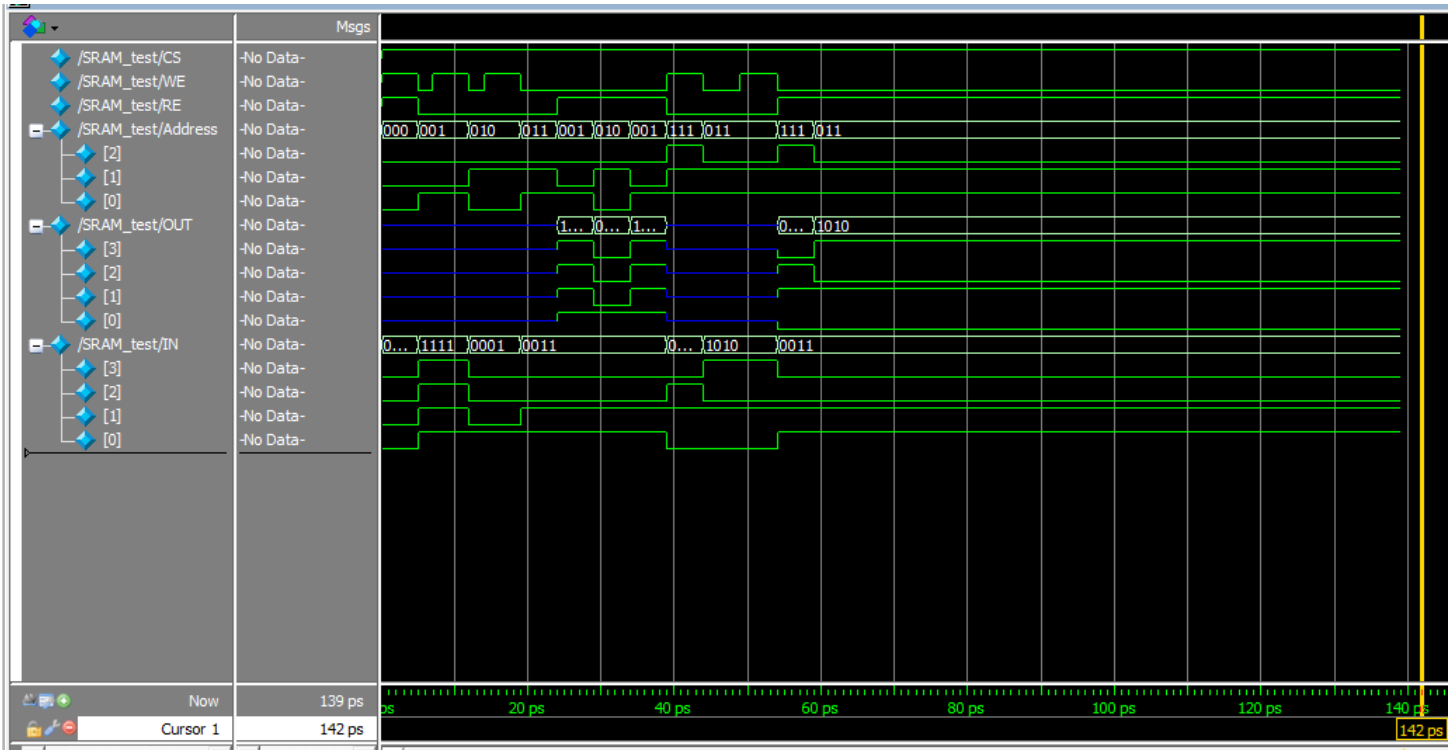
```
module RAM(CS, WE, RE, Address, IN, OUT);
  input CS,WE,RE;
  input [2:0] Address;
  input [3:0] IN;
  output [3:0] OUT;
  reg [3:0] RAM[0:7];

  reg [3:0] DIN, DOUT;

  assign OUT = DOUT;

  always @(WE==1) begin // IF WE IS 1 WRITES
DATA TO RAM LOCATION
       DIN = IN;
       if (CS==1'b1) begin // CS SHOULD BE ALWAYS
1 IF 0 CHIP TURNS OFF
        RAM[Address] = DIN; // STORES DATA ON
RAM
     end
     end

  always @(RE,Address) begin // ANY CHANGES
ON BOTH WILL CHECK CONDITIONS BELOW
       if (CS==1'b1 && RE==1 && WE == 0) begin //
IF TRUE READS DATA on RAM ADDRESS GIVEN
        DOUT = RAM[Address];
     end
     else  DOUT =  4'bZZZZ ; //  SETS OUTPUT IN
IDLE / WAITING STATE
     end

endmodule
```

```
module SRAM_test();
  reg CS, WE , RE;
  reg [2:0] Address;
  wire [3:0] OUT;
  reg [3:0] IN;
  RAM TEST1(CS, WE, RE, Address, IN, OUT);
  initial begin
   CS = 1'b1;
   WE = 1'b1;
   RE = 1'b1;
   Address = 0;
   IN = 0;
 end
  initial begin
   #5 WE = 1'b0; RE = 1'b0; CS = 1'b1; IN =
4'b1111 ; Address = 3'b001;
   #2 WE = 1'b1; RE = 1'b0; CS = 1'b1; IN =
4'b1111 ; Address = 3'b001;
   #5 WE = 1'b0; RE = 1'b0; CS = 1'b1; IN =
4'b0001 ; Address = 3'b010;
   #2 WE = 1'b1; RE = 1'b0; CS = 1'b1; IN =
4'b0001 ; Address = 3'b010;
   #5 WE = 1'b0; RE = 1'b0; CS = 1'b1; IN =
4'b0011 ; Address = 3'b011;
   #5 WE = 1'b0; RE = 1'b1; CS = 1'b1; IN =
4'b0011 ; Address = 3'b001;
   #5 WE = 1'b0; RE = 1'b1; CS = 1'b1; IN =
4'b0011 ; Address = 3'b010;
   #5 WE = 1'b0; RE = 1'b1; CS = 1'b1; IN =
4'b0011 ; Address = 3'b001;
   //INPUT
   #5 WE = 1'b1; RE = 1'b0; CS = 1'b1; IN =
4'b0110 ; Address = 3'b111;
   #5 WE = 1'b0; RE = 1'b0; CS = 1'b1; IN =
4'b1010 ; Address = 3'b011;
   #5 WE = 1'b1; RE = 1'b0; CS = 1'b1; IN =
4'b1010 ; Address = 3'b011;
   //OUTPUT
   #5 WE = 1'b0; RE = 1'b1; CS = 1'b1; IN =
4'b0011 ; Address = 3'b111;
   #5 WE = 1'b0; RE = 1'b1; CS = 1'b1; IN =
4'b0011 ; Address = 3'b011;

   #80 $finish;
   end
endmodule
```


Msgs
/SRAM_test/CS
/SRAM_test/WE
/SRAM_test/RE
/SRAM_test/Address
[2]
[1]
[0]
/SRAM_test/OUT
[3]
[2]
[1]
[0]
/SRAM_test/IN
[3]
[2]
[1]
[0]
-No Data-
000
001
010
011
001
010
001
111
011
111
011
1...
0...
1...
0...
1010
0...
1111
0001
0011
0...
1010
0011
Now
139 ps
Cursor 1
142 ps
20 ps
40 ps
60 ps
80 ps
100 ps
120 ps
140 ps
142 ps