

```

module RAM_16x8_ex2 (clk, CS, W, R, addr, i_data, o_data);

    input clk;
    input CS, W, R;
    input [3:0] addr;
    input [7:0] i_data;
    output reg [7:0] o_data;

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

    always @ (CS or W or R or posedge clk)
        if(CS && W)
            Mem[addr] <= i_data;
        else if (CS && R)
            o_data <= Mem[addr];

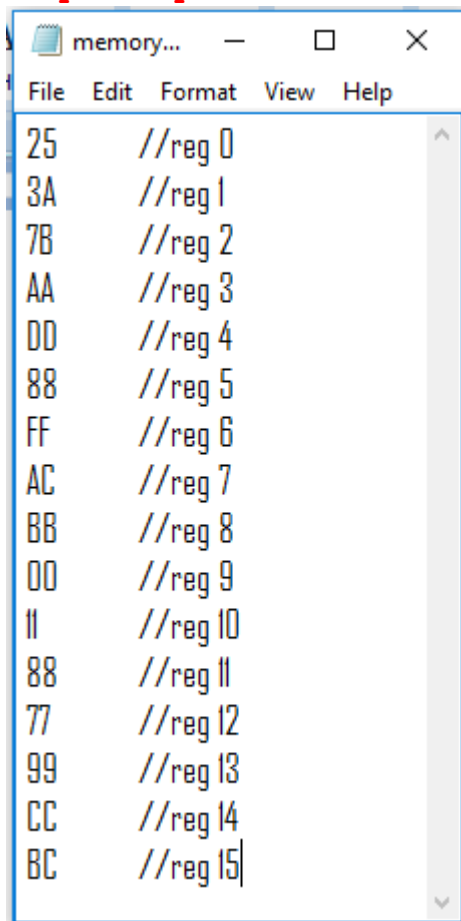
    initial //initializing Mem
        $readmemh ("memory.dat", Mem);

    Initial //Displaying the data read from Mem
        $monitor ($time, " %h", o_data);

endmodule

```

**//memory.dat. The .dat file must be in the same directory as design and testbench. Otherwise you will need to specify the complete path.**



```

module RAM_16x8_ex2_testbench();

    reg clk;
    reg [3:0] address;
    reg CE, write, read;
    reg [7:0] data_in;
    wire [7:0] data_out;

    RAM_16x8_ex2 ram1 (
        .clk(clk),
        .addr(address),
        .CS (CE),
        .W(write),
        .R(read),
        .i_data(data_in),
        .o_data(data_out));

    initial begin
        $display("Reading from Memory (already
initialized/populated using $readmemh)...");

        clk = 1;
        CE = 0;
        read = 0;
        #10 read = 1;
        #50 CE = 1;
        #10 address = 15;
        #10 address = 1;
        #10 address = 2;
        #10 address = 3;
        #10 address = 7;
        #10 address = 6;
        #10 address = 5;
        #10 address = 4;
        #10 address = 8;
        #10 address = 9;
        #10 address = 10;
        #10 address = 11;
        #10 address = 0;
        #10 address = 14;
        #10 address = 13;
        #10 address = 12;

        #500 $finish;
    end

    always begin
        #5 clk = ~clk;
    end

endmodule

```

run -all

# Reading from Memory (already initialized/populated using \$readmemh)...

#	0	xx
#	70	bc
#	80	3a
#	90	7b
#	100	aa
#	110	ac
#	120	ff
#	130	88
#	140	dd
#	150	bb
#	160	00
#	170	11
#	180	88
#	190	25
#	200	cc
#	210	99
#	220	77