

```
1  /*-----
2  Name Lamin Jammeh
3  Class: EE417 Summer 2024
4  Lesson 11 HW Question 2
5  Group: Ron Kalin/ Lamin Jammeh
6  Project Description: Testbench for the RAM Module
7  -----*/
8
9  module RAM_tb ();
10
11  parameter memory_height = 8;
12  parameter data_width    = 4;
13
14  //define the wires and registers testbench
15  reg          clock, reset;
16  reg          read_or_write;
17  reg [memory_height-1:0] address;
18  reg [data_width-1:0] data_in;
19  wire [data_width-1:0] data_out;
20
21  //probe the each address line to determine the content
22  wire [memory_height-1:0] address_0, address_1, address_2, address_3, address_4,
    address_5, address_6, address_7;
23
24  //instantiate the Unit under test UUT
25  RAM UUT (clock, reset, address, read_or_write, data_in, data_out);
26
27  //assign the probes for the address to the address to lines in the UUT
28  assign address_0 = UUT.memory[0];
29  assign address_1 = UUT.memory[1];
30  assign address_2 = UUT.memory[2];
31  assign address_3 = UUT.memory[3];
32  assign address_4 = UUT.memory[4];
33  assign address_5 = UUT.memory[5];
34  assign address_6 = UUT.memory[6];
35  assign address_7 = UUT.memory[7];
36
37  //intantiate the clock cycle
38  initial
39  begin
40      clock = 0;
41      forever #5 clock = ~clock;
42  end
43
44  //initialize the RAM
45  initial
46  begin
47      reset      = 1;
48      read_or_write = 0;
49      address     = 0;
50      data_in     = 4'b0000;
51
52
53
54  //load data to the memory
55  #10;
56      reset      = 1'b0;
57      address     = 0;
58      data_in     = 4'b1101;
59      read_or_write = 1'b1;
60
61  #10;
62      data_in     = 4'b0011;
63      address     = 1;
64      read_or_write = 1'b1;
65
66  #10;
```

```
67         data_in      = 4'b1100;
68         address      = 2;           //load address 2
69         read_or_write = 1'b1;       // write to memory
70
71     #10;
72         data_in      = 4'b1010;
73         address      = 3;           //load address 3
74         read_or_write = 1'b1;       // write to memory
75     #10;
76         address      = 4;           //load address 4
77         data_in      = 4'b1101;
78         read_or_write = 1'b1;       // write to memory
79
80     #20;
81         data_in      = 4'b0011;
82         address      = 5;           //load address 5
83         read_or_write = 1'b1;       // write to memory
84
85     #10;
86         data_in      = 4'b1100;
87         address      = 6;           //load address 6
88         read_or_write = 1'b1;       // write to memory
89
90     #10;
91         data_in      = 4'b1010;
92         address      = 7;           //load address 7
93         read_or_write = 1'b1;       // write to memory
94
95     //read data from memory
96     #10;
97         address = 0;
98         read_or_write = 1'b0;
99     #10;
100        address = 1;
101        read_or_write = 1'b0;
102    #10;
103        address = 2;
104        read_or_write = 1'b0;
105    #10;
106        address = 3;
107        read_or_write = 1'b0;
108    #10;
109        address = 4;
110        read_or_write = 1'b0;
111    #10;
112        address = 5;
113        read_or_write = 1'b0;
114    #10;
115        address = 6;
116        read_or_write = 1'b0;
117    #10;
118        address = 7;
119        read_or_write = 1'b0;
120    //check the reset function
121    #10;
122        reset = 1;
123    #10;
124        reset = 0;
125    #20    $stop;
126    end
127    endmodule
```