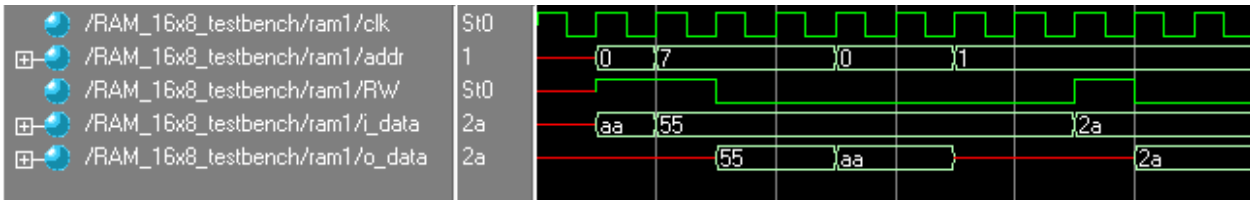
```
module RAM_16x8_ex1 (clk, addr, RW, i_data, o_data);

     input clk;
     input [3:0] addr;
     input RW;
     input [7:0] i_data;
     output reg[7:0] o_data;

     reg [7:0] Mem [0:15];

     always @ (posedge clk)
               if (RW)
                         Mem[addr] <= i_data;
               else
                         o_data <= Mem[addr];

endmodule
```

```
module RAM_16x8_ex1_testbench();

     reg clk;
     reg [3:0] address;
     reg RW_enable;
     reg [7:0] data_in;
     wire [7:0] data_out;

     RAM_16x8_ex1 ram1 (
        .clk(clk),
        .addr(address),
        .RW(RW_enable),
        .i_data(data_in),
        .o_data(data_out));

     initial begin
        $display("RAM test bench.");

        clk = 1;
        #10 RW_enable = 1;
        address = 0;
        data_in = 8'haa;  //1010 1010
        #10 address = 7;
        data_in = 8'h55;  //0101 0101

        #10 RW_enable = 0;
        #10 $display($time, "  Mem[%d] = %h", address, data_out);
        #10 address = 0;
        #10 $display($time, "  Mem[%d] = %h", address, data_out);
        #10 address = 1;
        #10 $display($time, "  Mem[%d] = %h", address, data_out);

        #10 RW_enable = 1;
        address = 1;
        data_in = 8'h2a;  //0010 1010

        #10 RW_enable = 0;
        #10 $display($time, "  Mem[%d] = %h", address, data_out);

        #200 $finish;
    end

    always begin
        #5 clk = ~clk;
    end

endmodule
```




```
# RAM test bench.
#                40  Mem[ 7] = 55
#                60  Mem[ 0] = aa
#                80  Mem[ 1] = xx
#               110  Mem[ 1] = 2a
```