```
////////////////////////////////////////////////////////////////////////////////
//
// Company:
// Engineer:        Ralnikov Vadim Dmitryevich
//
// Create Date:     20:00 04/03/2019
// Design Name:
// Module Name:     test_bench
// Project Name:
// Target Devices:
// Tool versions:
// Description:     test_bench for memory RAM
//
////////////////////////////////////////////////////////////////////////////////

`timescale 1ns / 100ps
`default_nettype none


module test_bench;

parameter PERIOD = 10;


wire                CLK;                                //  in, u[ 1], Clock freq 100MHz
wire                EN;                                 //  in, u[ 1], enable work 1 -enable
wire  [  10 :  0 ]  ADDRESS;                            //  in, u[ 1], address input 0 to 2048
wire                WE;                                 //  in, u[ 1], enable write 1 - enable
wire  [  7 :  0 ]   DI;                                 //  in, u[ 8], input data, parallel

wire  [  7 :  0 ]   DO;                                 // out, u[ 8], out data, parallel




reg                 r_clk     =  1'b0;
reg                 r_en      =  1'b0;
reg   [  10 :  0 ]  r_adr     = 11'b0;
reg                 r_we      =  1'b0;
reg   [  7 :  0 ]   r_di      =  8'b0;
reg   [  7 :  0 ]   r_counter =  8'b11111111;


assign CLK     = r_clk;
assign EN      = r_en;
assign ADDRESS = r_adr;
assign WE      = r_we;
assign DI      = r_di;


integer    i;

//-----

task do_write;

input  [  10 : 0 ]  i_addr;
input  [  7 : 0 ]   i_data;

   begin

  #20

  @(posedge CLK);

  r_en = 1;
  r_we = 1;
  r_adr = i_addr;
  r_di = i_data;

  #20

  @(posedge CLK);

  r_we = 0;

   end

endtask

//-----

task do_read;

input  [  10 : 0 ]  i_addr;

   begin

  #20

  r_we = 0;

  @(posedge CLK);

  r_en = 1;
  r_we = 0;
  r_adr = i_addr;

   end

endtask

//-----

initial begin
            forever
            #(PERIOD/2) r_clk = ~r_clk;
        end


always @(posedge WE)                                    //  cycle for data write
   begin
       r_counter <= r_counter - 8'b00000001;
         if(r_counter==0)
       r_counter<=8'b11111111;

   end



initial begin                                           //  cycle for address write

  for (i=0; i<2048; i = i+1)
    begin

#10;

do_write (i, r_counter);

   end

#10;
do_read (0);
#40;
do_read (11'b00000000001);
#40;
do_read (11'b00000000111);
#40;
do_read (11'b11110000111);
#40;
do_read (11'b11111111111);

   end

```

```
memory
    memory_inst
    (
        .CLK                    (CLK),                  //  in, u[ 1], Clock freq 100MHz
        .EN                     (EN),                   //  in, u[ 1], enable work 1 -enable
        .ADDRESS                (ADDRESS),              //  in, u[11], address from 0 to 2048
        .WE                     (WE),                   //  in, u[ 1], write enable, 1 -enable
        .DI                     (DI),                   //  in, u[ 8], input data, parallel

        .DO                     (DO)                    // out, u[ 8], output data, parallel
    );

endmodule
```