOAD R1, imm(R2)



Fase ACCÉS a memòria i
exec

Registres implicats:

R@, R2, IR

-Accés al banc de registres

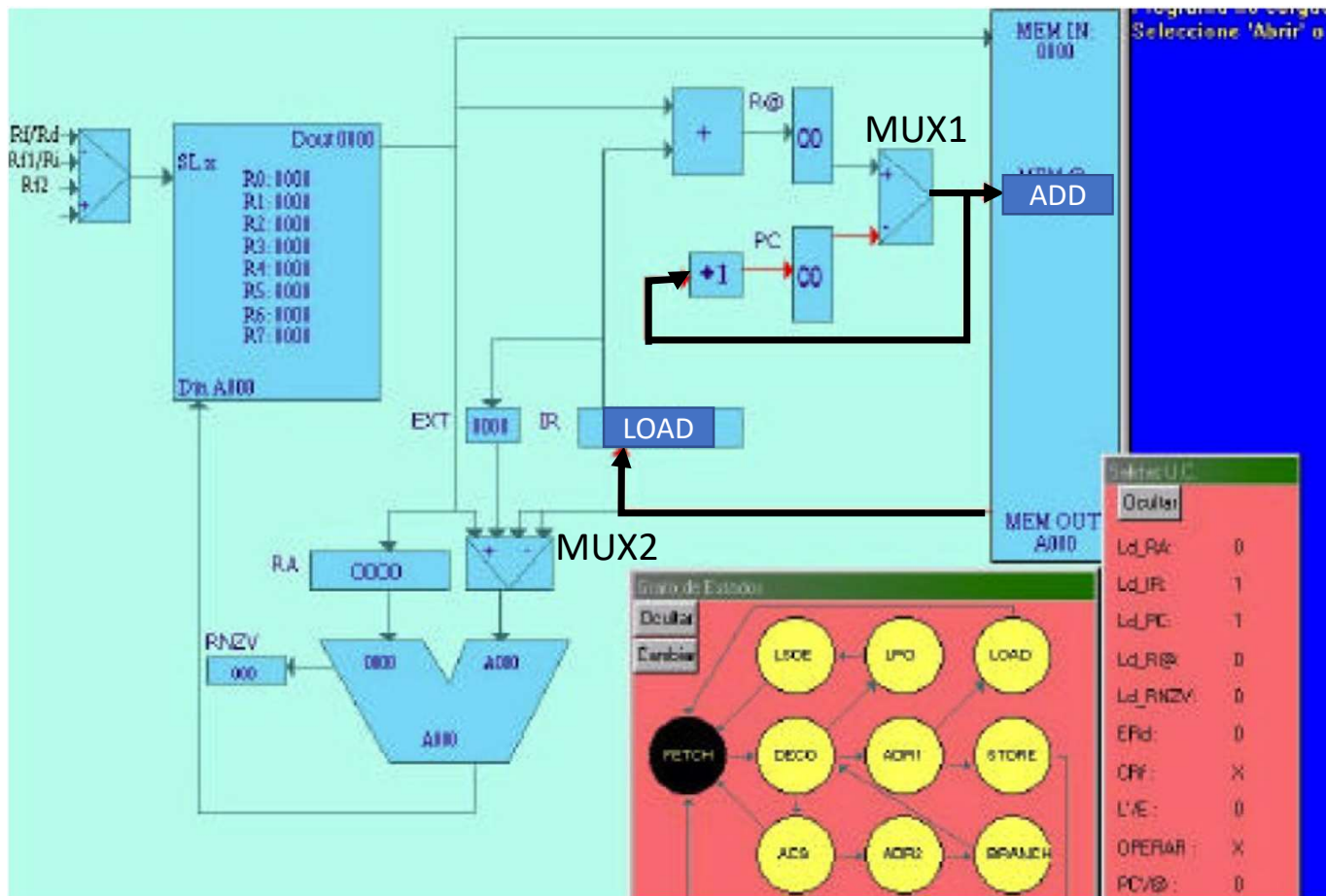
-Escriptura de R1

-Lectura R@

-ALU => permet el pas

(1 cicle)

ADD R3,R2,R3



Fase FETCH o càrrega de la Instrucció.

Registres implicats:

PC, IR

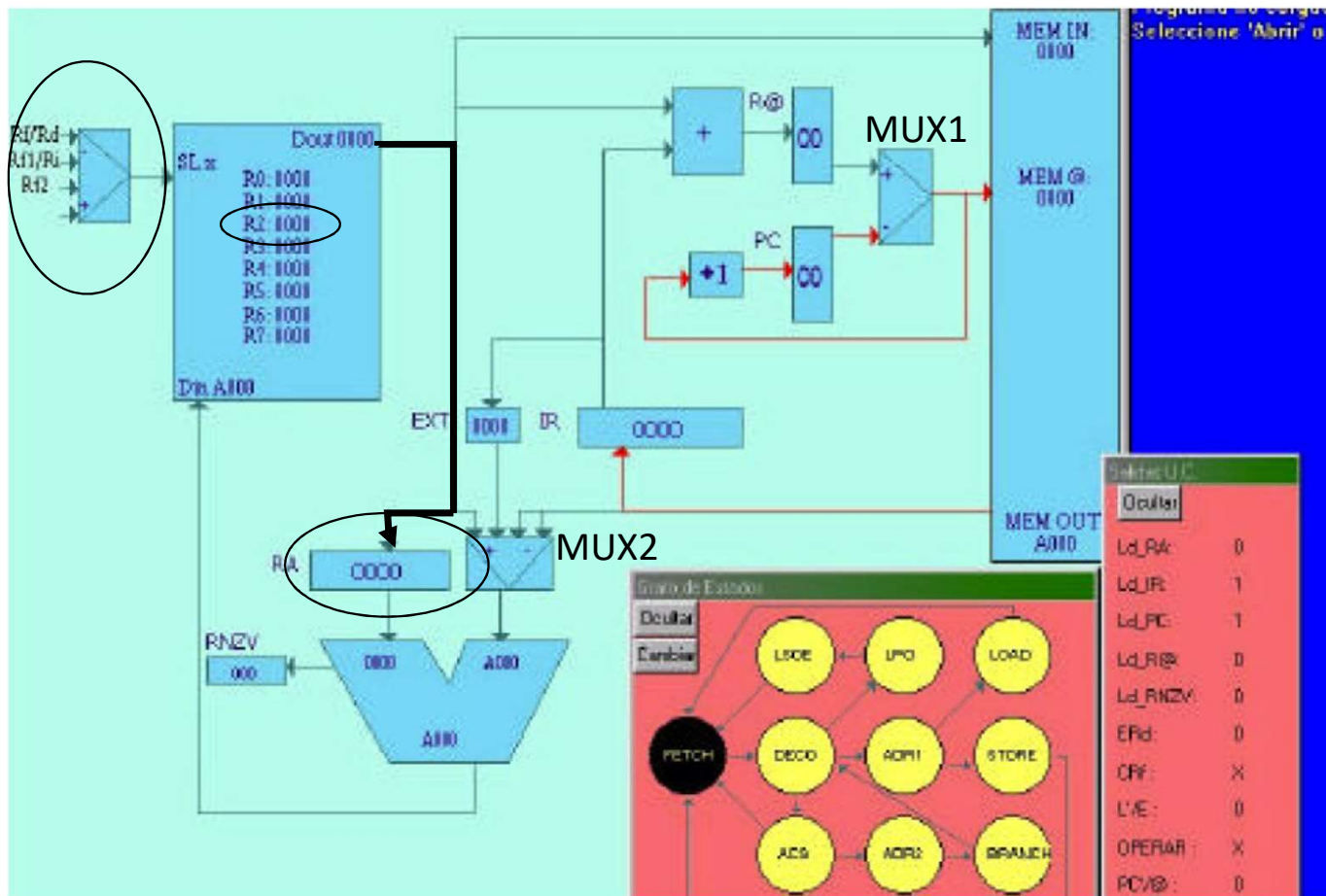
-Accés a memòria (LECTURA)

-Escriptura de PC

-Escriptura de IR

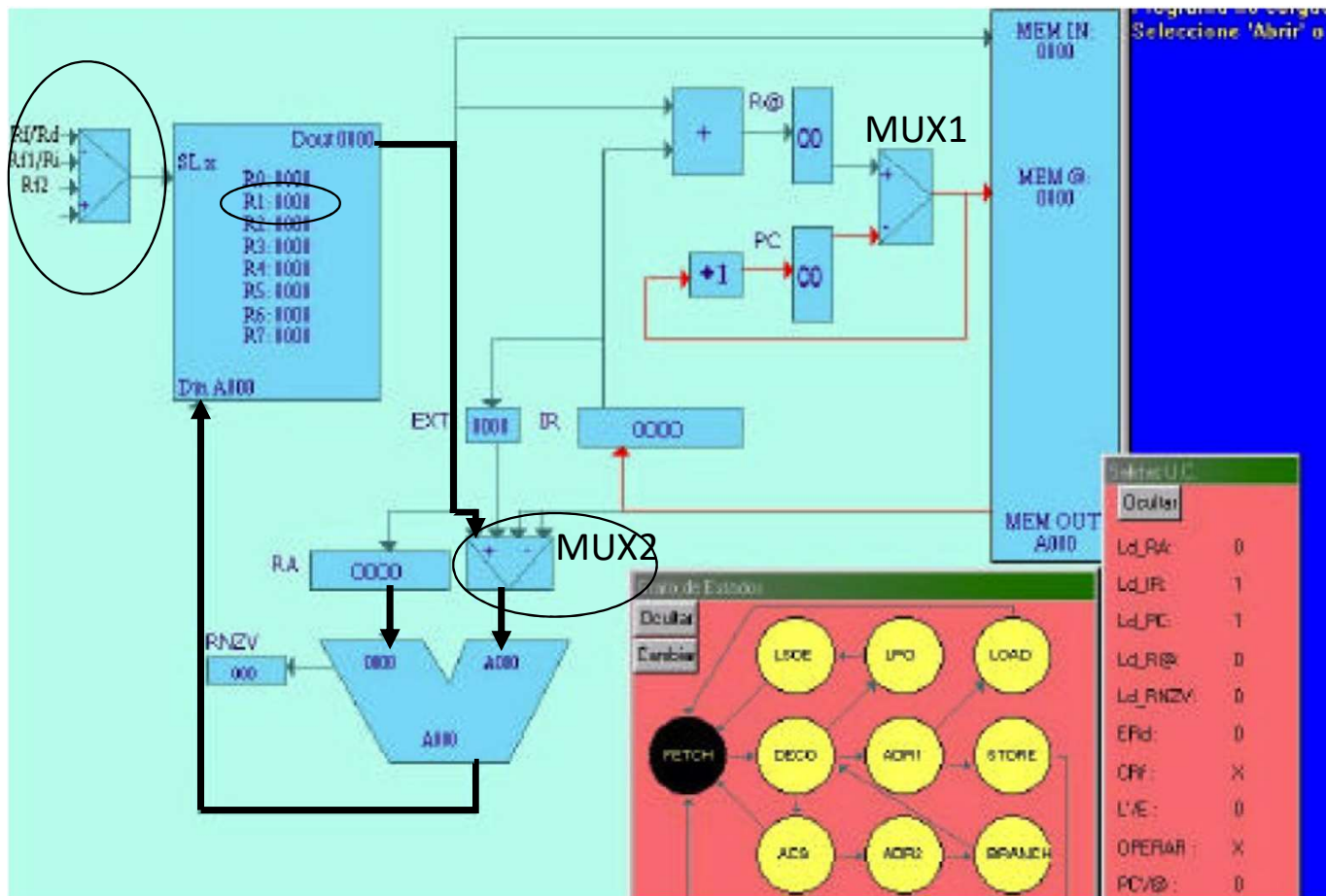
(1 cicle)

ADD R3,R2,R3



Busquem R2
 Registres implicats:
 ACC, R2
 -Accés a banc de registres
 -Escriptura en ACC
 -Lectura R2
 R2 => ACC
 (1 cicle)

ADD R3,R2,R3



Busquem R3 i executem

Registres implicats:

ACC, R3

-Accés a banc de registres

-Lectura en ACC

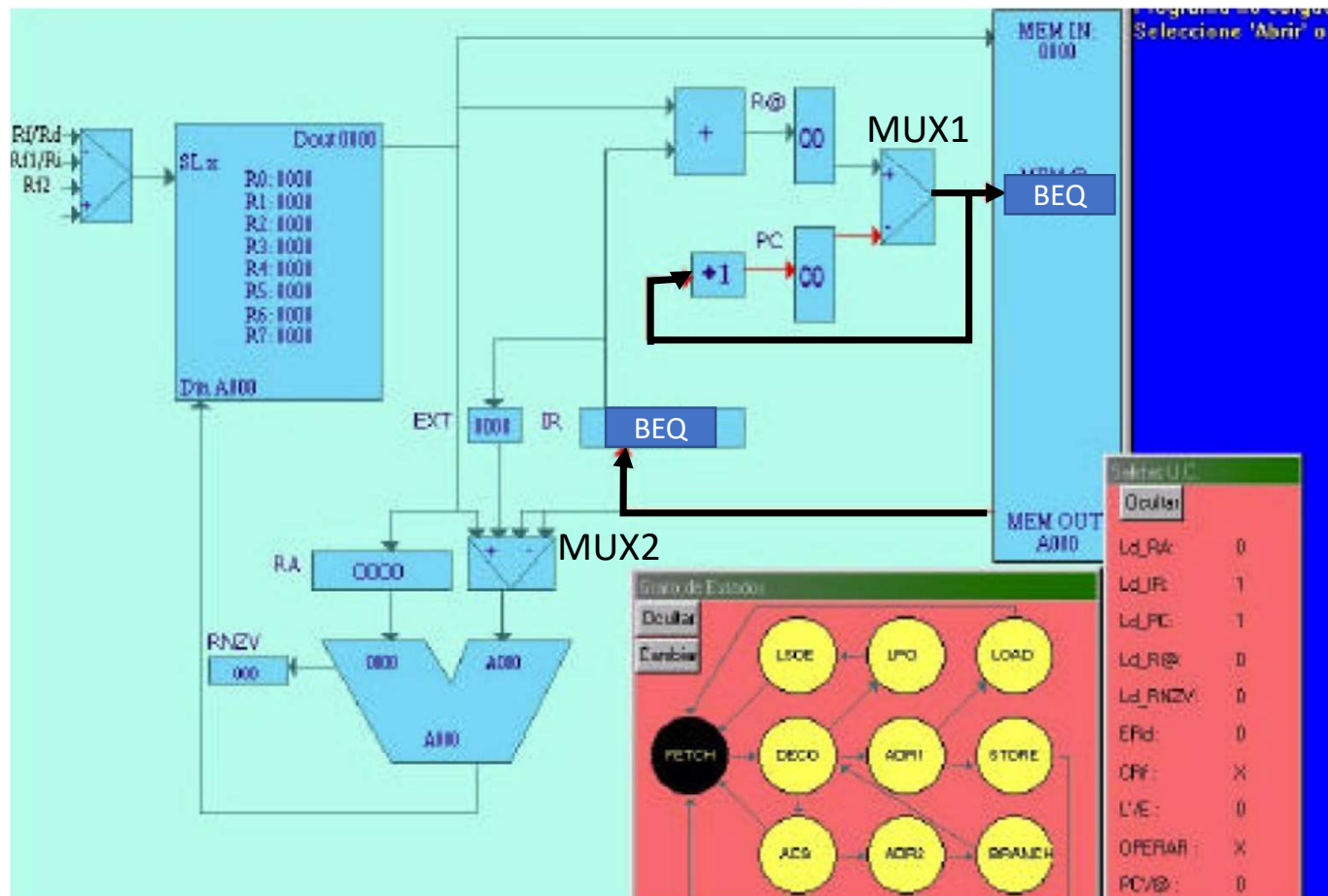
-Lectura R3 i escriptura en R3

-ALU (suma)

-Resultat ALU es guarda en R3

(1 cicle)

BEQ etiqueta



Fase FETCH o càrrega de la Instrucció.

Registres implicats:

PC, IR

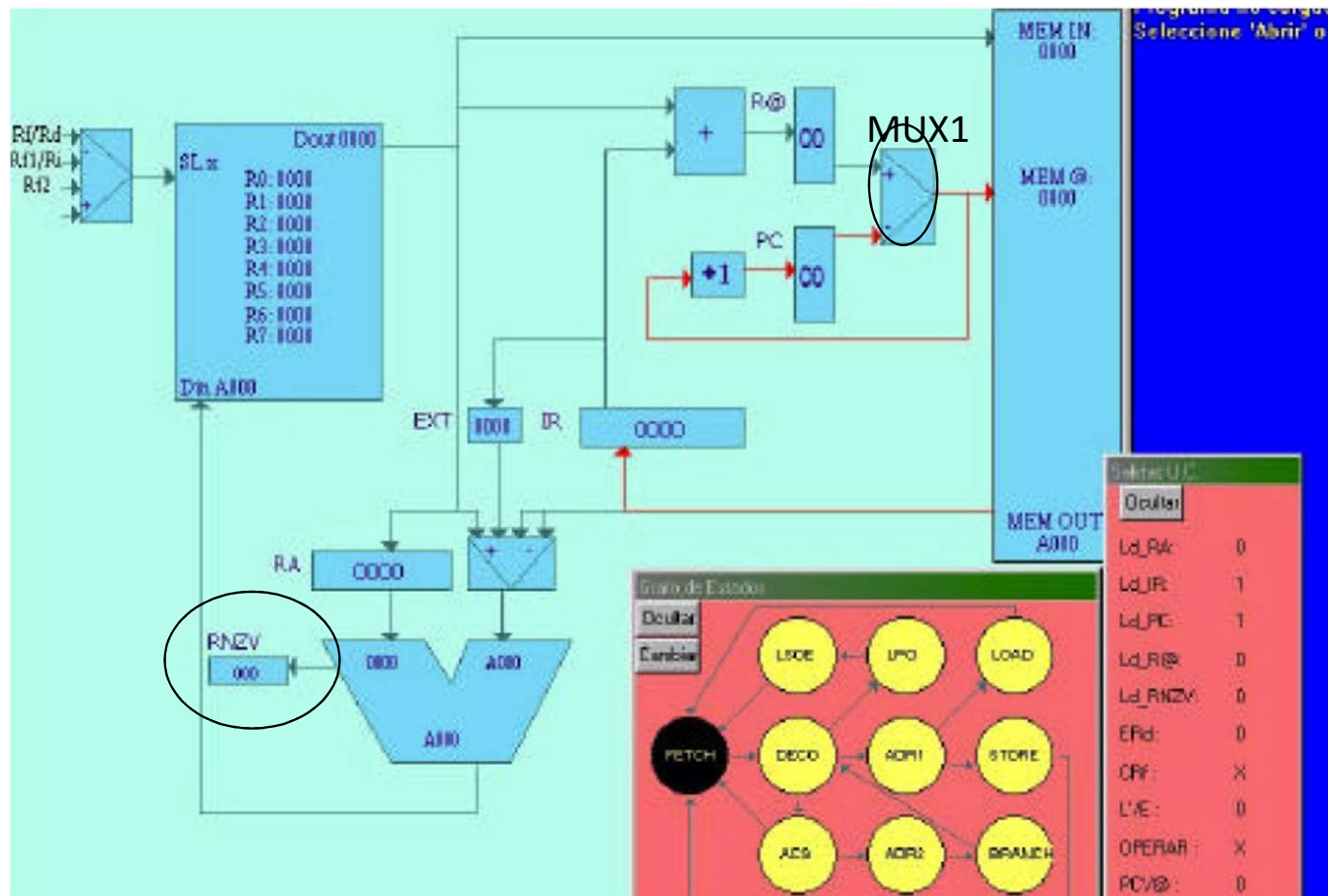
-Accés a memòria (LECTURA)

-Escriptura de PC

-Escriptura de IR

(1 cicle)

BEQ etiqueta

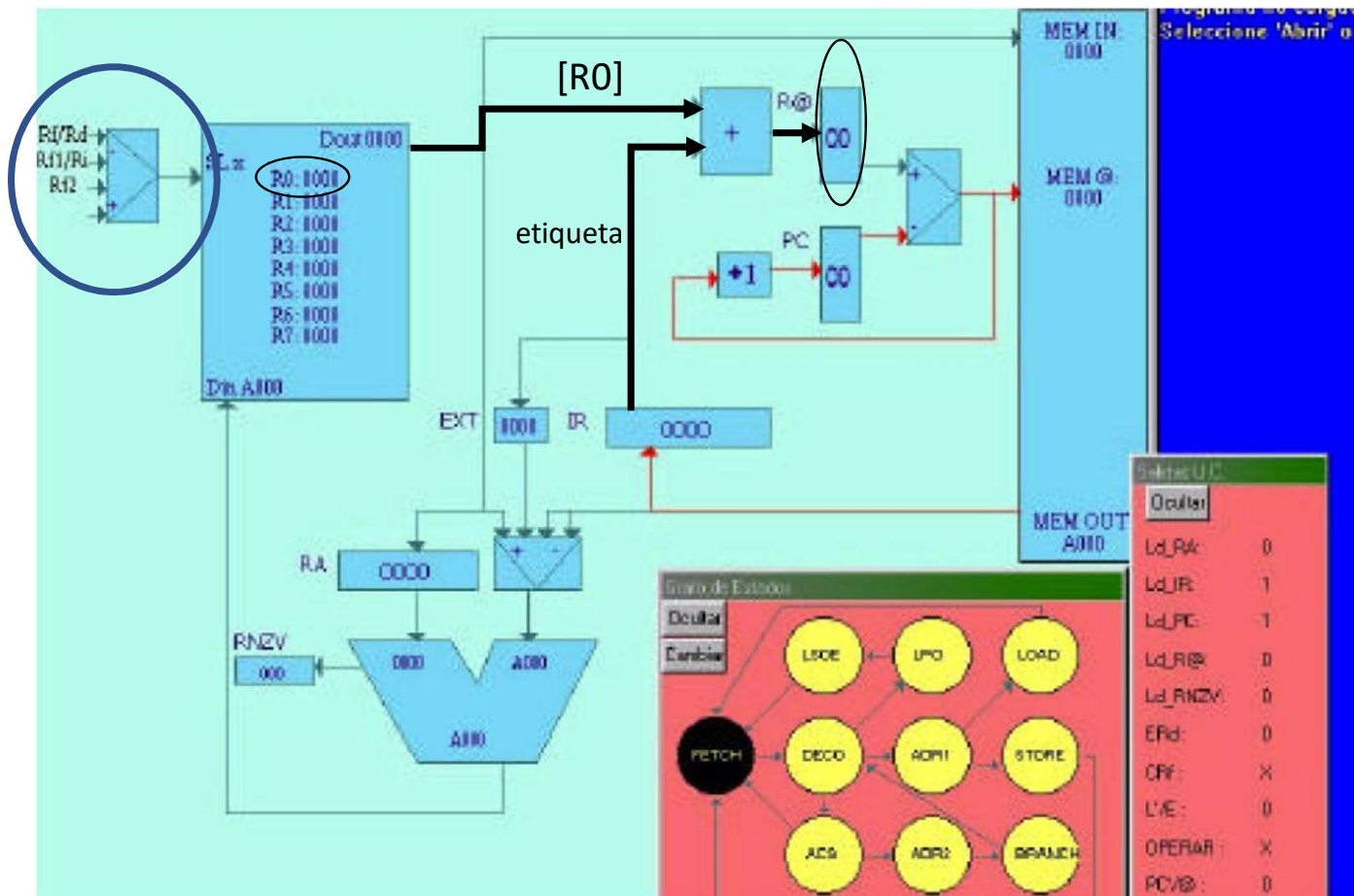


Fase de decodificació

- Analitzem el contingut de RNZV (Reg. Estat)

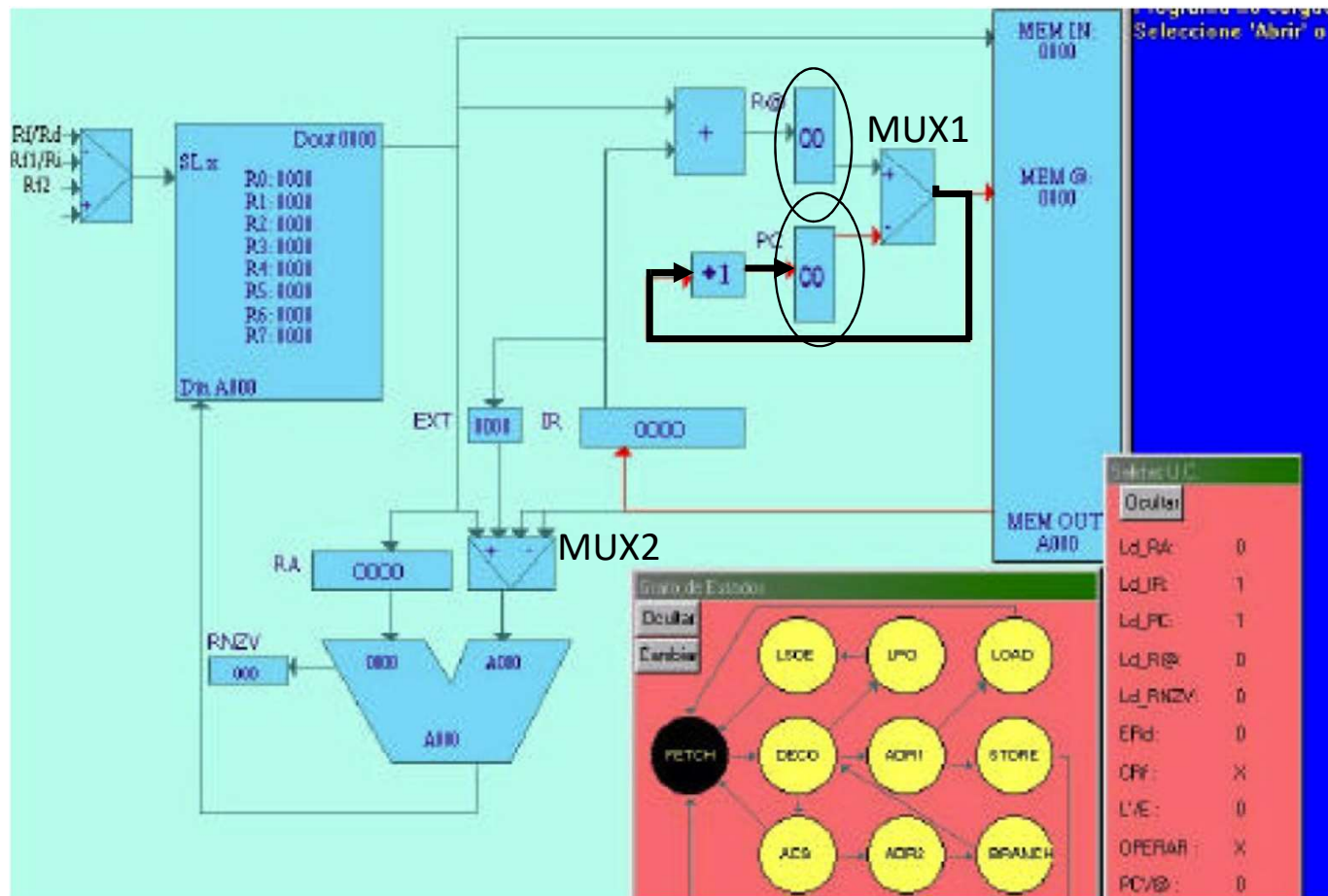
- Si $Z == 1$ MUX_SEL = 1
- Si $Z == 0$ MUX_SEL = 0

BEQ etiqueta



Fase ACCÉS previ a memòria
MUX_SEL = 1:
Registres implicats:
R@, R0, IR
-Accés al banc de registres
-Escriptura de R@
-Lectura R0
-L'etiqueta ve del IR
(1 cicle)

BEQ etiqueta



MUX_SEL = 1:
Fase ACCÉS a propera
Adreça de memòria
Registres implicats:
R@, PC
-Escriptura de PC
-Lectura R@
(1 cicle)

Single Cycle RISC-V Processor

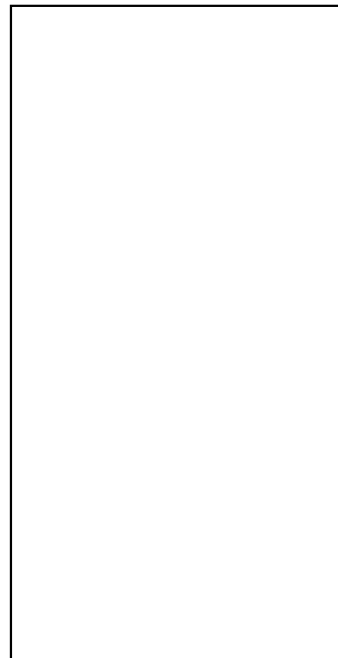
- Analitzem ara el nostre microprocessador...

ENTRADES

Opcode (6 bits)

Funct3 (3 bits)

Funct7 (7 bits)



SORTIDES

MUX1

MUX2

MUX3 (from Branch module)

MUX4 (from Branch module)

MUX5 (2 bits)

W_PC

W_RF

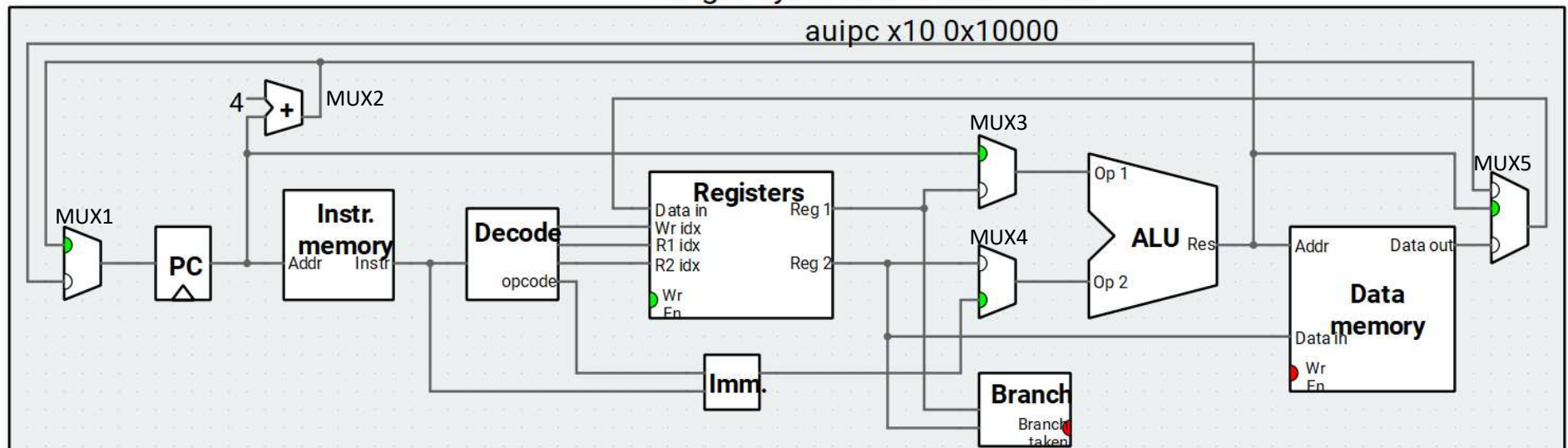
W_Mem

ALU (x bits)

Dec_Reg (5 bits)

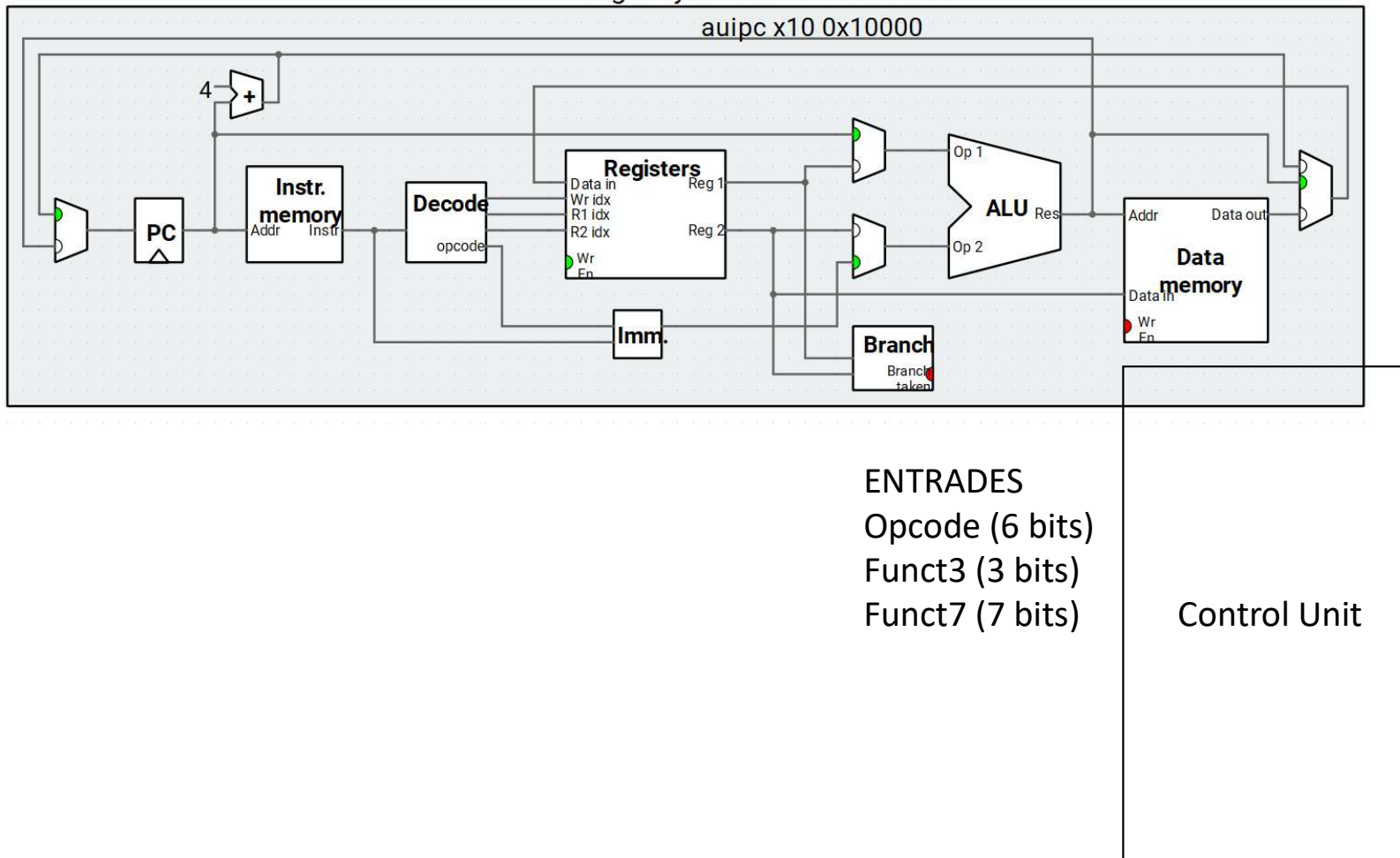
31	30	25	24	21	20	19	15	14	12	11	8	7	6	0				
funct7				rs2			rs1		funct3		rd			opcode		R-type		
imm[11:0]						rs1		funct3		rd			opcode		I-type			
imm[11:5]				rs2			rs1		funct3		imm[4:0]			opcode		S-type		
imm[12]		imm[10:5]			rs2			rs1		funct3		imm[4:1]		imm[11]		opcode		B-type
imm[31:12]									rd			opcode			U-type			
imm[20]		imm[10:1]				imm[11]		imm[19:12]			rd			opcode		J-type		

Single Cycle RISC-V Processor



Single Cycle RISC-V Processor

Single Cycle RISC-V Processor



ENTRADES

Opcode (6 bits)

Funct3 (3 bits)

Funct7 (7 bits)

Control Unit

SORTIDES

MUX1

MUX2

MUX3 (from Branch module)

MUX4 (from Branch module)

MUX5 (2 bits)

W_PC

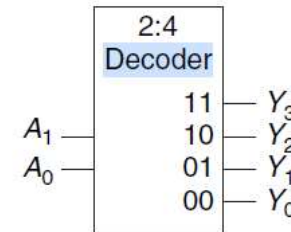
W_RF

W_Mem

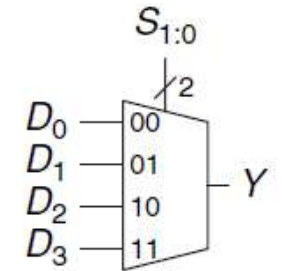
ALU (x bits)

Dec_Reg (5 bits)

lw a1, offset(a2)



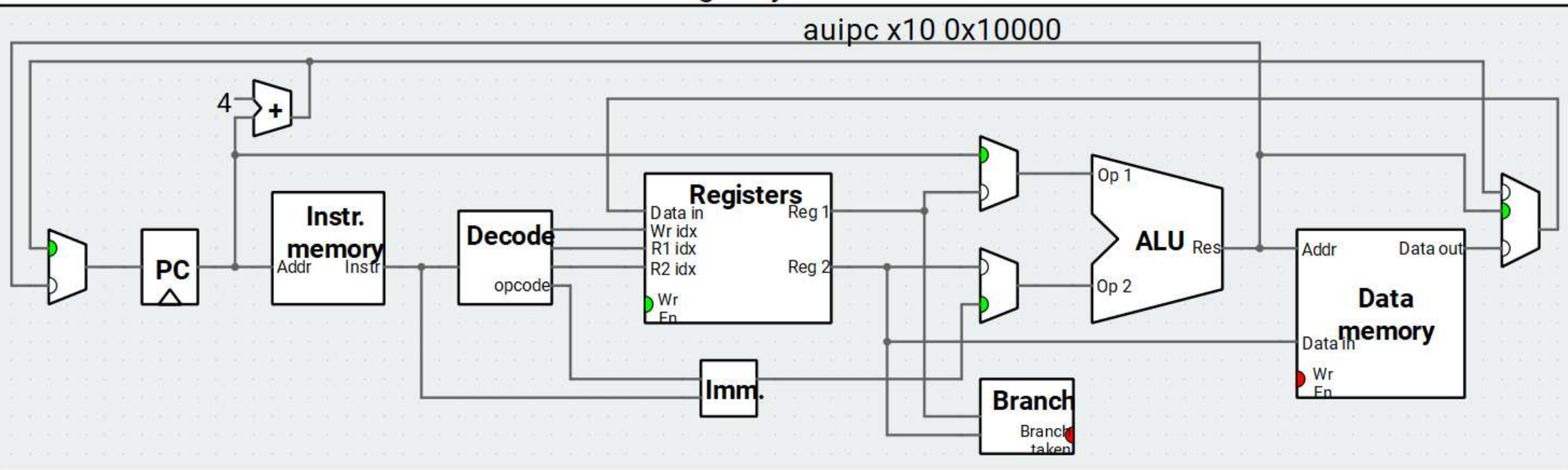
A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



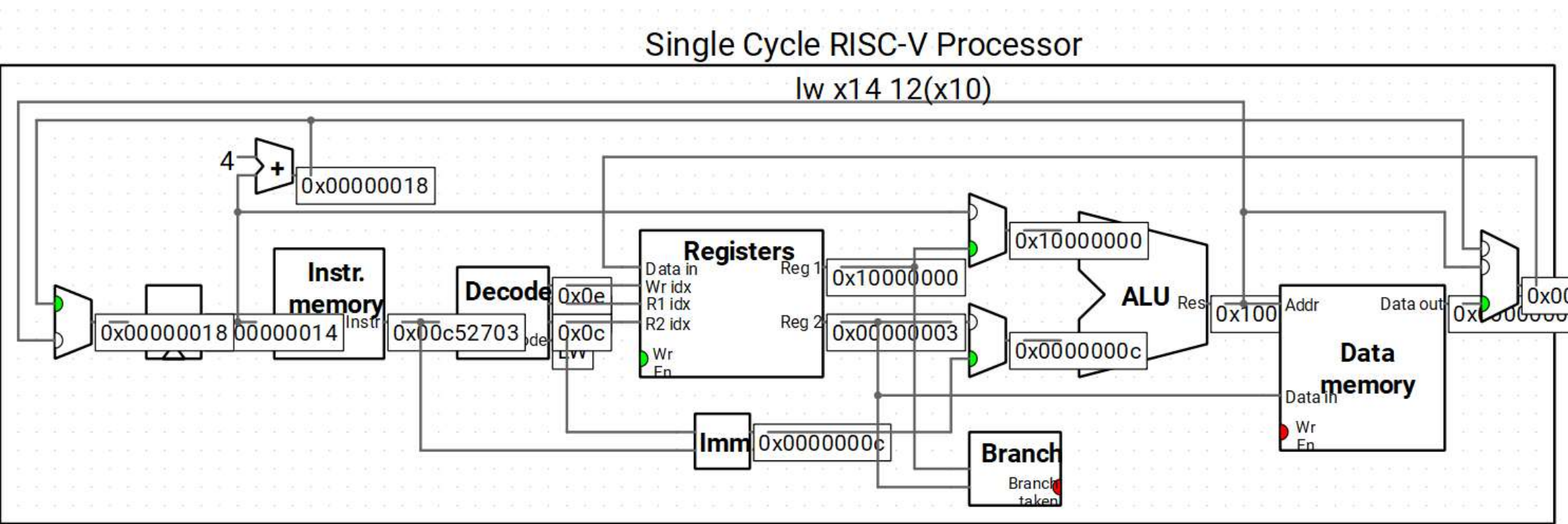
4:1 multiplexer

Single Cycle RISC-V Processor

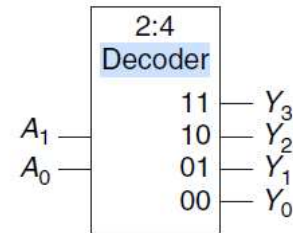
auipc x10 0x10000



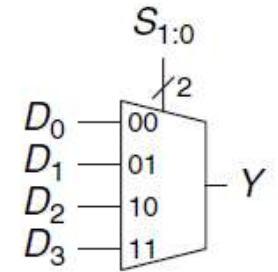
Veure-ho al simulador...



add a1, a2, a3



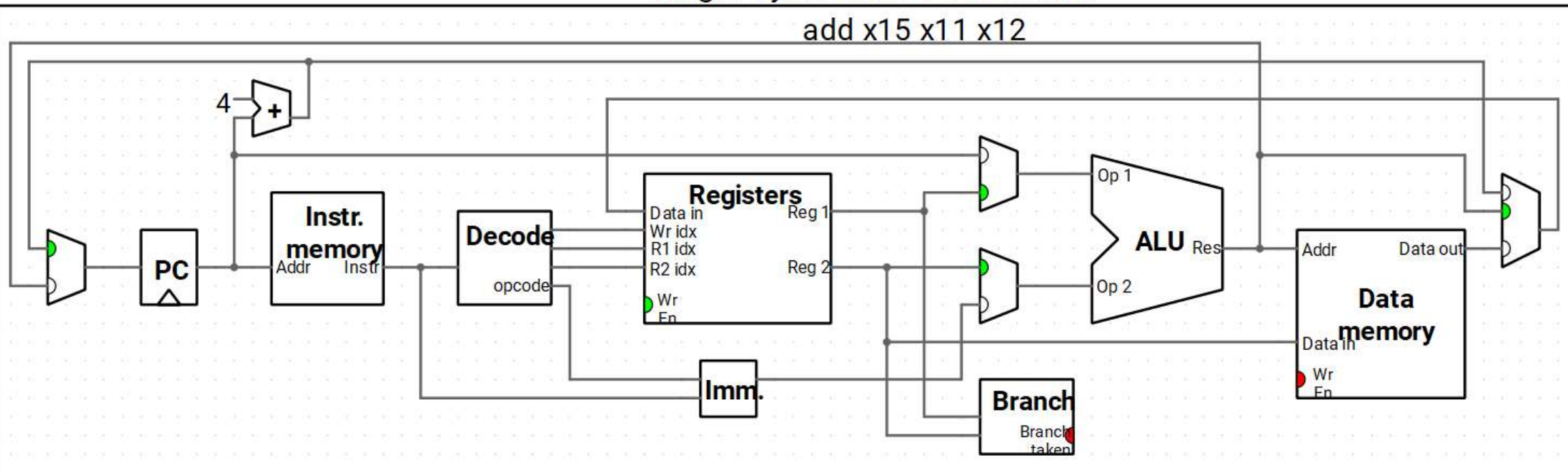
A ₁	A ₀	Y ₃	Y ₂	Y ₁	Y ₀
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



4:1 multiplexer

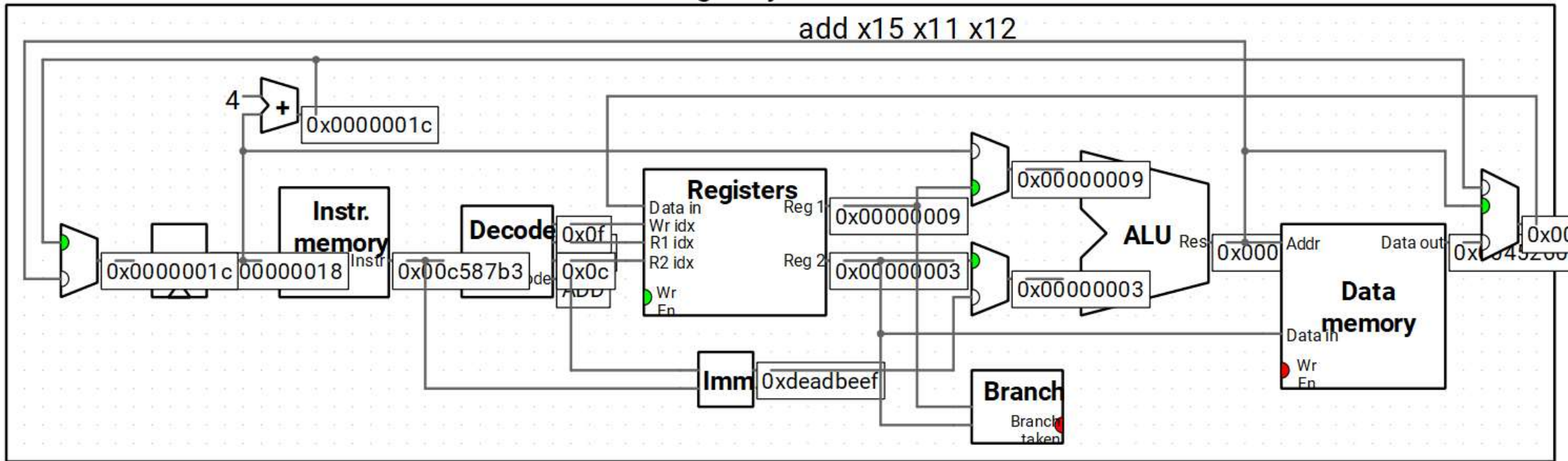
Single Cycle RISC-V Processor

add x15 x11 x12



Veure-ho al simulador

Single Cycle RISC-V Processor



Veure-ho al simulador

Single Cycle RISC-V Processor

bge x13 x0 -12 <loop>

