4.5 Testing the new Risc V Pipelined Module

Before simulating it on our bubble sort developed in Task1, we have tested it on these instructions:

```
add x1, x2, x3
beq x5, x0, Exit
add x1, x2, x3
beq x0, x0, Finish
Exit:add x1, x1, x1
lw x2, 0(x4)
add x3, x1, x2
Finish: add x0, x0, x0
```

After coding this module, we tested it on the some test instructions which have been populated in the Instruction Memory Shown below:

```
module Instruction_Memory
(
input [63:0] Inst_Address,
output reg [31:0] Instruction
);
reg[7:0] inst_mem [31:0];
initial
begin
// 0x00000033   add x0 x0 x0    Finish: add x0, x0, x0
// 00000000 00000000 000000000 00110011
inst_mem[31] = 8'b00000000;
inst_mem[30] = 8'b00000000;
inst_mem[29] = 8'b00000000;
inst_mem[28] = 8'b00110011;

// 0x002081b3   add x3 x1 x2    add x3, x1, x2
// 00000000 00100000 10000001 10110011
inst_mem[27] = 8'b00000000;
inst_mem[26] = 8'b00100000;
inst_mem[25] = 8'b10000001;
inst_mem[24] = 8'b10110011;

// 0x00022103   lw x2 0(x4)     lw x2, 0(x4)
// 000000 00000010 00100001 00000011
// changing to ld
// 000000 00000010 0 010 00010 0000011
// 000000 00000010 0 011 00010 0000011
// 000000 00000010 00110001 00000011
inst_mem[23] = 8'b00000000;
inst_mem[22] = 8'b00000010;
inst_mem[21] = 8'b00100001;
inst_mem[20] = 8'b00000011;
```

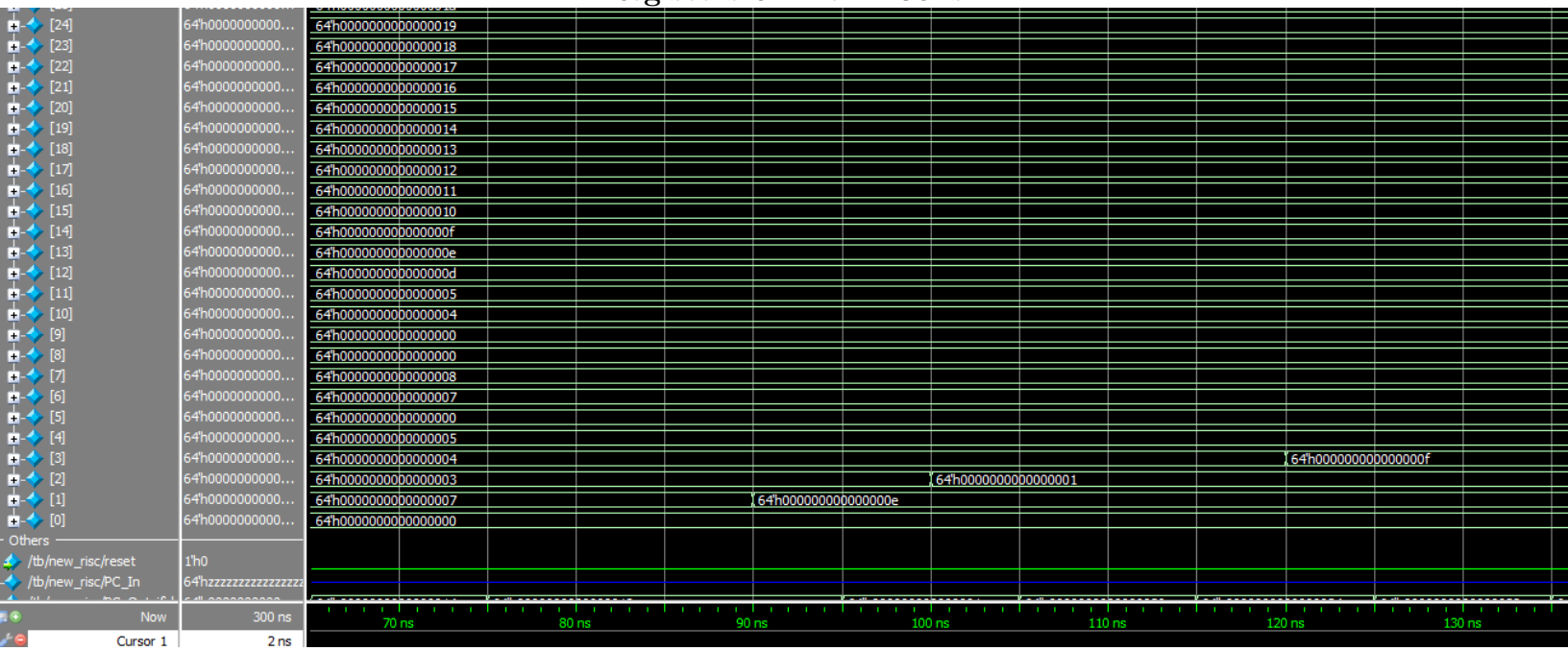
```
33
34 // 0x001080b3   add x1 x1 x1    Exit:add x1, x1, x1
35 // 00000000 00010000 10000000 10110011
36 inst_mem[19] = 8'b00000000;
37 inst_mem[18] = 8'b00010000;
38 inst_mem[17] = 8'b10000000;
39 inst_mem[16] = 8'b10110011;
40
41 // 0x00000863   beq x0 x0 16    beq x0, x0, Finish
42 // 00000000 00001000 01100011
43 inst_mem[15] = 8'b00000000;
44 inst_mem[14] = 8'b00000000;
45 inst_mem[13] = 8'b00001000;
46 inst_mem[12] = 8'b01100011;
47
48 // 0x003100b3   add x1 x2 x3    add x1, x2, x3
49 // 00000000 00110001 00000000 10110011
50 inst_mem[11] = 8'b00000000;
51 inst_mem[10] = 8'b00110001;
52 inst_mem[9] = 8'b00000000;
53 inst_mem[8] = 8'b10110011;
54
55 // 0x00028663   beq x5 x0 12    beq x5, x0, Exit
56 // 00000000 00000010 10000110 01100011
57 inst_mem[7] = 8'b00000000;
58 inst_mem[6] = 8'b00000010;
59 inst_mem[5] = 8'b10000110;
60 inst_mem[4] = 8'b01100011;
61
62 // 0x003100b3   add x1 x2 x3    add x1, x2, x3
63 // 00000000 00110001 00000000 10110011
64 inst_mem[3] = 8'b00000000;
65 inst_mem[2] = 8'b00110001;
66 inst_mem[1] = 8'b00000000;
67 inst_mem[0] = 8'b10110011;
68
69 end
70 always @(Inst_Address)
71 begin
72 assign Instruction = {inst_mem[Inst_Address+3], inst_mem[Inst_Address
      +2],inst_mem[Inst_Address+1],inst_mem[Inst_Address]};
73 end
74 endmodule
```

Listing 21: Instruction memory to check for stall and flush

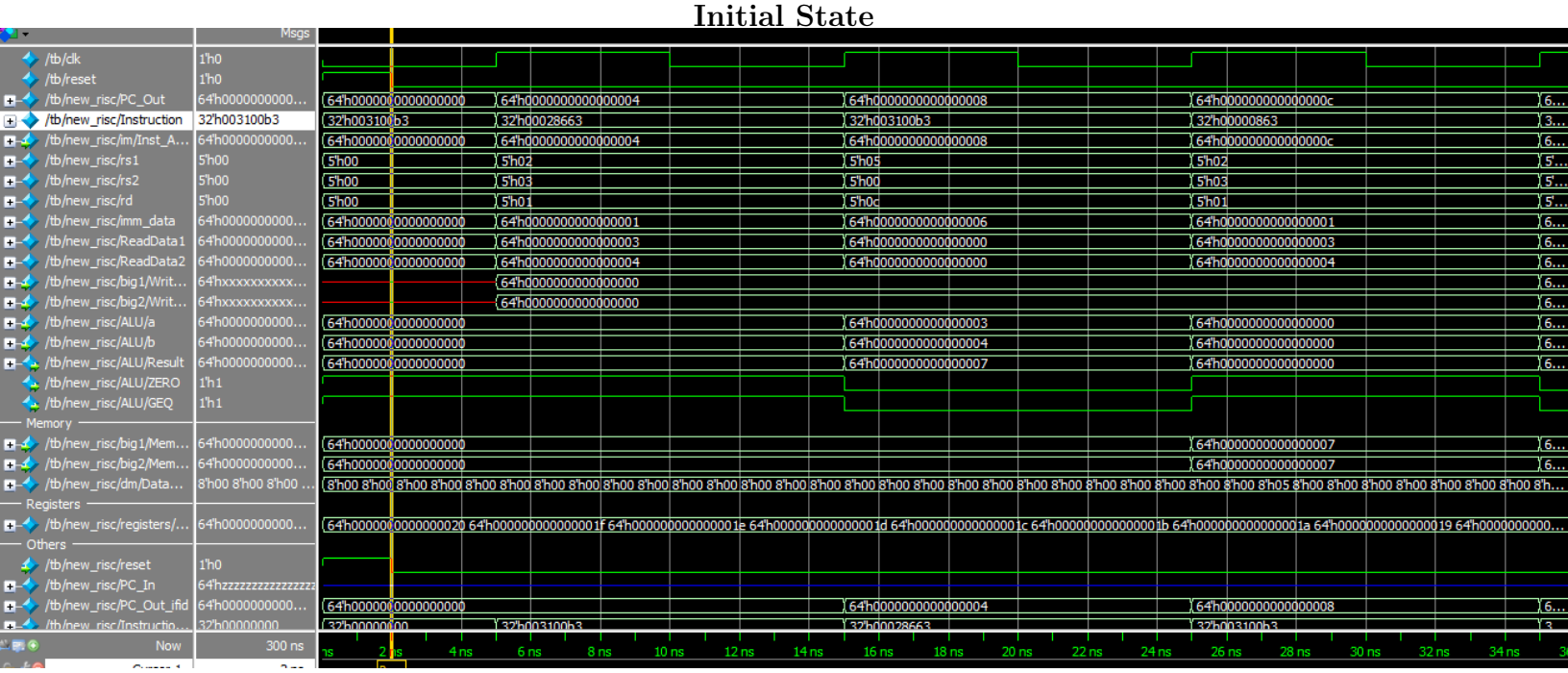
4.5.0.1 RESULTS

We have tested these instructions for 130ns.

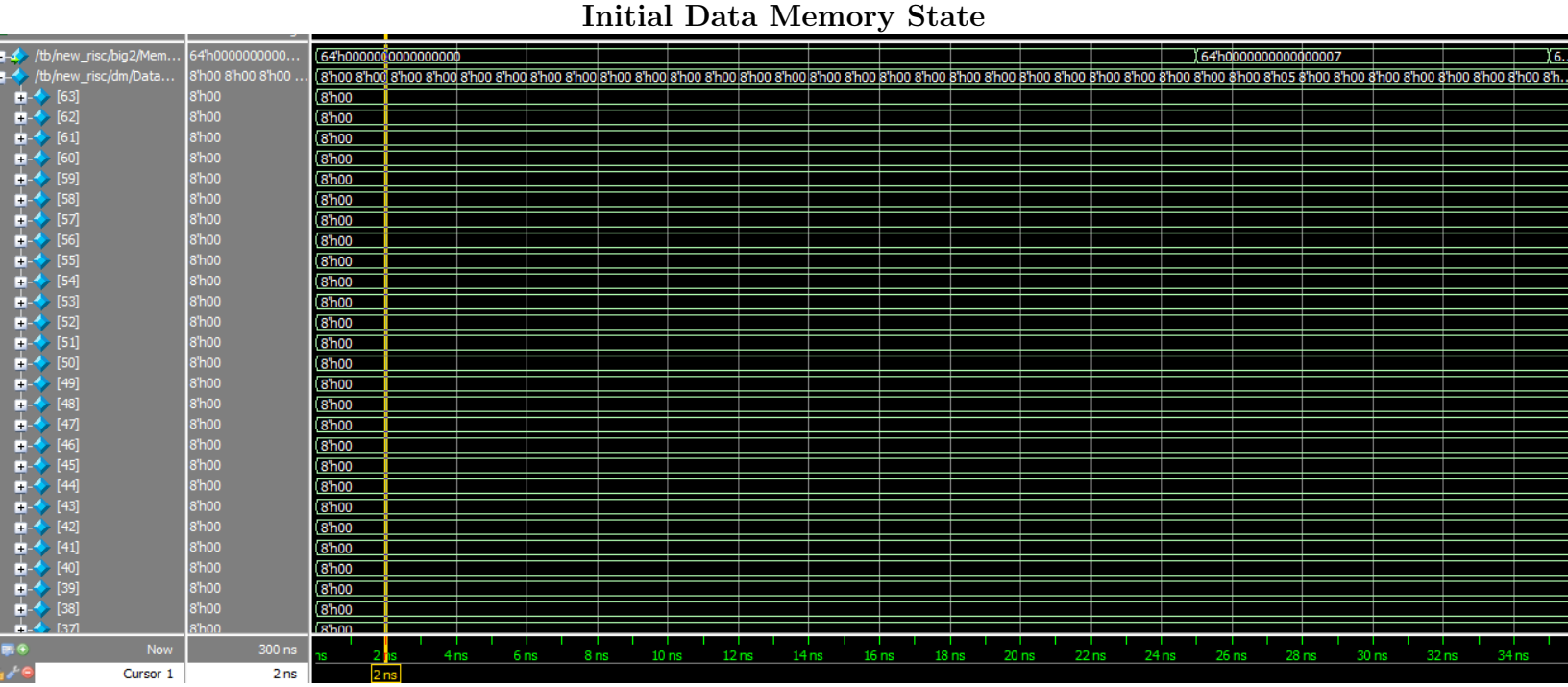
Registers 0-24 till 130ns

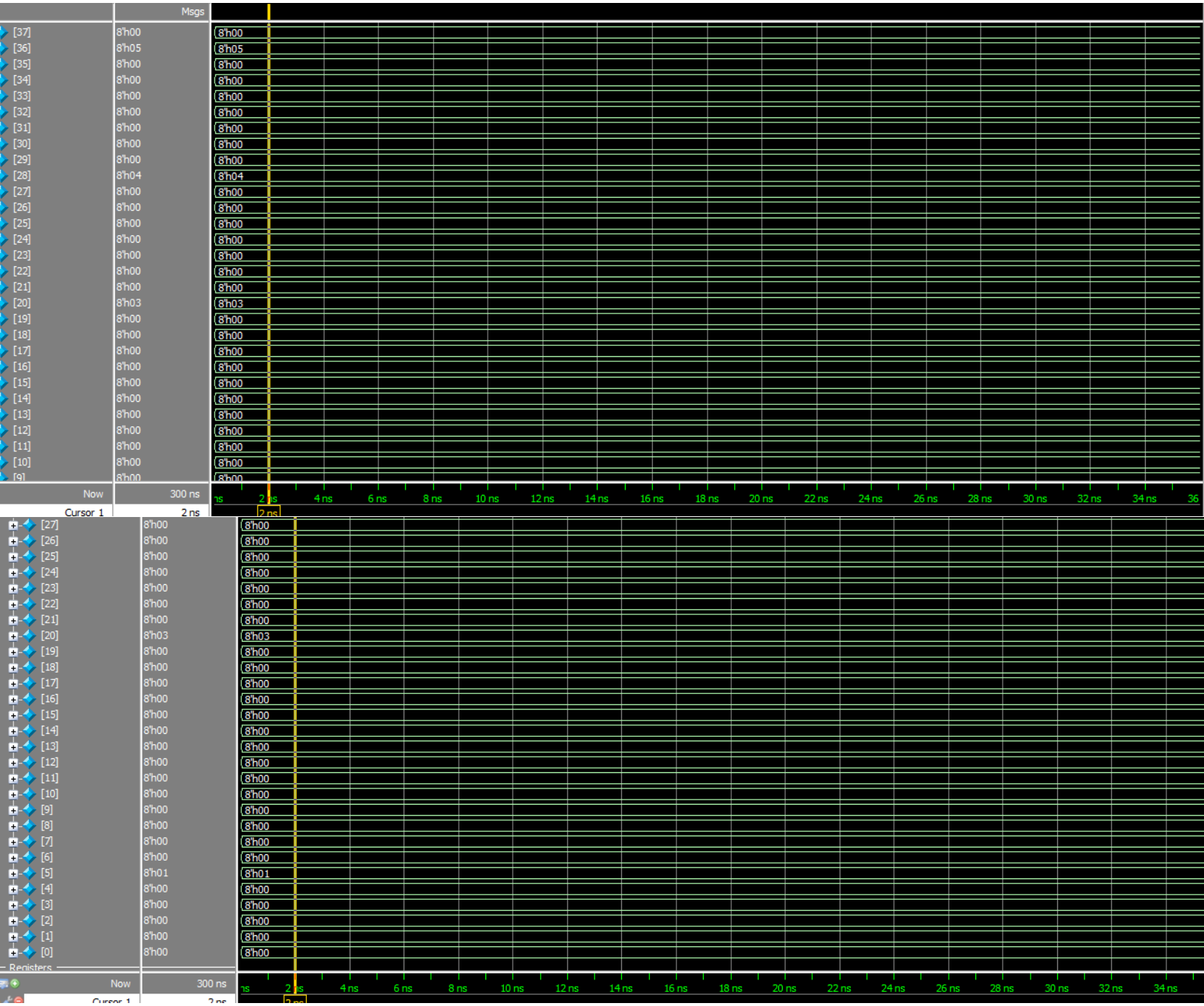


Initial State

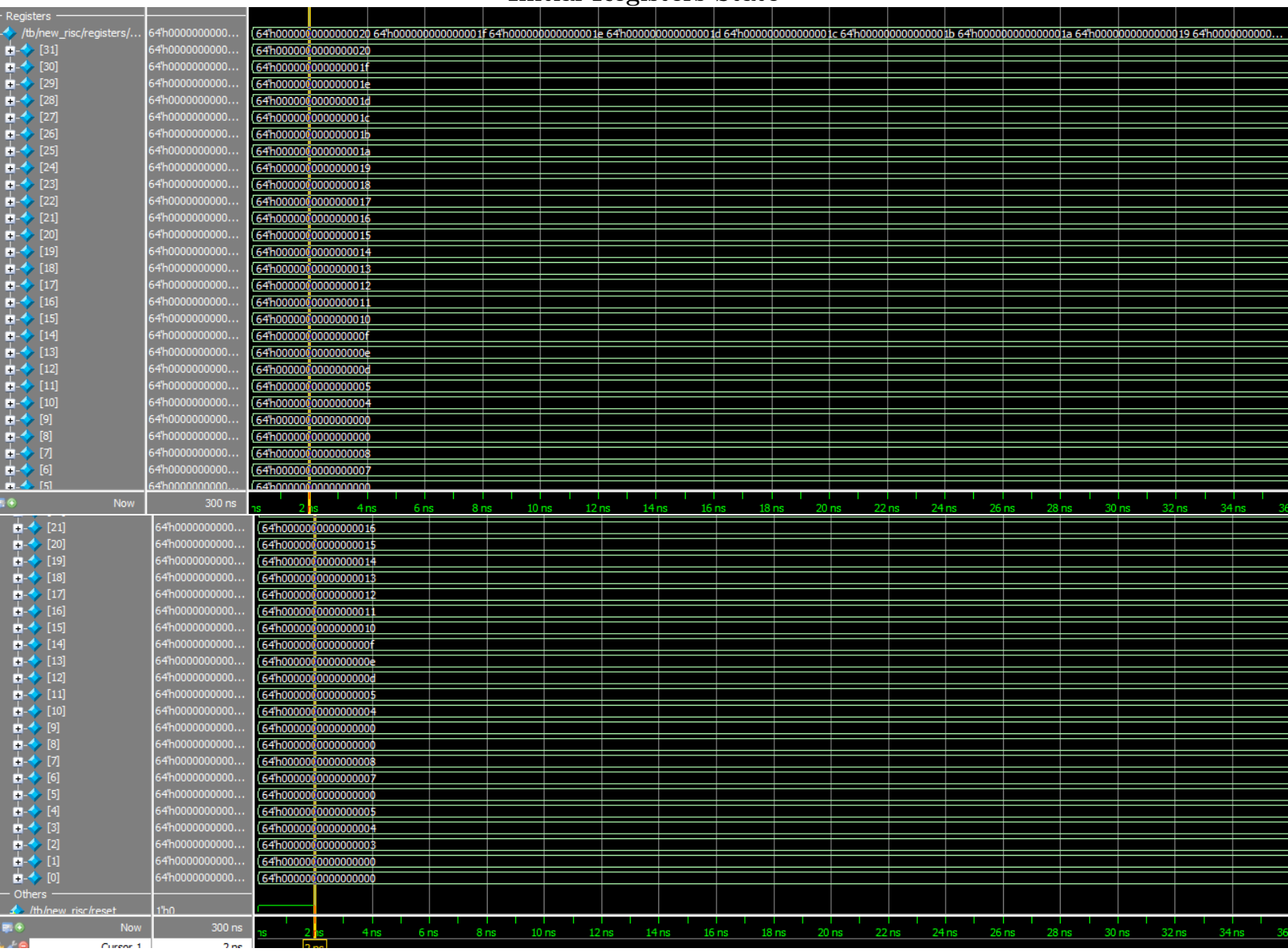


Initial Data Memory State

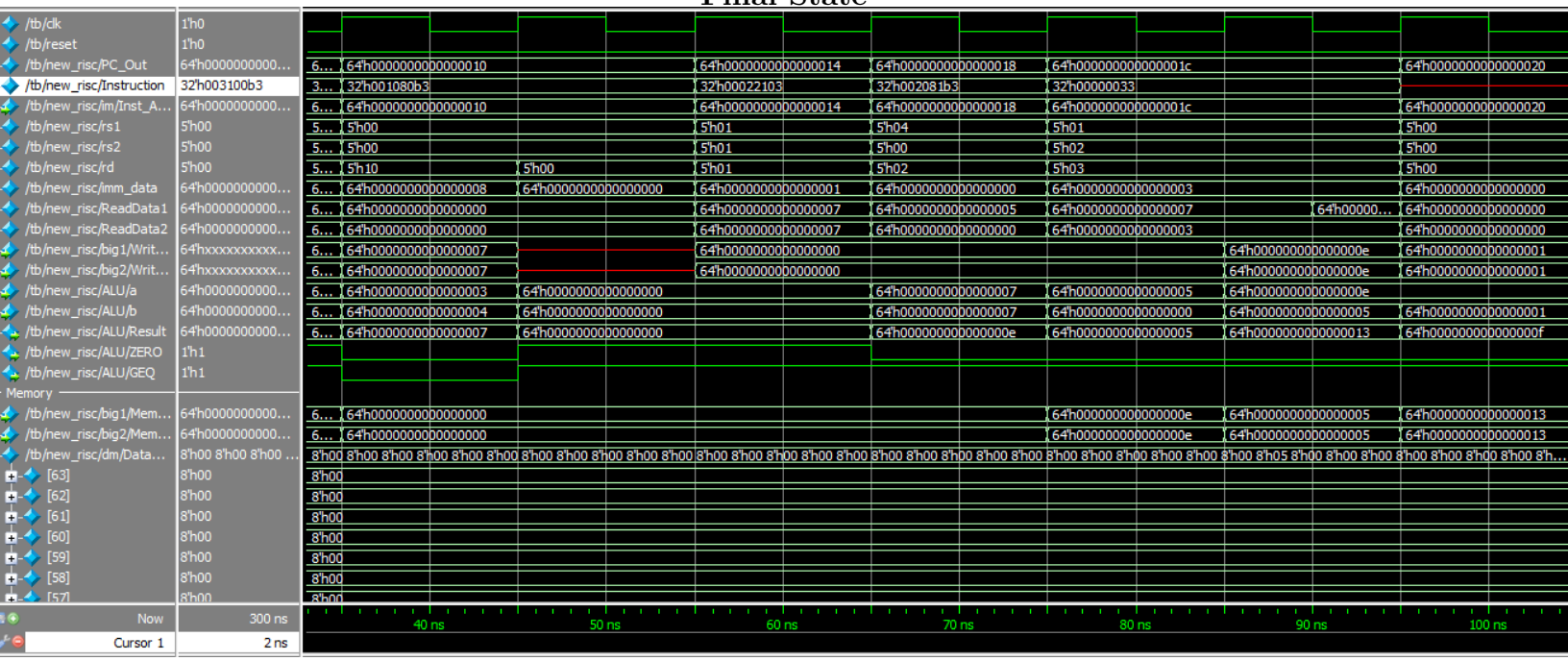



Msgs
[37] 8'h00
[36] 8'h05
[35] 8'h00
[34] 8'h00
[33] 8'h00
[32] 8'h00
[31] 8'h00
[30] 8'h00
[29] 8'h00
[28] 8'h04
[27] 8'h00
[26] 8'h00
[25] 8'h00
[24] 8'h00
[23] 8'h00
[22] 8'h00
[21] 8'h00
[20] 8'h03
[19] 8'h00
[18] 8'h00
[17] 8'h00
[16] 8'h00
[15] 8'h00
[14] 8'h00
[13] 8'h00
[12] 8'h00
[11] 8'h00
[10] 8'h00
Now 300 ns
Cursor 1 2 ns
2 ns 4 ns 6 ns 8 ns 10 ns 12 ns 14 ns 16 ns 18 ns 20 ns 22 ns 24 ns 26 ns 28 ns 30 ns 32 ns 34 ns
[27] 8'h00
[26] 8'h00
[25] 8'h00
[24] 8'h00
[23] 8'h00
[22] 8'h00
[21] 8'h00
[20] 8'h03
[19] 8'h00
[18] 8'h00
[17] 8'h00
[16] 8'h00
[15] 8'h00
[14] 8'h00
[13] 8'h00
[12] 8'h00
[11] 8'h00
[10] 8'h00
[9] 8'h00
[8] 8'h00
[7] 8'h00
[6] 8'h00
[5] 8'h01
[4] 8'h00
[3] 8'h00
[2] 8'h00
[1] 8'h00
[0] 8'h00
Registers
Now 300 ns
2 ns 4 ns 6 ns 8 ns 10 ns 12 ns 14 ns 16 ns 18 ns 20 ns 22 ns 24 ns 26 ns 28 ns 30 ns 32 ns 34 ns

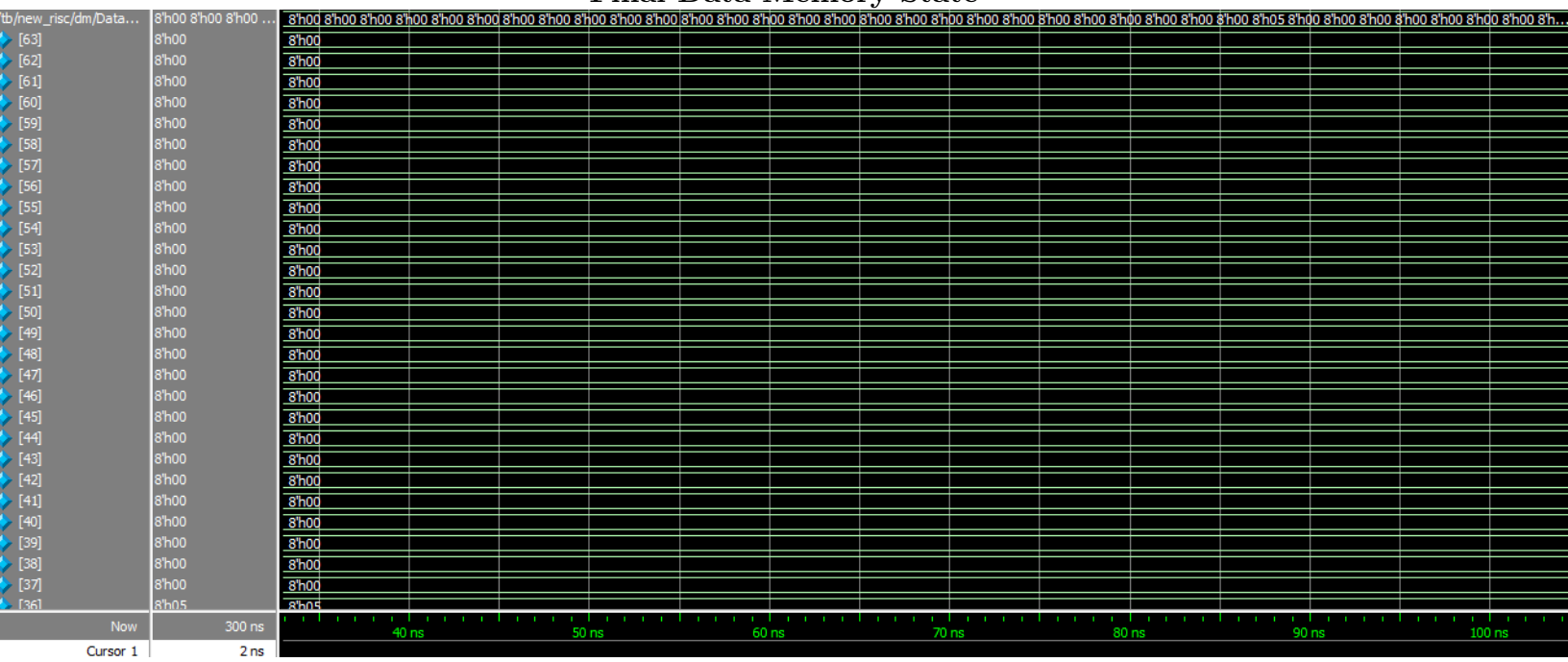
Initial Registers State

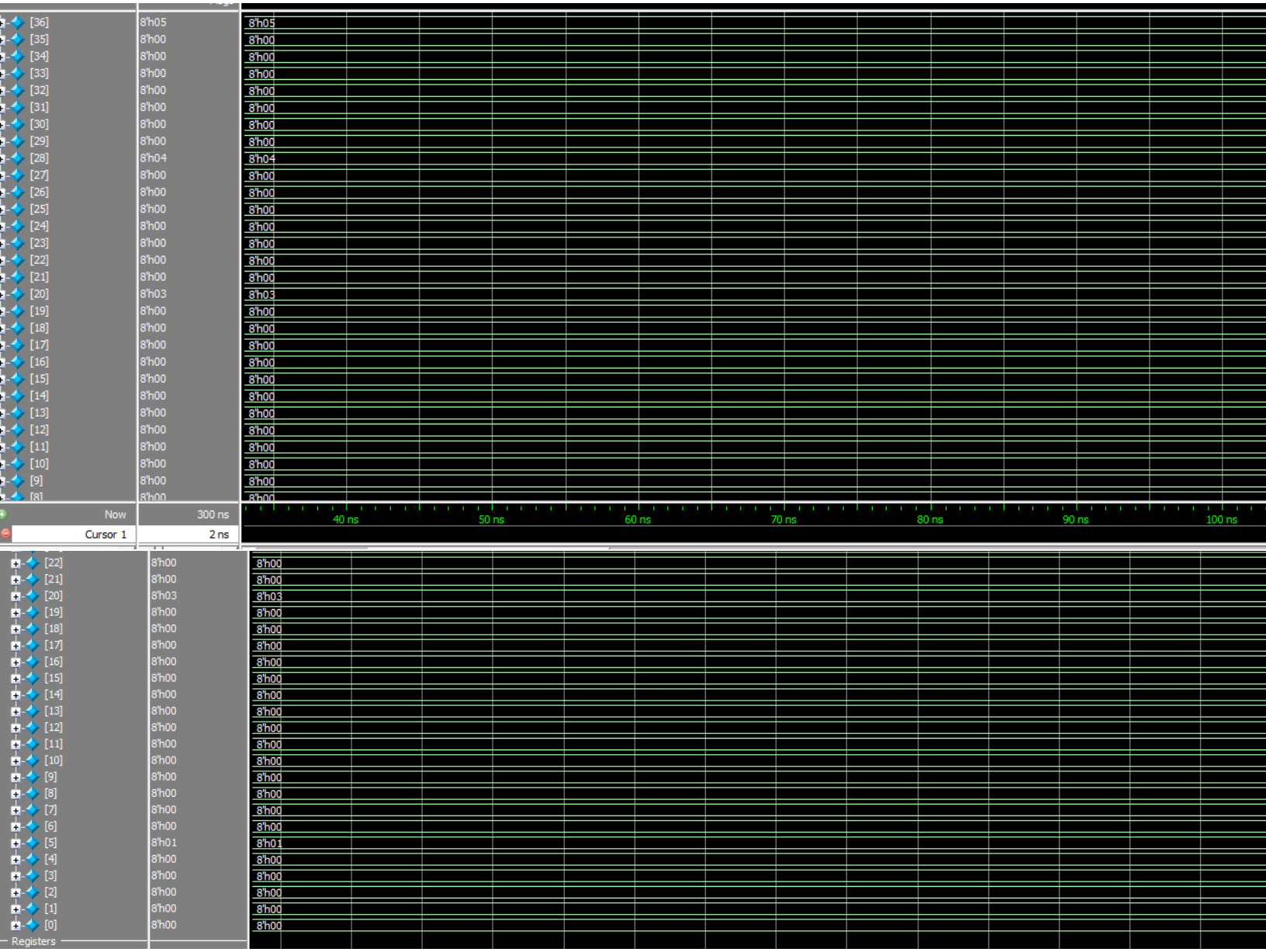


Final State

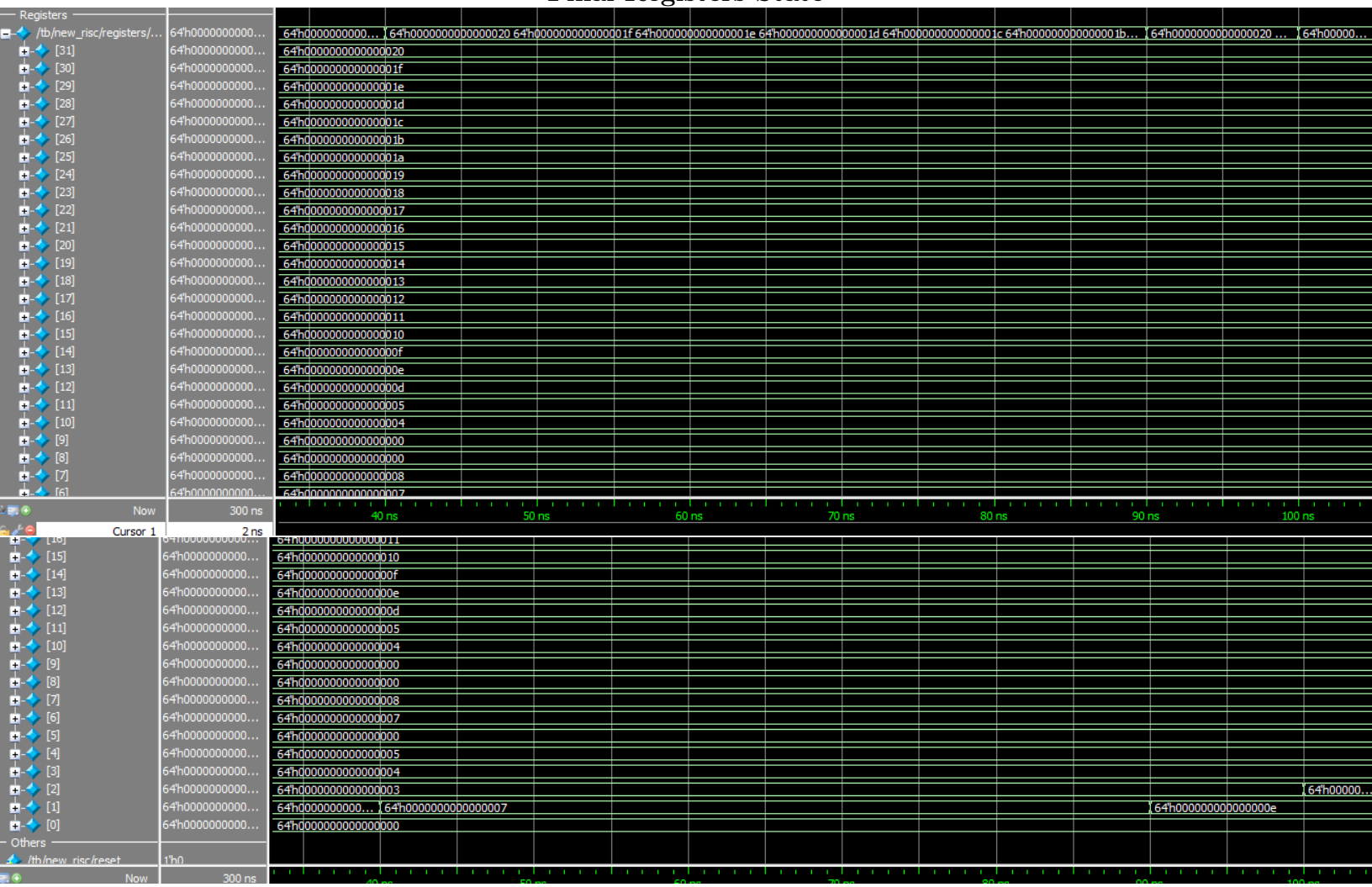


Final Data Memory State

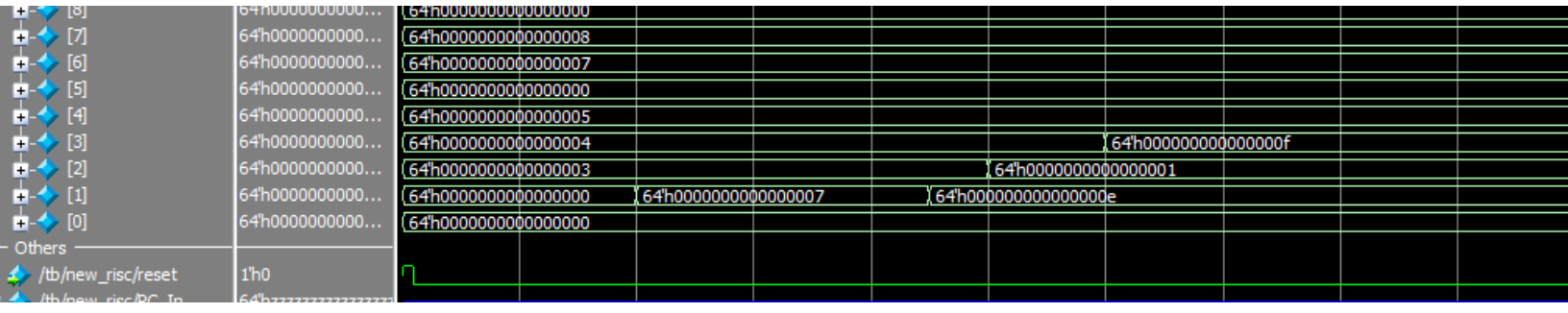



[36] 8'h05
[35] 8'h00
[34] 8'h00
[33] 8'h00
[32] 8'h00
[31] 8'h00
[30] 8'h00
[29] 8'h00
[28] 8'h04
[27] 8'h00
[26] 8'h00
[25] 8'h00
[24] 8'h00
[23] 8'h00
[22] 8'h00
[21] 8'h00
[20] 8'h03
[19] 8'h00
[18] 8'h00
[17] 8'h00
[16] 8'h00
[15] 8'h00
[14] 8'h00
[13] 8'h00
[12] 8'h00
[11] 8'h00
[10] 8'h00
[9] 8'h00
[8] 8'h00
Now 300 ns
Cursor 1 2 ns
40 ns
50 ns
60 ns
70 ns
80 ns
90 ns
100 ns
[22] 8'h00
[21] 8'h00
[20] 8'h03
[19] 8'h00
[18] 8'h00
[17] 8'h00
[16] 8'h00
[15] 8'h00
[14] 8'h00
[13] 8'h00
[12] 8'h00
[11] 8'h00
[10] 8'h00
[9] 8'h00
[8] 8'h00
[7] 8'h00
[6] 8'h00
[5] 8'h01
[4] 8'h00
[3] 8'h00
[2] 8'h00
[1] 8'h00
[0] 8'h00
Registers

Final Registers State

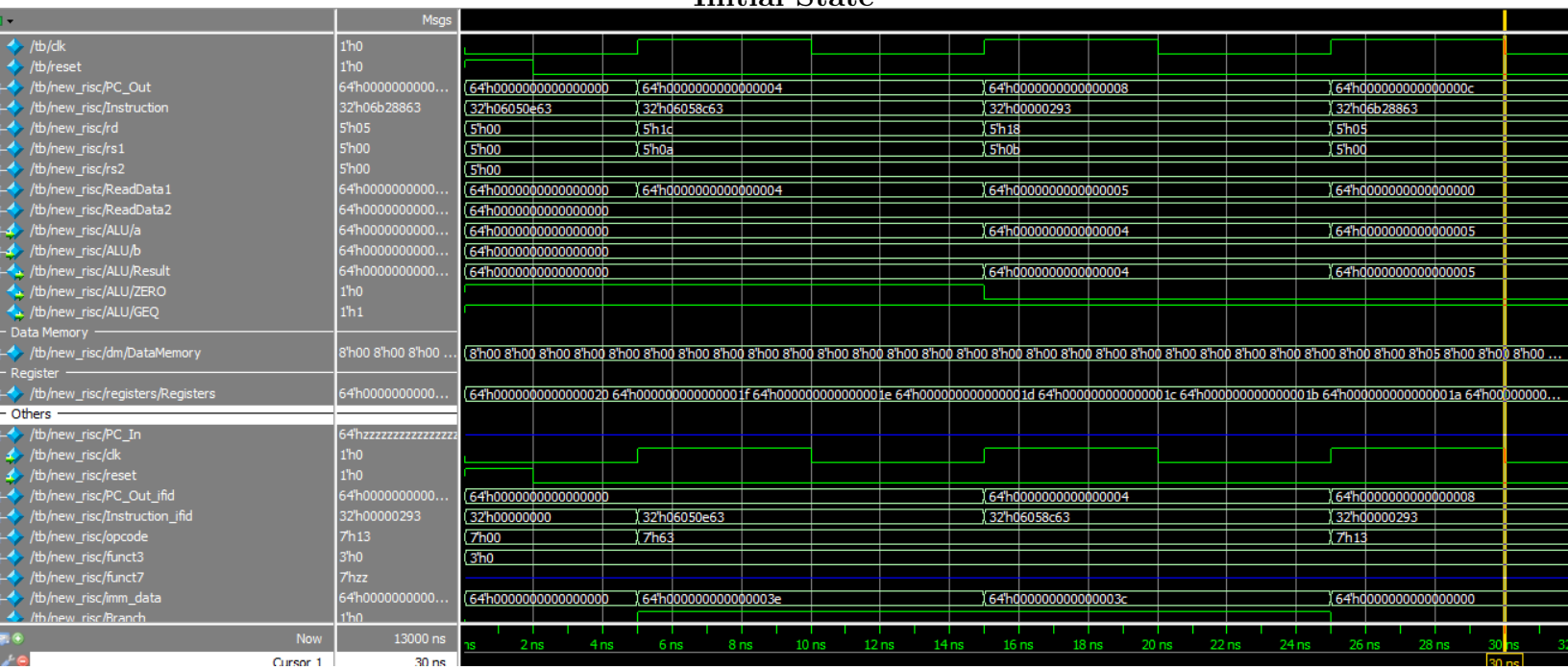


Major changes to be observed in the Resulting simulating wave is as follows:

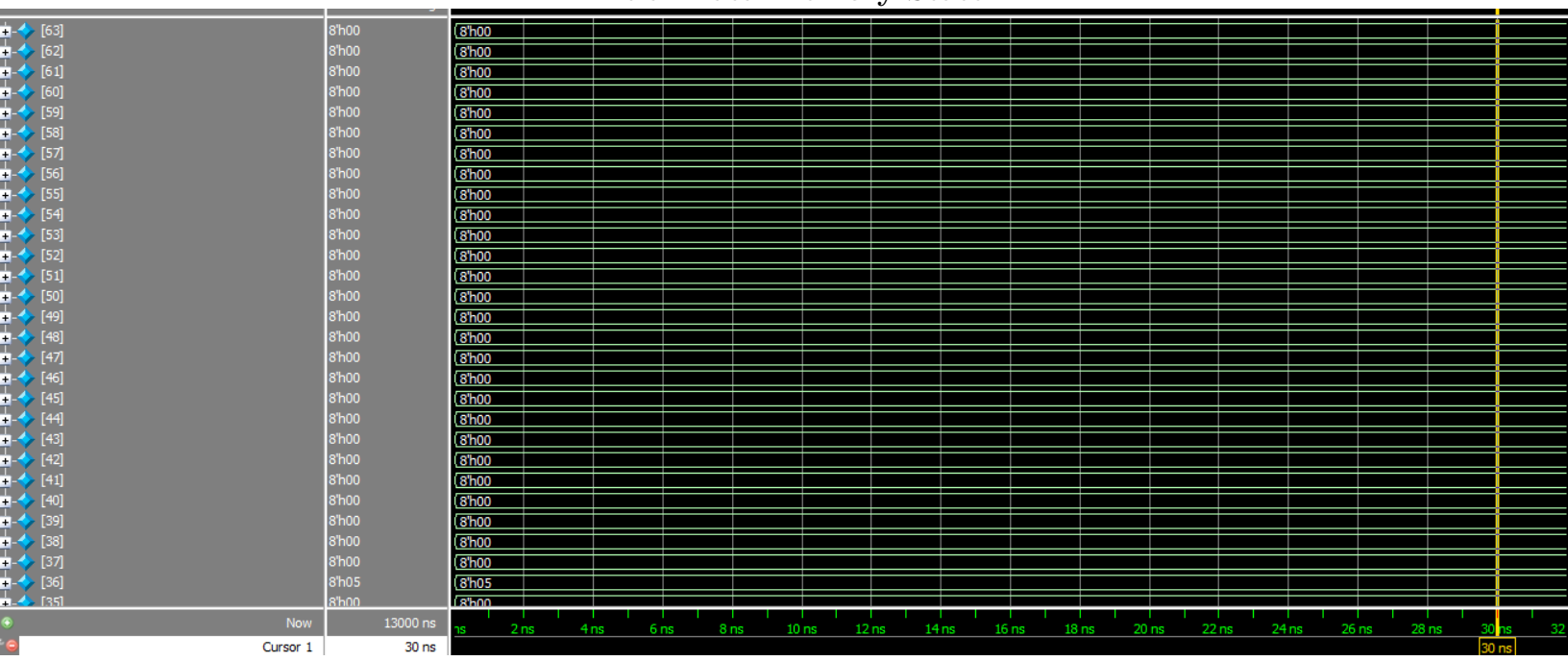


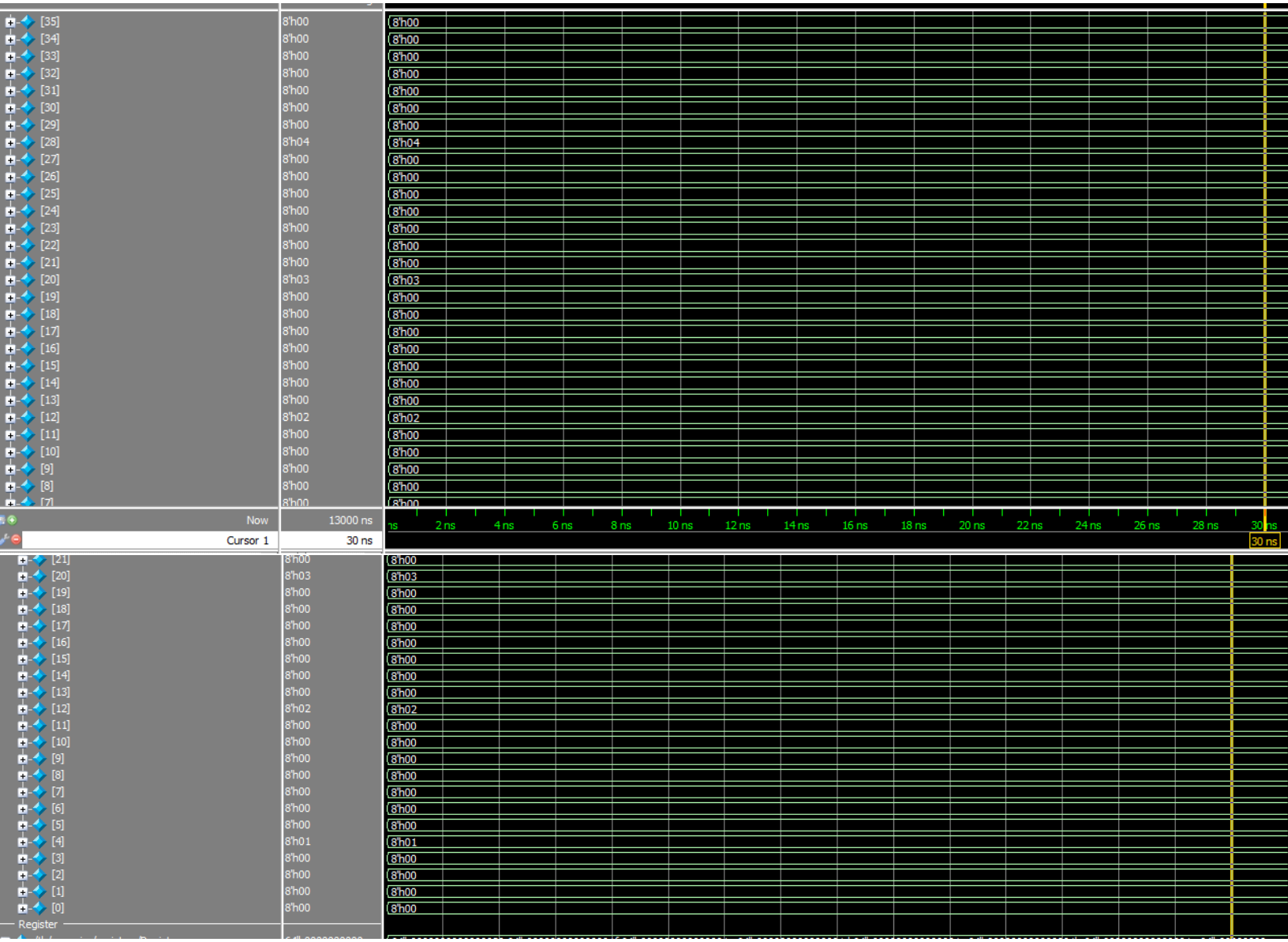
4.6 Simulation and Testing on Bubble Sort

Initial State

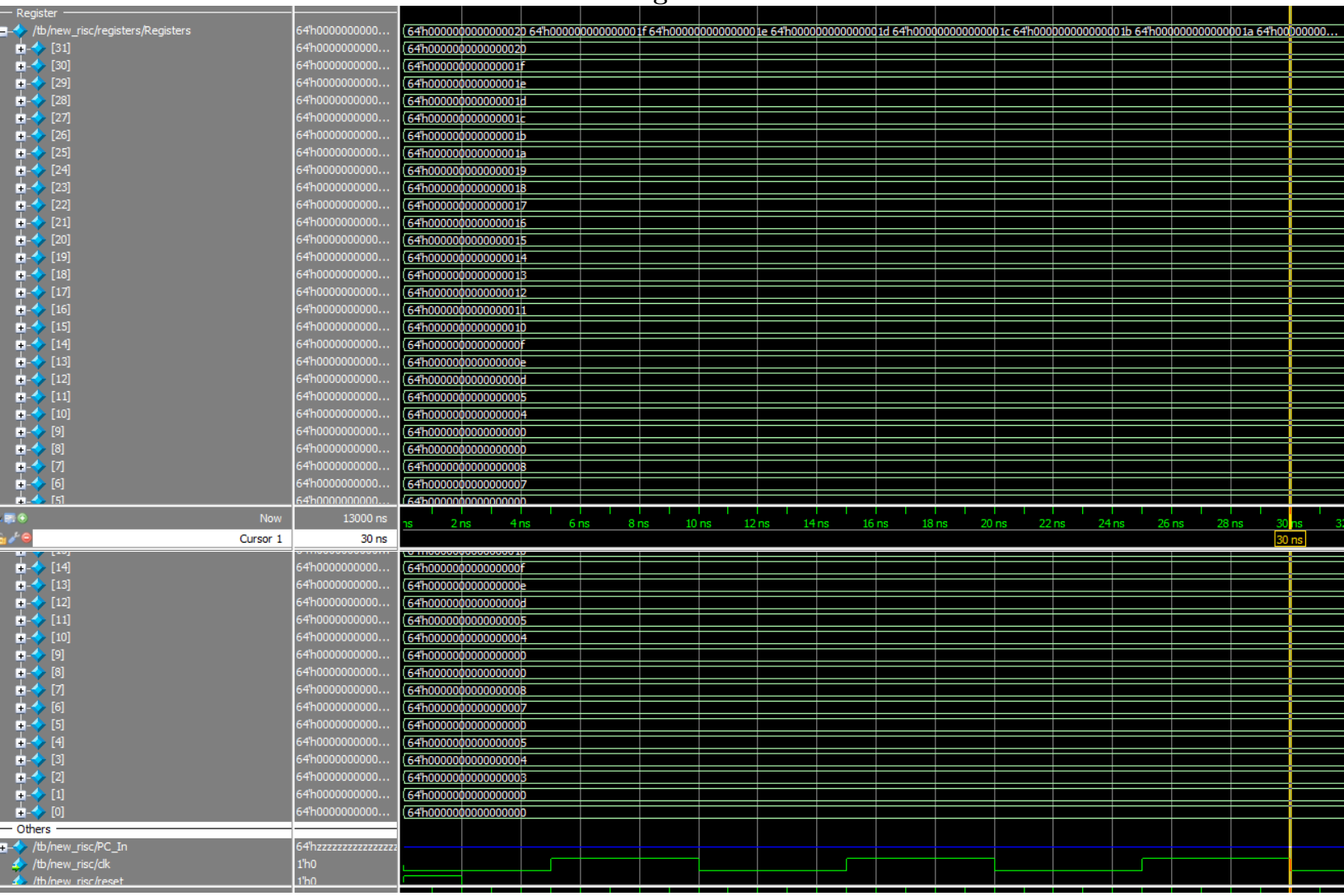


Initial Data Memory State

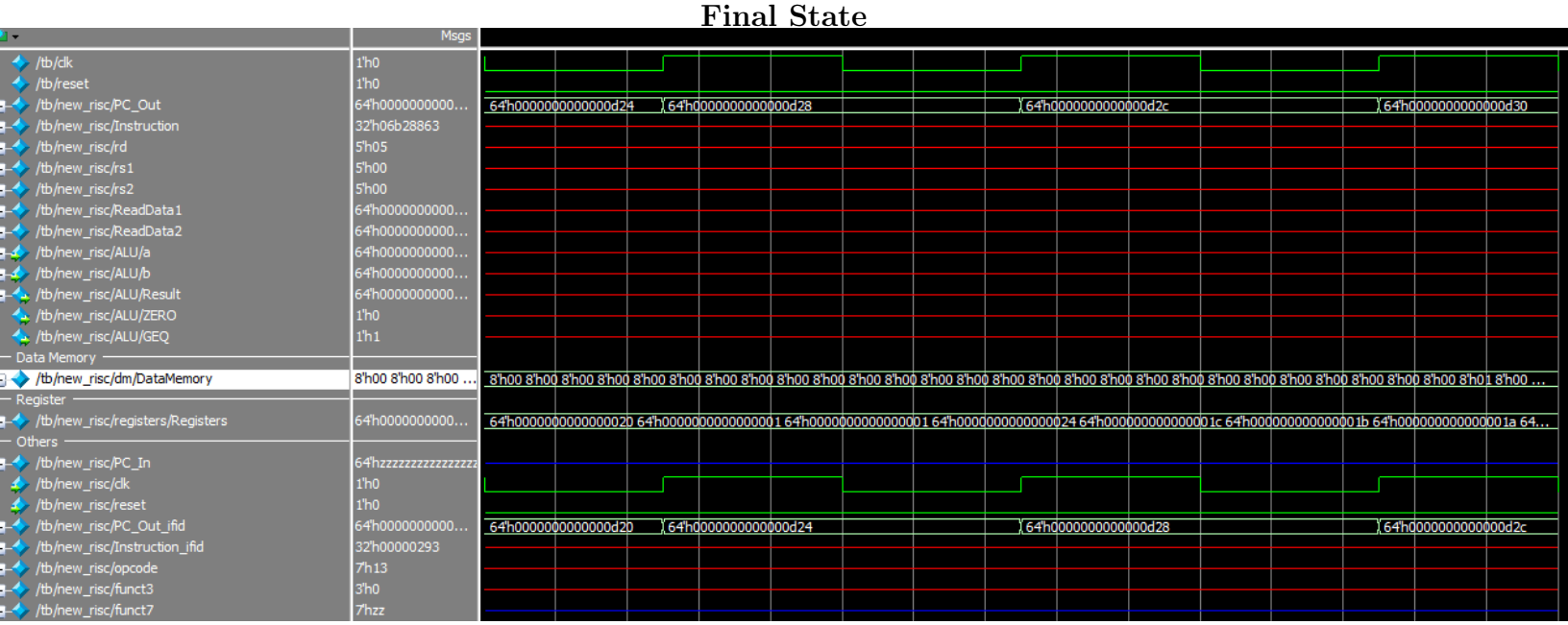



[35] 8'h00
[34] 8'h00
[33] 8'h00
[32] 8'h00
[31] 8'h00
[30] 8'h00
[29] 8'h00
[28] 8'h04
[27] 8'h00
[26] 8'h00
[25] 8'h00
[24] 8'h00
[23] 8'h00
[22] 8'h00
[21] 8'h00
[20] 8'h03
[19] 8'h00
[18] 8'h00
[17] 8'h00
[16] 8'h00
[15] 8'h00
[14] 8'h00
[13] 8'h00
[12] 8'h02
[11] 8'h00
[10] 8'h00
[9] 8'h00
[8] 8'h00
[7] 8'h00
Now 13000 ns
Cursor 1 30 ns
2 ns 4 ns 6 ns 8 ns 10 ns 12 ns 14 ns 16 ns 18 ns 20 ns 22 ns 24 ns 26 ns 28 ns 30 ns
30 ns
[21] 8'h00
[20] 8'h03
[19] 8'h00
[18] 8'h00
[17] 8'h00
[16] 8'h00
[15] 8'h00
[14] 8'h00
[13] 8'h00
[12] 8'h02
[11] 8'h00
[10] 8'h00
[9] 8'h00
[8] 8'h00
[7] 8'h00
[6] 8'h00
[5] 8'h00
[4] 8'h01
[3] 8'h00
[2] 8'h00
[1] 8'h00
[0] 8'h00
Register

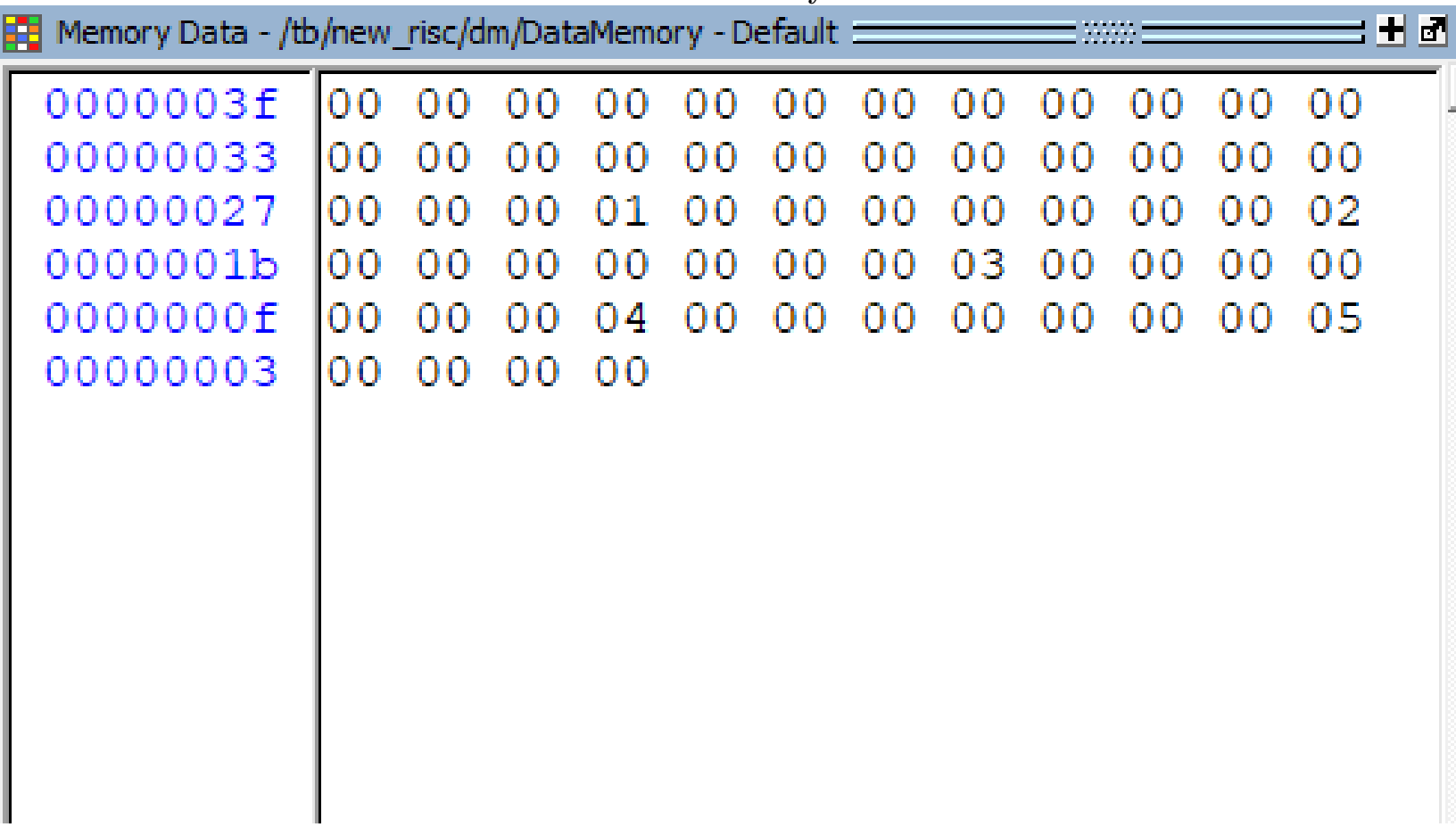
Initial Registers State

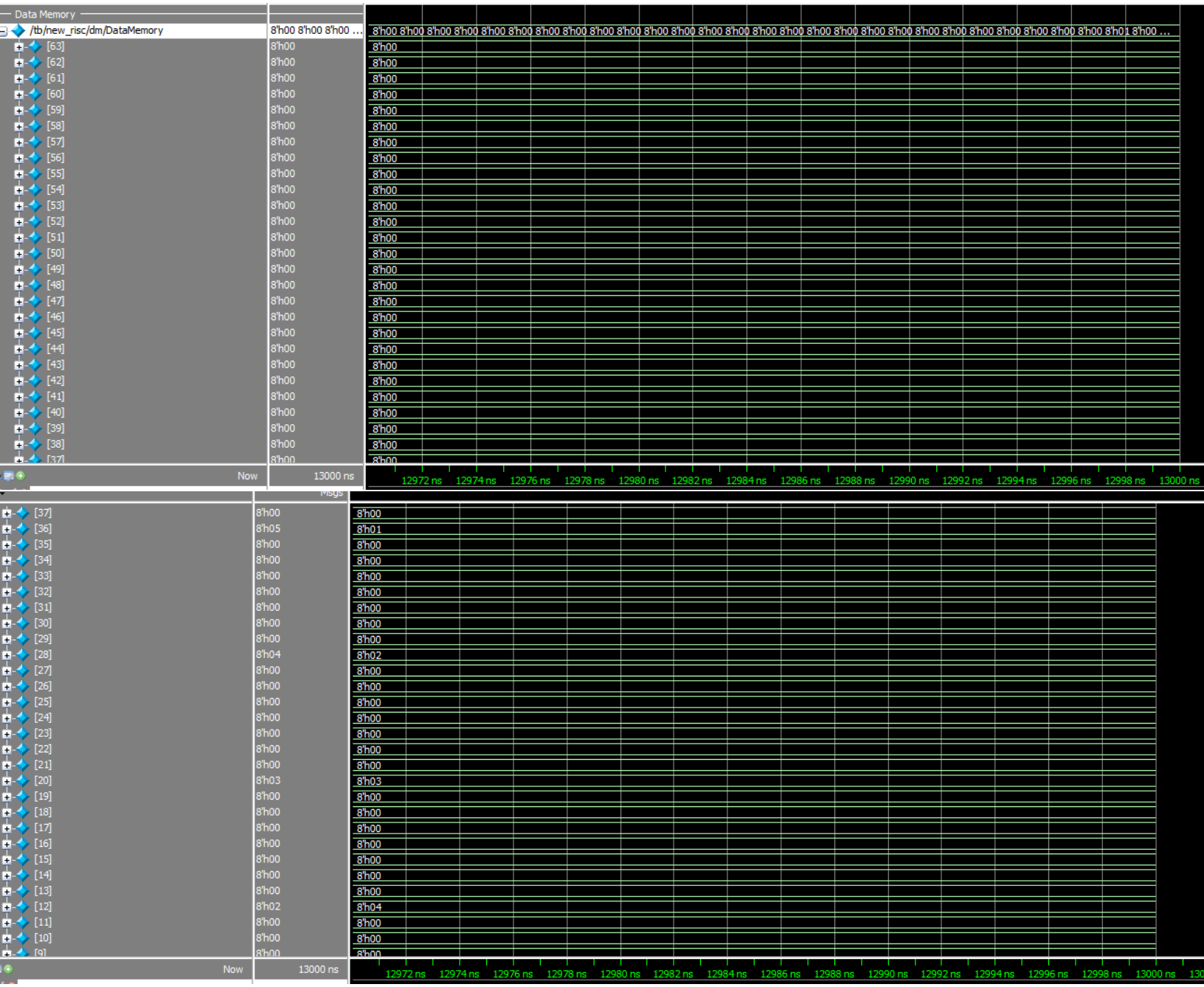


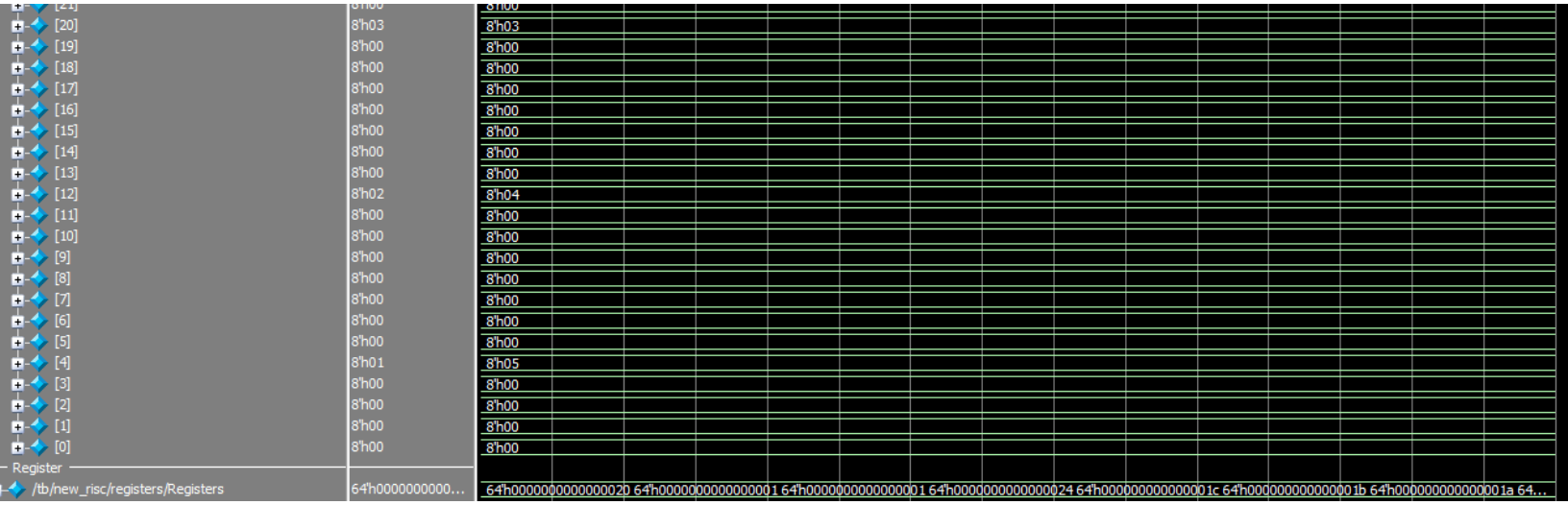
Final State



Final Data Memory State



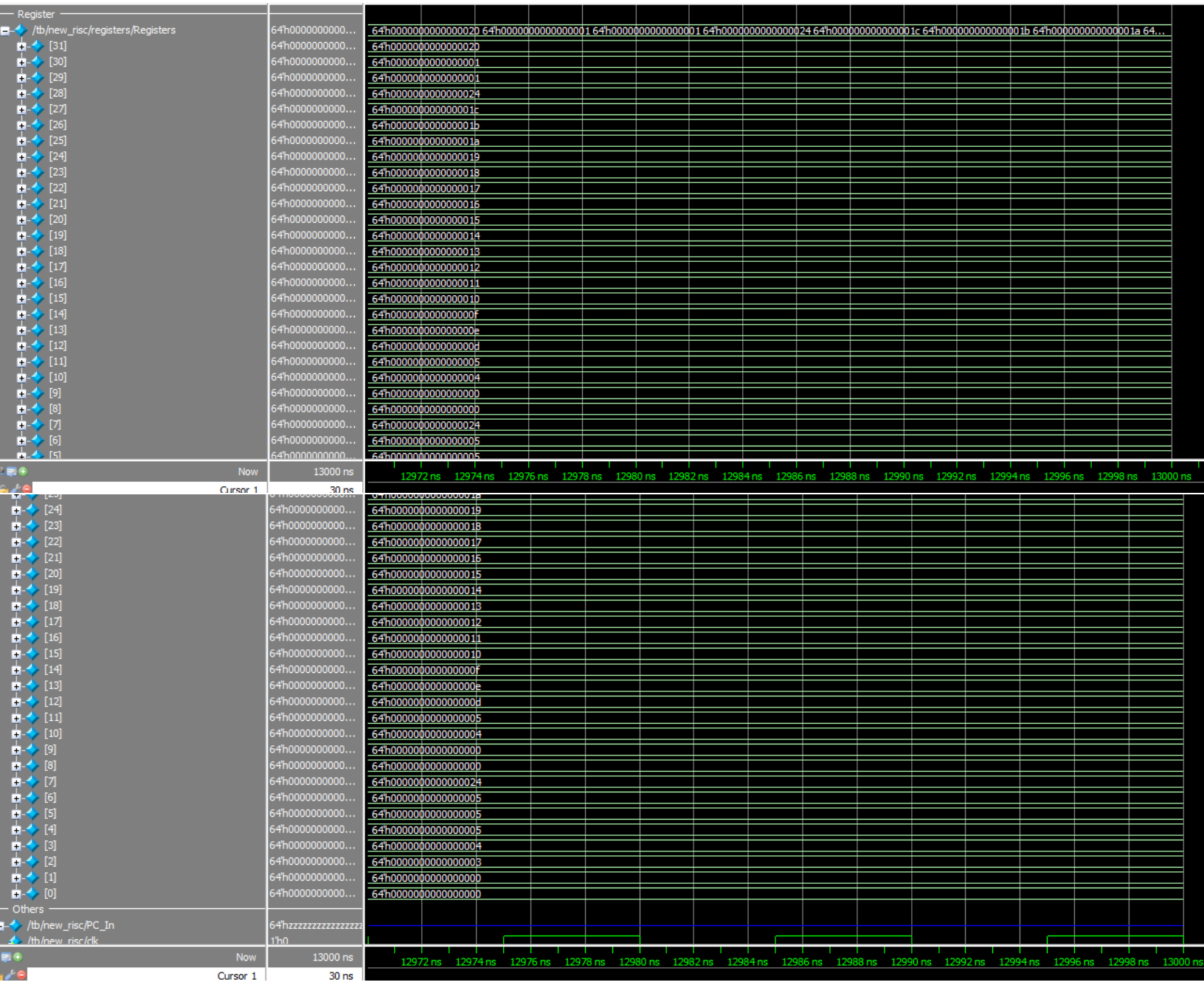

Data Memory
/tb/new_risc/dm/DataMemory
8'h00 8'h00 8'h00 ...
[63]
[62]
[61]
[60]
[59]
[58]
[57]
[56]
[55]
[54]
[53]
[52]
[51]
[50]
[49]
[48]
[47]
[46]
[45]
[44]
[43]
[42]
[41]
[40]
[39]
[38]
[37]
Now
13000 ns
12972 ns
12974 ns
12976 ns
12978 ns
12980 ns
12982 ns
12984 ns
12986 ns
12988 ns
12990 ns
12992 ns
12994 ns
12996 ns
12998 ns
13000 ns
[37]
[36]
[35]
[34]
[33]
[32]
[31]
[30]
[29]
[28]
[27]
[26]
[25]
[24]
[23]
[22]
[21]
[20]
[19]
[18]
[17]
[16]
[15]
[14]
[13]
[12]
[11]
[10]
Now
13000 ns


[20] 8'h03
[19] 8'h00
[18] 8'h00
[17] 8'h00
[16] 8'h00
[15] 8'h00
[14] 8'h00
[13] 8'h00
[12] 8'h02
[11] 8'h00
[10] 8'h00
[9] 8'h00
[8] 8'h00
[7] 8'h00
[6] 8'h00
[5] 8'h00
[4] 8'h01
[3] 8'h00
[2] 8'h00
[1] 8'h00
[0] 8'h00
Register
/tb/new_risc/registers/Registers
64'h0000000000...
8'h03
8'h00
8'h04
8'h05
64'h0000000000000020 64'h0000000000000001 64'h0000000000000001 64'h0000000000000024 64'h000000000000001c 64'h000000000000001b 64'h000000000000001a 64...

Final Registers State

Memory Data - /tb/new_risc/registers/Registers

0000001f	0000000000000020	0000000000000001
0000001d	0000000000000001	0000000000000024
0000001b	000000000000001c	000000000000001b
00000019	000000000000001a	0000000000000019
00000017	0000000000000018	0000000000000017
00000015	0000000000000016	0000000000000015
00000013	0000000000000014	0000000000000013
00000011	0000000000000012	0000000000000011
0000000f	0000000000000010	000000000000000f
0000000d	000000000000000e	000000000000000d
0000000b	0000000000000005	0000000000000004
00000009	0000000000000000	0000000000000000
00000007	0000000000000024	0000000000000005
00000005	0000000000000005	0000000000000005
00000003	0000000000000004	0000000000000003
00000001	0000000000000000	0000000000000000


Register
/tb/new_risc/registers/Registers
[31]
[30]
[29]
[28]
[27]
[26]
[25]
[24]
[23]
[22]
[21]
[20]
[19]
[18]
[17]
[16]
[15]
[14]
[13]
[12]
[11]
[10]
[9]
[8]
[7]
[6]
[5]
[4]
[3]
[2]
[1]
[0]
Others
/tb/new_risc/PC_In
/tb/new_risc/clk
Now
13000 ns
Cursor 1
30 ns
12972 ns
12974 ns
12976 ns
12978 ns
12980 ns
12982 ns
12984 ns
12986 ns
12988 ns
12990 ns
12992 ns
12994 ns
12996 ns
12998 ns
13000 ns