



Escola Politècnica Superior
d'Enginyeria de Manresa

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PRÀCTICA 5: DISSENY DEL CAMI DE DADES

Sergi Carol Bosch i Enric Lenard Uró

Grau en Enginyeria de Sistemes TIC

Arquitectura de Computadors

Curs 2013-14, Grup 10, G12 de pràctiques

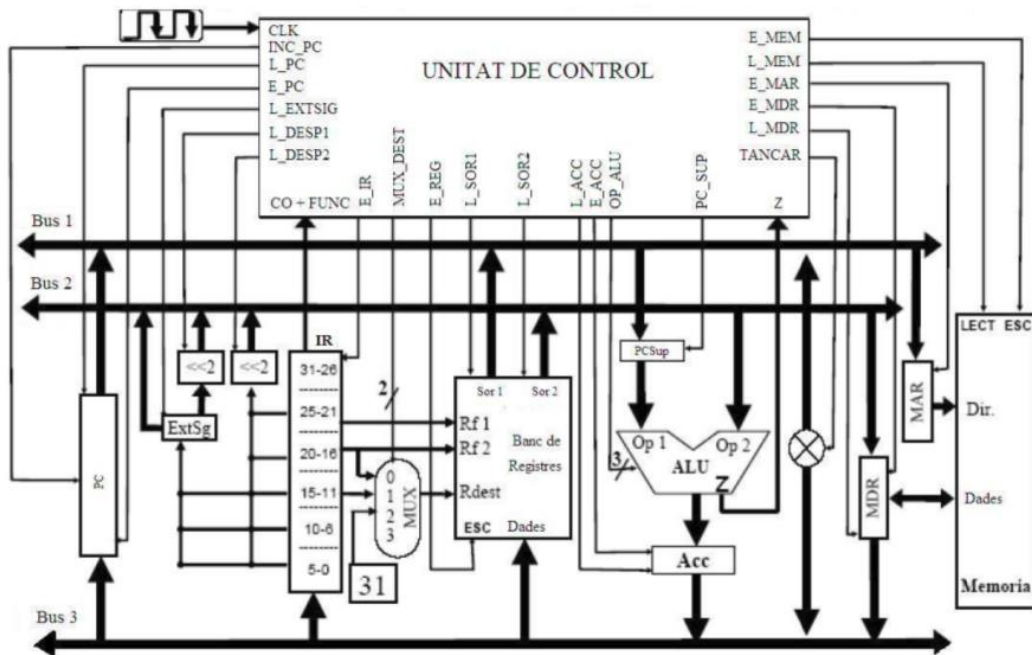
Realització de la pràctica: 12/05/2014

Lliurament del treball: 26/05/2014

REALITZACIÓ PRÀCTICA

En aquesta pràctica l'alumne ha de completar el disseny del processador que s'ha estudiat a classe. En les practiques anteriors s'han desenvolupat els elements més importants del camí de dades: l'ALU, el banc de registres, el comptador de programa i la unitat de control.

La resta d'elements que s'han d'incloure són els que observem en l'esquema següent, que es faciliten en el fitxer PRAC5.zip.



MICROPROCESSADOR (MP)

Dissenyem el codi de programa que realitza el funcionament desitjat, creant les entrades i sortides necessàries.

Primer realitzem l'entitat, on les seves entrades estaran formades per un RESET i una senyal de CLOCK.

Posteriorment adjuntem tot els arxius anteriors i creem les senyals necessàries per la seva correcta execució.

Finalment definim tots els components dels arxius anteriors (ALU, BancRegistre, CP i UC) utilitzant el PORT MAP.

ADJUNTAT DOCUMENT PDF AMB EL CODI

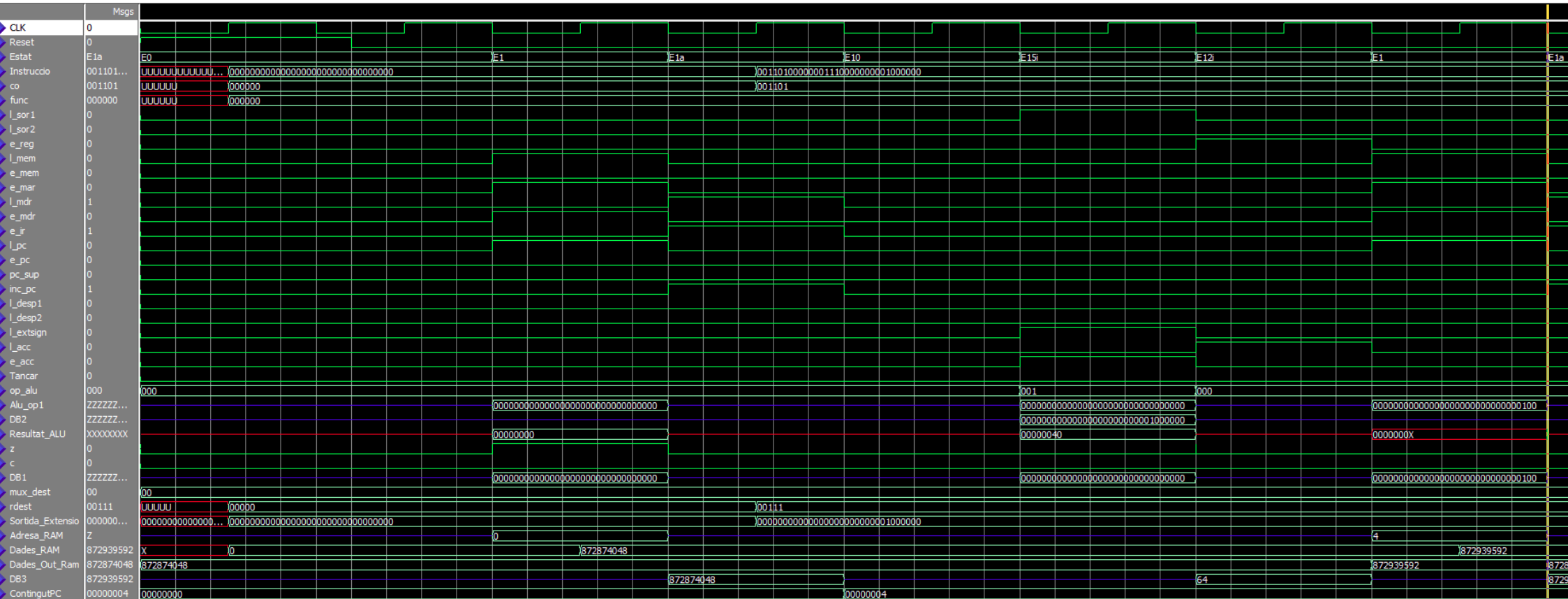
TEST BENCH MP

A continuació comprovem mitjançant la simulació que el disseny funciona correctament. Realitzem un test bench simple, format per un clock i un reset, on posteriorment afegirem els senyal necessaris per veure transició d'estats a la gràfica.

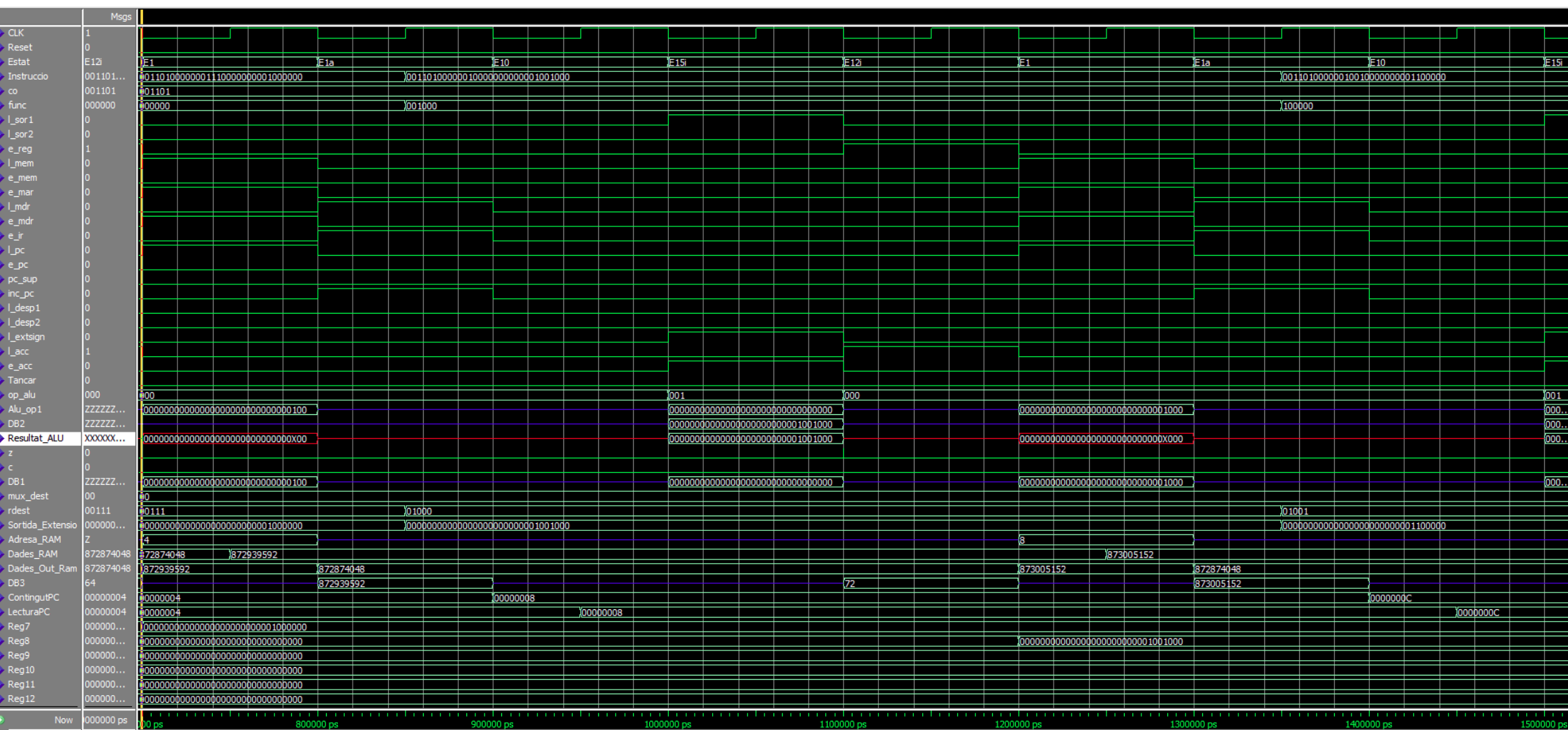
```
44     i1 : Microprocessador
45     PORT MAP (
46     -- list connections between master ports and signals
47     CLK => CLK,
48     Reset => Reset
49     );
50     init : PROCESS
51     -- variable declarations
52     BEGIN
53         -- code that executes only once
54     WAIT;
55     END PROCESS init;
56     clock : PROCESS
57     BEGIN
58         CLK <= '0';
59         WAIT FOR 50 ns;
60         CLK <= '1';
61         WAIT FOR 50 ns;
62     end process clock;
63     always : PROCESS
64     -- optional sensitivity list
65     -- (      )
66     -- variable declarations
67     BEGIN
68
69         RESET <= '1';
70         WAIT FOR 120 ns;
71         RESET <= '0';
72         WAIT FOR 20 us;
73     WAIT;
74         -- code executes for every event on sensitivity list
75     WAIT;
76     END PROCESS always;
77     END Microprocessador_arch;
78
```

SIMULACIÓ GRÀFICA MC

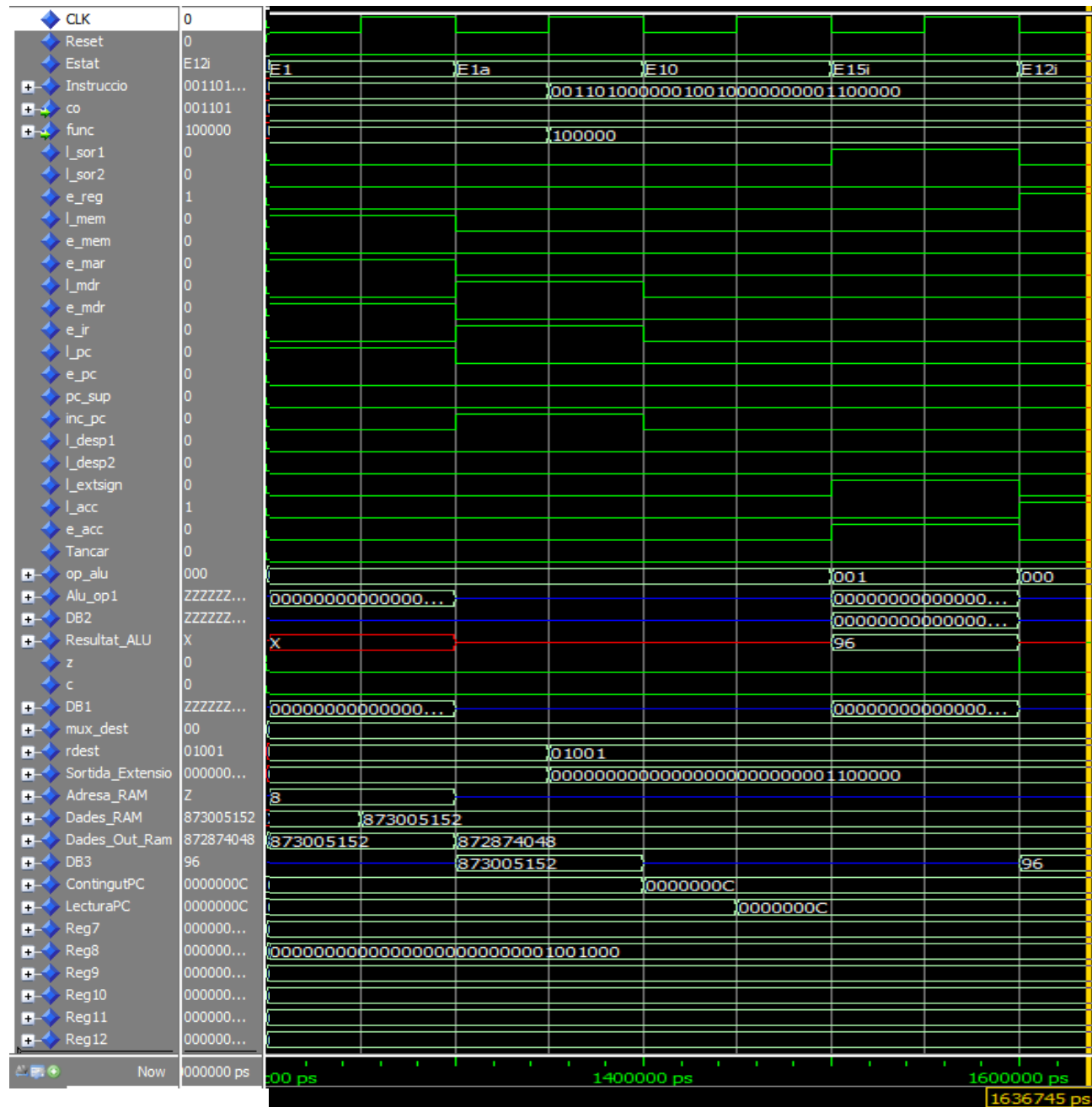
- Grafica 0x00 ori \$7, \$0, 64 ("00110100000001110000000001000000"), -- 0x00 ori \$7, \$0, 64 (0x34070040)



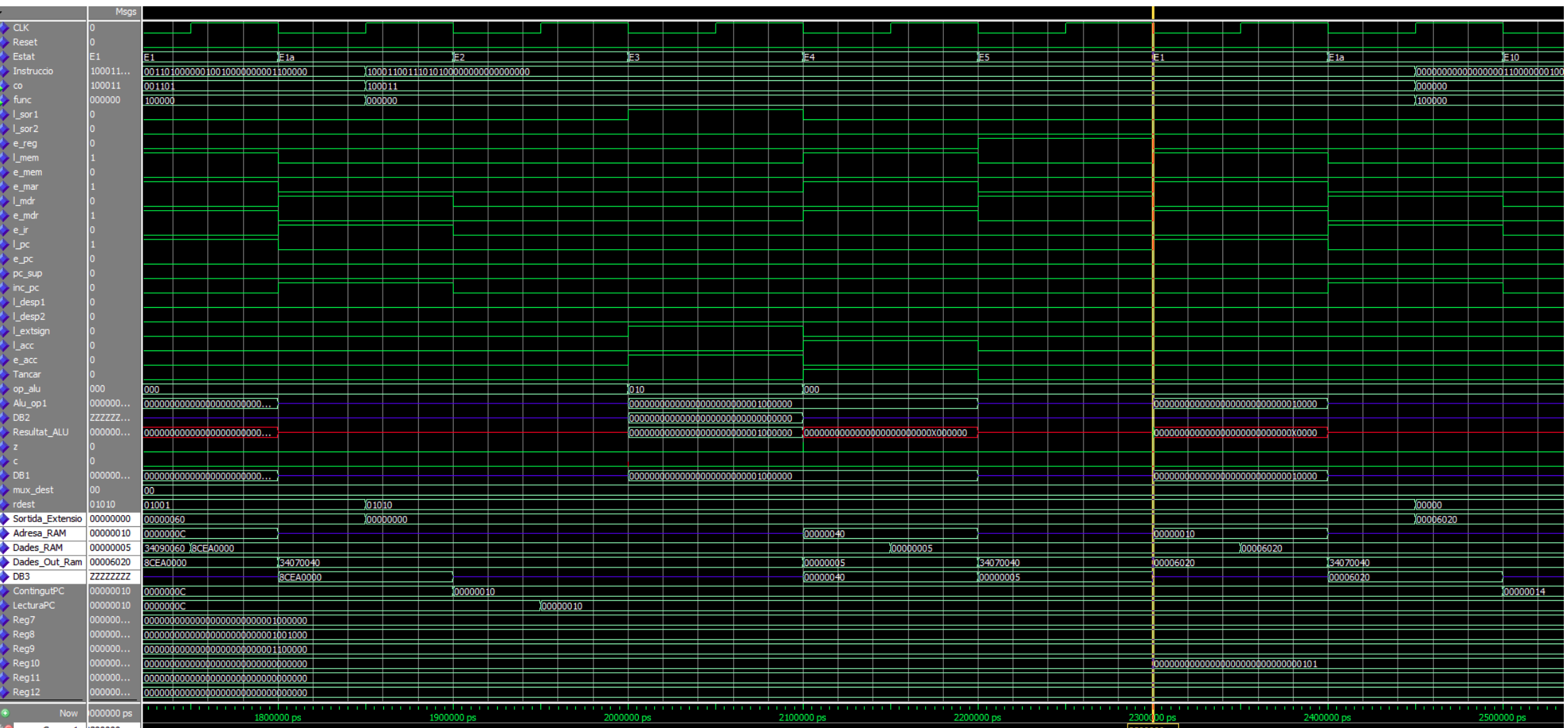
- Grafica 0x04 ori \$8, \$0, 72 ("00110100000010000000000001001000"), -- 0x04 ori \$8, \$0, 72 (0x34080048)



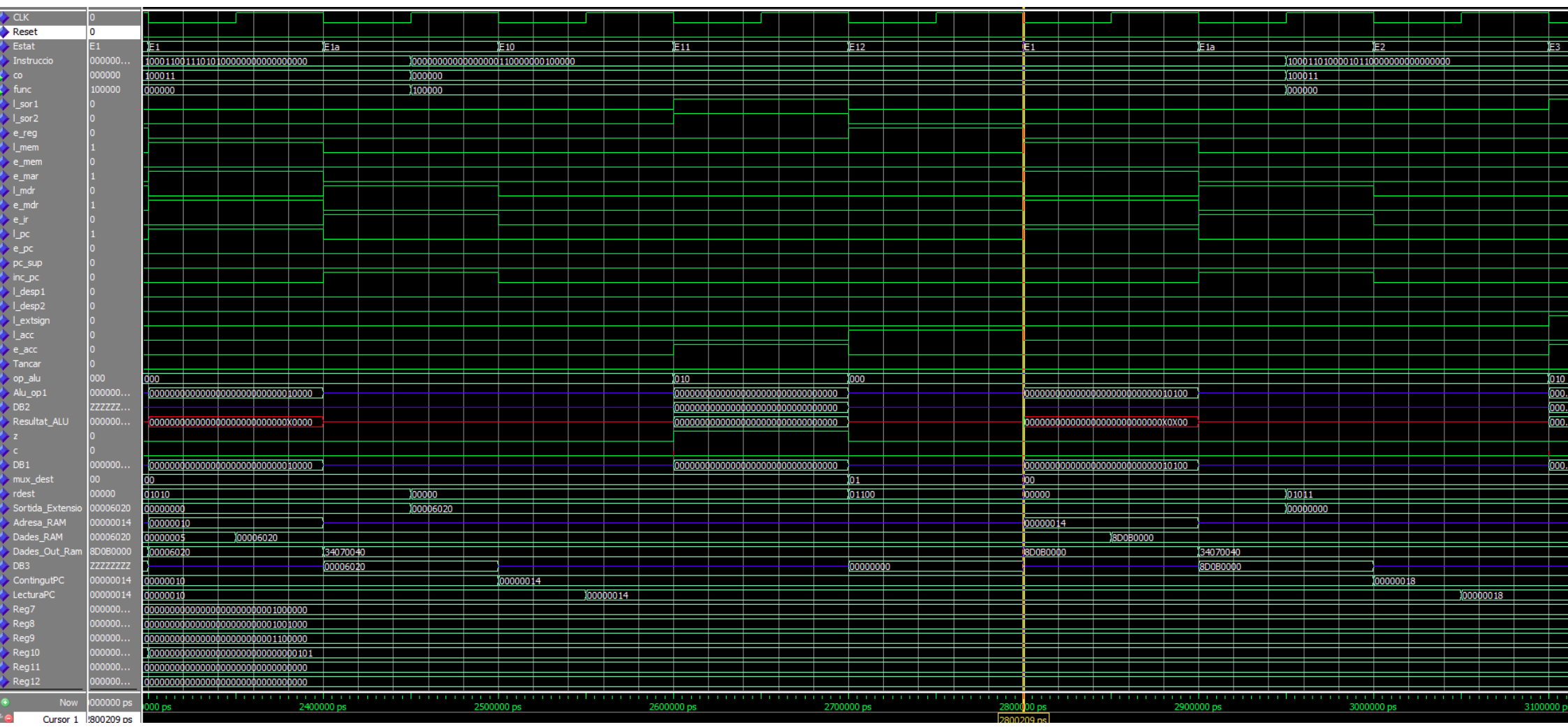
- **Grafica 0x08 ori \$9, \$0, 96** ("00110100000010010000000001100000"), -- 0x08 ori \$9, \$0, 96 (0x34090060)



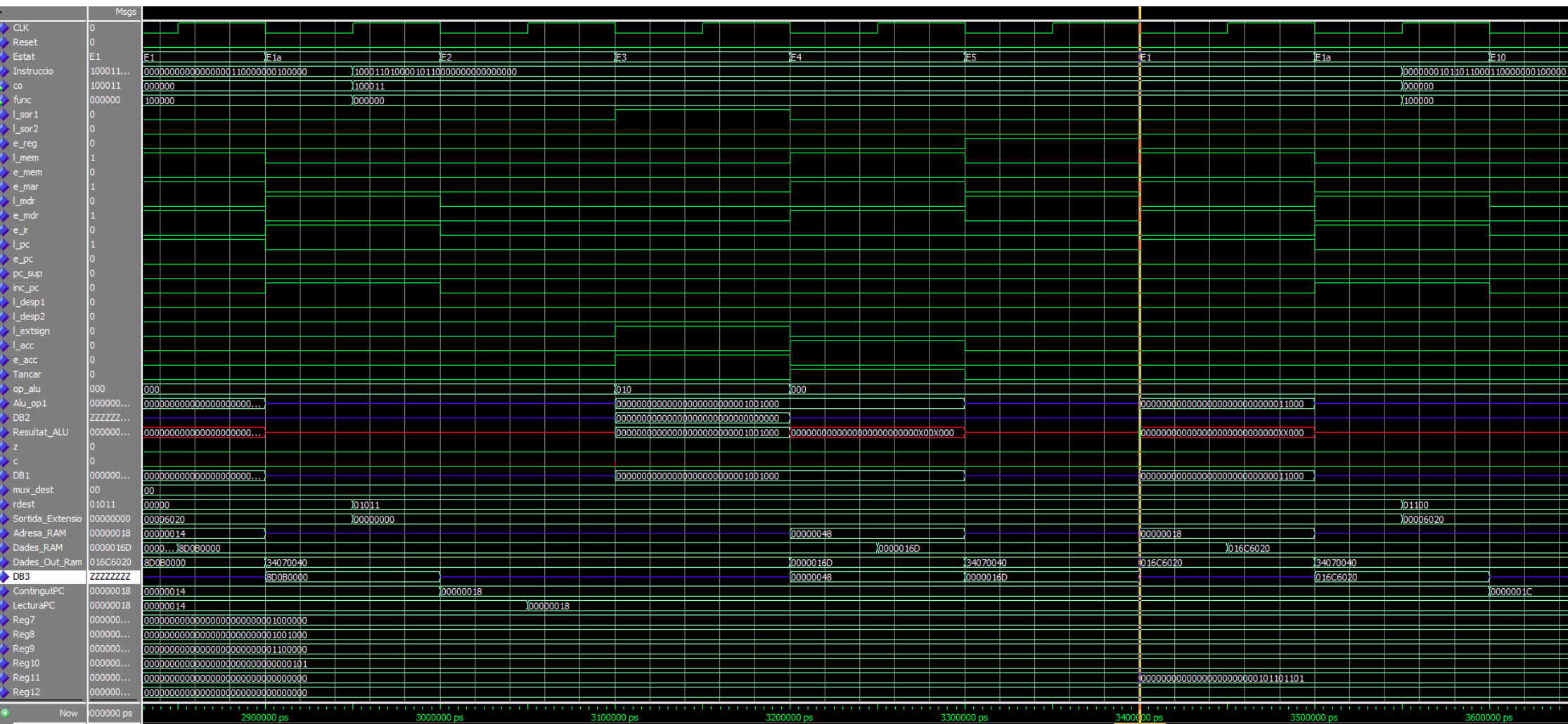
- Grafica 0x0C lw \$10, 0(\$7) ("10001100111010100000000000000000"), -- 0x0C lw \$10, 0(\$7) (0x8CEA0000)



- **Grafica 0x10 add \$12, \$0, \$0** ("0000000000000000000110000000100000"), -- 0x10 add \$12, \$0, \$0 (0x00006020)



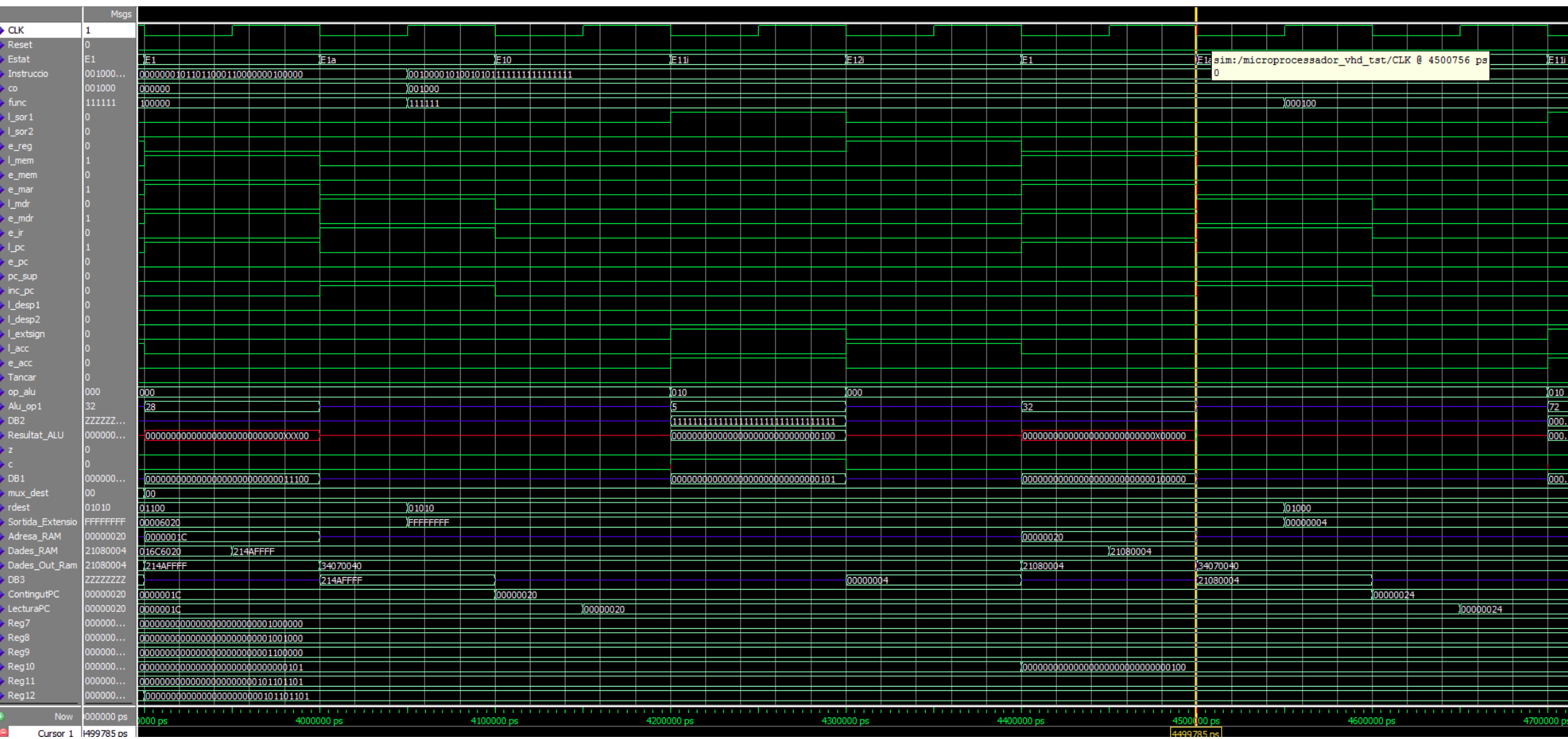
- **Grafica 0x14 lw \$11, 0(\$8)** ("10001101000010110000000000000000"), -- 0x14 lw \$11, 0(\$8) (0x8D0B0000)



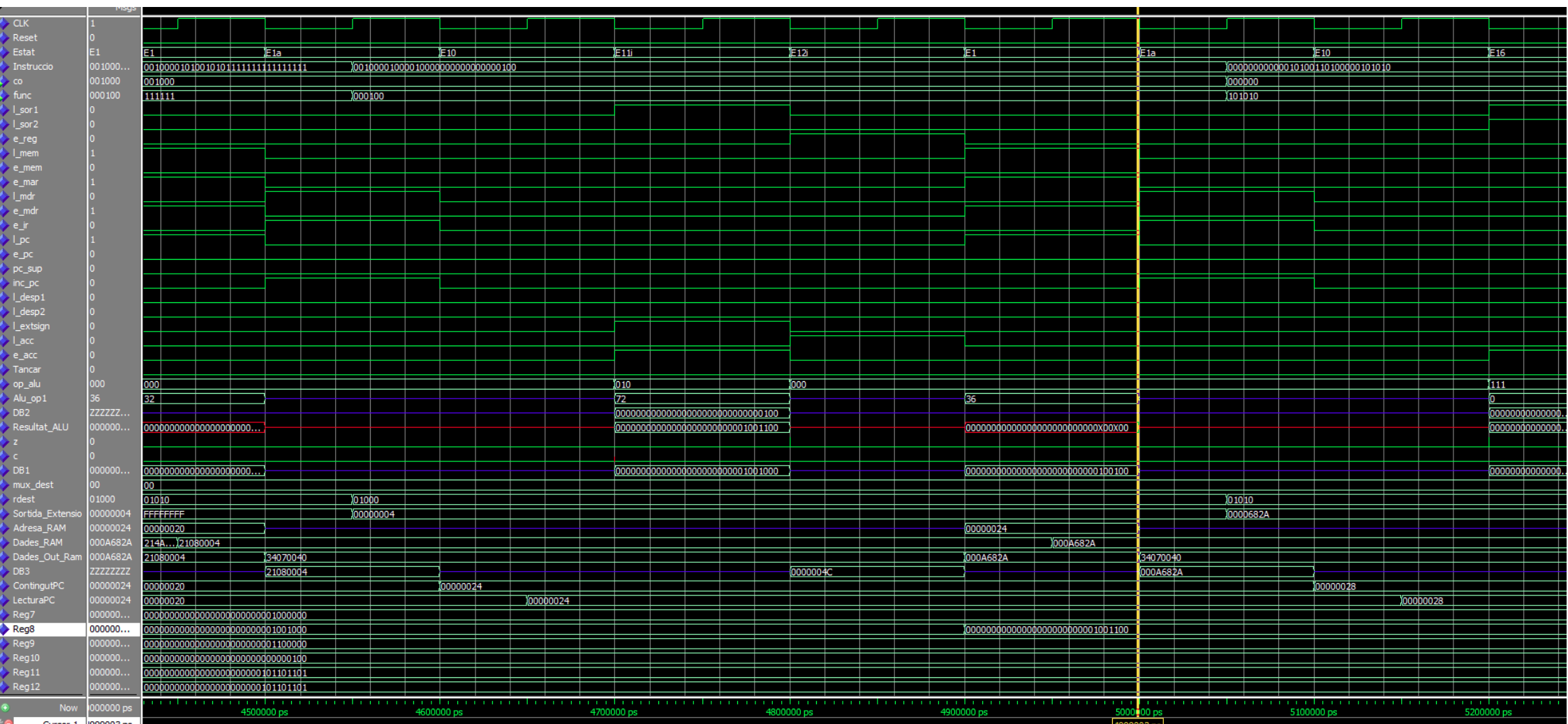
- **Grafica 0x18 add \$12, \$11, \$12** ("0000000010110110001100000000100000"), -- 0x18 add \$12, \$11, \$12 (0x016C6020)



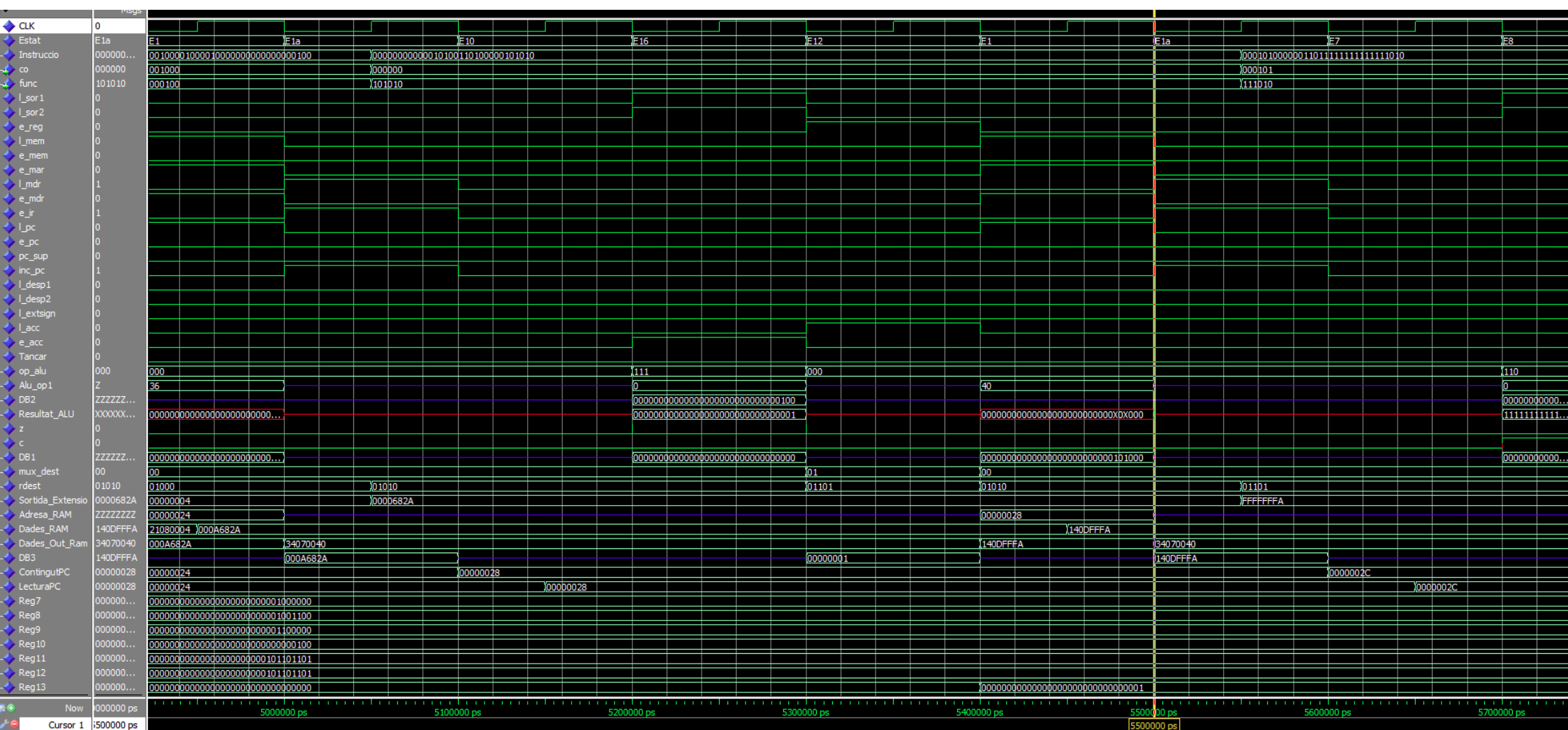
- Grafica 0x1C addi \$10, \$10, -1 ("00100001010010101111111111111111"), -- 0x1C addi \$10, \$10, -1 (0x214AFFFF)



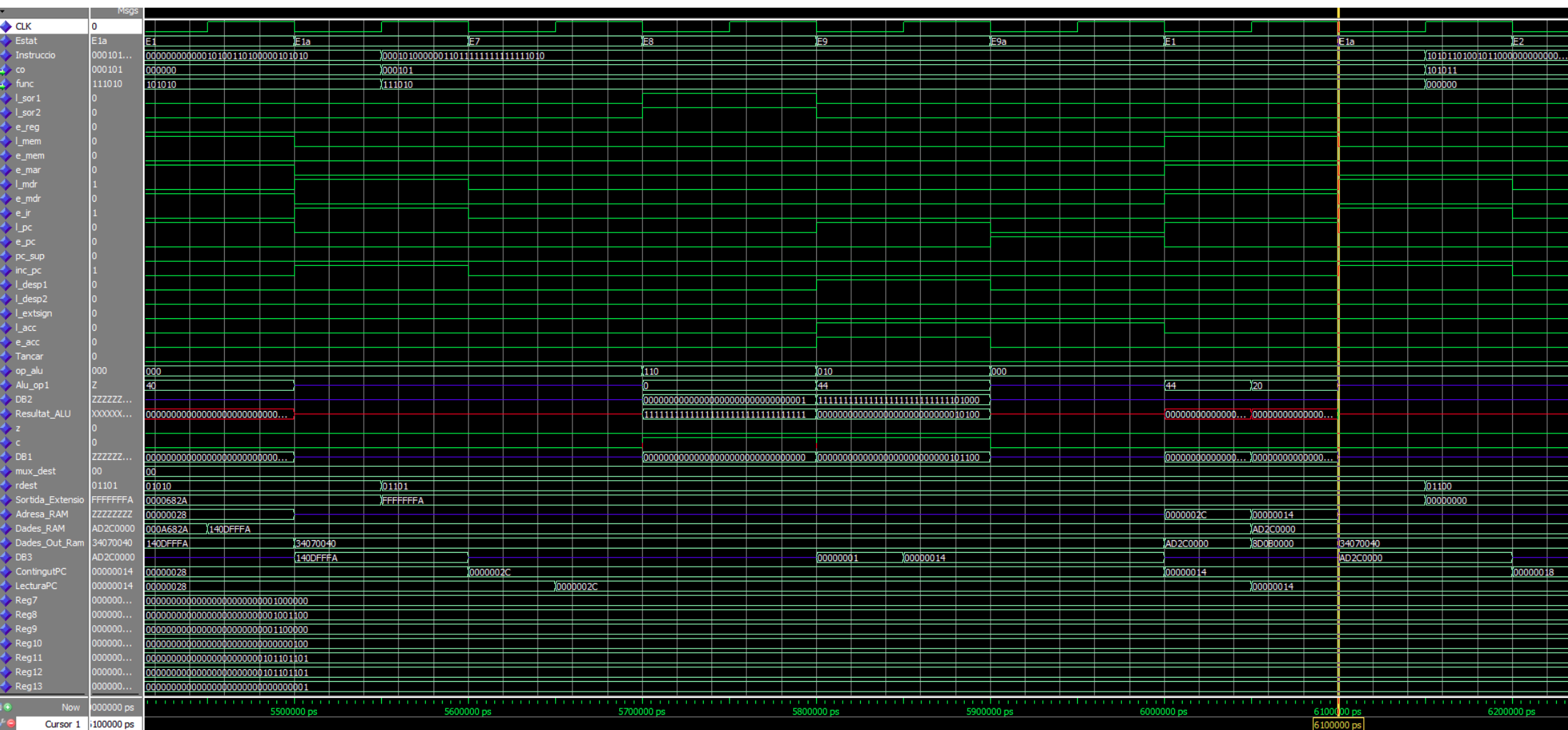
- **Grafica 0x20 addi \$8, \$8, 4** ("00100001000010000000000000000000100"), -- 0x20 addi \$8, \$8, 4 (0x21080004)



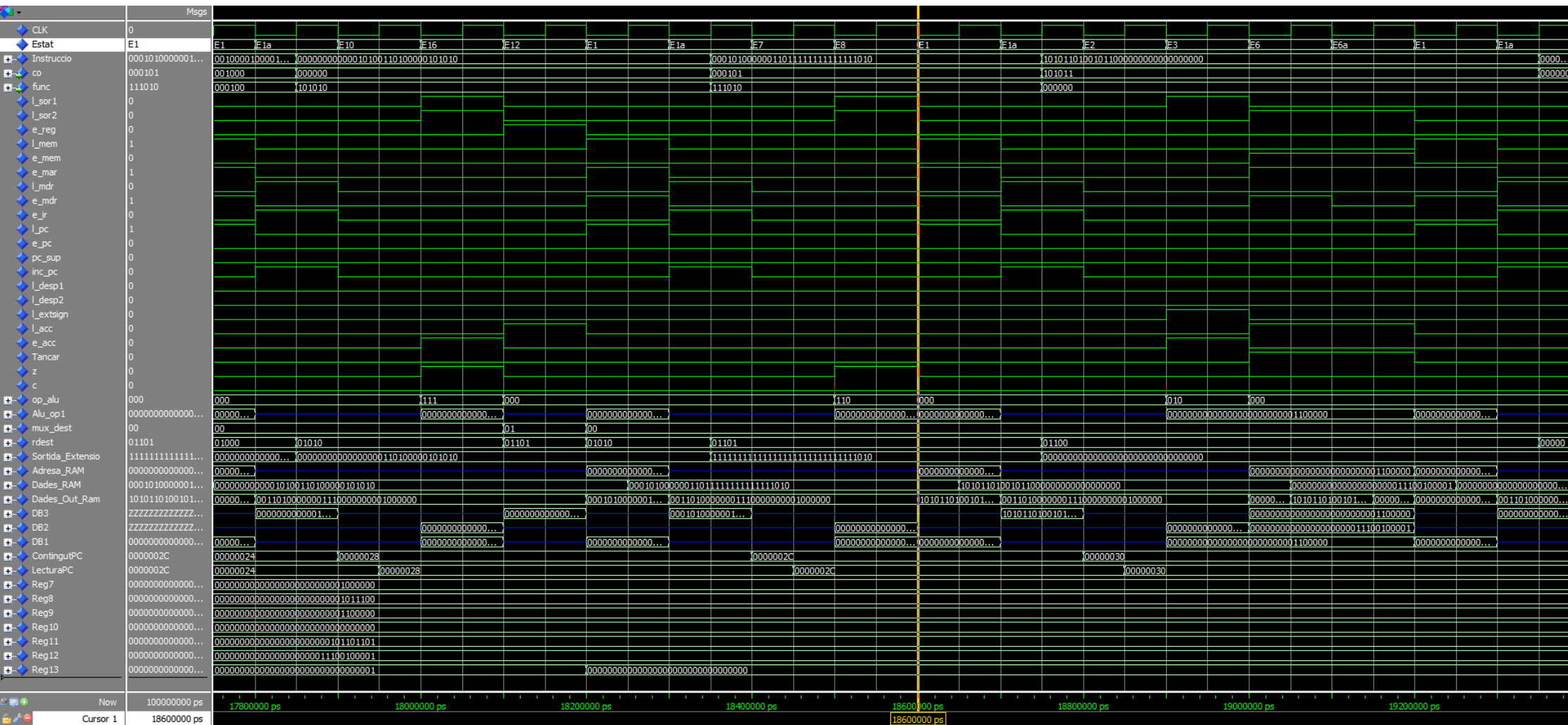
- **Grafica 0x24 slt \$13, \$0, \$10** ("00000000000010100110100000101010"), -- 0x24 slt \$13, \$0, \$10 (0x000A682A)



- **Grafica 0x28 bne \$0, \$13, bucle** ("00010100000011011111111111111010"), -- 0x28 bne \$0, \$13, bucle (0x140DFFFFB)



- Grafica 0x2C sw \$12, 0(\$9) ("10101101001011000000000000000000"), -- 0x2C sw \$12, 0(\$9) (0xAD2C0000)



TAULA CONTINGUT MC

Omplim la taula amb el contingut dels 3 Busos i dels registres utilitzats pel programa en l'última fase de l'última instrucció.

	VALOR (hex)
BUS 1	0x60
BUS 2	0x721
BUS 3	0x60
Registre 7	0x40
Registre 8	0x5C
Registre 9	0x60
Registre 10	0x00
Registre 11	0x16D
Registre 12	0x721
Registre 13	0x00

CODI ASSEMBLADOR

```
.data 0x
.word

. text 0x00000000

0x00 ori $7, $0, 64

0x04 ori $8, $0, 72

0x08 ori $9, $0, 96

0x0C lw $10, 0($7)

0x10 add $12, $0, $0

0x14 lw $11, 0($8)

0x18 add $12, $11, $12

0x1C addi $10, $10, -1

0x20 addi $8, $8, 4

0x24 slt $13, $0, $10

0x28 bne $0, $13, bucle

0x2C sw $12, 0($9)
```