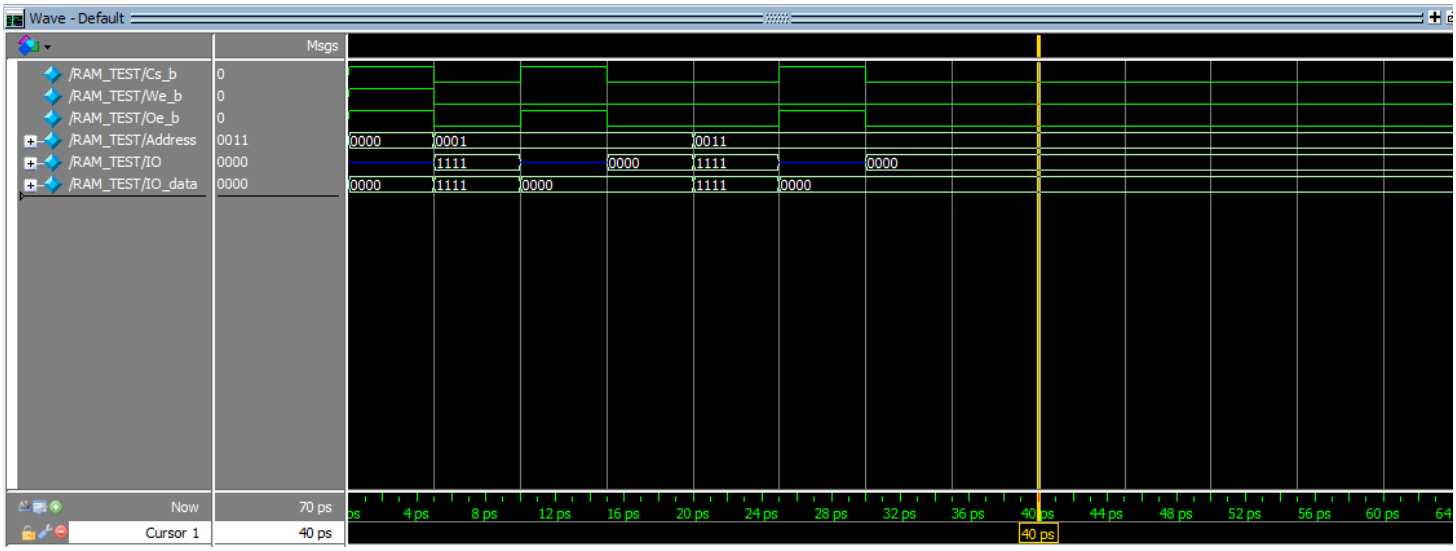
```
module simpleRAM(Cs_b, We_b, Oe_b, Address, IO);
 input Cs_b;
 input We_b;
 input Oe_b;
 input [3:0] Address;
 inout [3:0] IO;

 reg [3:0] RAM1[0:15];

 assign IO = (Cs_b==1'b1 | We_b ==1'b0 | Oe_b==1'b1) ? 4'bZZZZ :RAM1[Address];

 always @(We_b,Cs_b) begin
  @(negedge We_b)begin
     if (Cs_b==1'b0) begin
       RAM1[Address]=IO;
     end
   end
   end
endmodule
```

```
module RAM_TEST();
  reg Cs_b, We_b , Oe_b;
  reg [3:0] Address;
  wire [3:0] IO;
   reg [3:0] IO_data;
   assign IO = (We_b == 1'b0 && Cs_b==1'b0)? IO_data : 4'bzzzz ;

  simpleRAM tb1(Cs_b, We_b, Oe_b, Address, IO);
  initial begin
    Cs_b = 1'b1;
    We_b = 1'b1;
    Oe_b = 1'b1;
    Address = 0;
    IO_data = 0;
  end
  initial begin
 //INPUT
    #5 We_b = 1'b0; Oe_b = 1'b0; Cs_b = 1'b0; IO_data = 4'b1111 ; Address = 4'b0001;
    //OUTPUT
    #5 We_b = 1'b0; Oe_b = 1'b1; Cs_b = 1'b1; IO_data = 4'b0000 ; Address = 4'b0001;
    #5 We_b = 1'b0; Oe_b = 1'b0; Cs_b = 1'b0; IO_data = 4'b0000 ; Address = 4'b0001;
    //INPUT
    #5 We_b = 1'b0; Oe_b = 1'b0; Cs_b = 1'b0; IO_data = 4'b1111 ; Address = 4'b0011;
    //OUTPUT
    #5 We_b = 1'b0; Oe_b = 1'b1; Cs_b = 1'b1; IO_data = 4'b0000 ; Address = 4'b0011;
    #5 We_b = 1'b0; Oe_b = 1'b0; Cs_b = 1'b0; IO_data = 4'b0000 ; Address = 4'b0011;
    #40 $finish;
    end
endmodule
```




```
module counter(clk, reset, z);
input clk, reset;
output[2:0] z;
wire w0,w1,w2,w3,w4,w5;
reg [2:0]out;
wire [2:0]qp,q;

always @ (posedge clk) begin
if (reset == 1'b1) begin
  out <= 3'b000;
end
end
assign q =  out;
assign z[2] = qp[2];

D f1 (clk,w2,q[2],qp[2]);
D f2 (clk,w1,q[1],qp[1]);
D f3 (clk,w0,q[0],qp[0]);
xor (w2, z[2], reset);
```

```
xor (w1, z[1], reset);
xor (w0, z[0], reset);
or (w3, qp[2], q[0]);
and (w4, w3, qp[1]);
and (w5, q[2], q[1], qp[0]);
or (z[1], w5,w4);
xnor (z[0], q[2],q[0]);



endmodule
module testbench3 ();
  reg clk, reset;
  wire [2:0] z;

counter test (clk, reset, z);

initial begin
  clk = 1'b0;
  reset = 1'b0;
end
always #10 clk = ~clk;
initial begin
  #10 reset = 0;
  #100 reset = 1;
  #100 reset = 0;
  #500 $finish;
end
endmodule
```

```
module D(clk,D,Q,QP);
 input clk,D;
 output Q,QP;
 reg out;
 initial begin
 out = 1'b0;
 end
 always@(posedge clk)
  begin
  out = D;
 end
 assign  Q  = out;
 assign QP = ~out;
 endmodule
```