```
1  //Name: Ron Kalin  EE417 Summer 2024, Date: 07/25/24
2  //Group: Ron Kalin/ Lamin Jammeh
3  //Lesson 11 HW Question 01. Project: test-bench for FIFO
4  module FIFO_tb ();
5  parameter   stack_width =  5;   // width of stack and data paths (word length)
6  parameter   stack_height = 8;   // height of stack (# of words)
7  parameter stack_ptr_width = $clog2(stack_height); //3;// width of pointer to address stack
8  //declare outputs and inputs
9  wire  [stack_width -1 : 0]  Data_out;
10 wire                        stack_empty, stack_full;
11 wire                        stack_almost_empty, stack_almost_full;
12 reg   [stack_width -1 : 0]  Data_in;
13 reg   [stack_width -1 : 0]  Data_in_array [stack_height-1 : 0]; //internal register array
   of data
14 reg                         clk, rst;
15 reg                         write_to_stack, read_from_stack;
16
17 wire  [stack_width-1: 0]    stack0, stack1, stack2, stack3,
18                             stack4, stack5, stack6, stack7;
19 wire [stack_ptr_width-1: 0] read_ptr, write_ptr;
20 wire [stack_ptr_width  : 0] ptr_diff; //pointer difference
21 integer                     j;
22
23 //initialize the unit under test M1
24 FIFO  M1 ( Data_out, stack_empty, stack_almost_empty, stack_almost_full, stack_full,
   Data_in,
25            write_to_stack, read_from_stack, clk, rst );
26 // assign probe wires for troubleshooting and visibility
27 assign   stack0  = M1.stack [0];
28 assign   stack1  = M1.stack [1];
29 assign   stack2  = M1.stack [2];
30 assign   stack3  = M1.stack [3];
31 assign   stack4  = M1.stack [4];
32 assign   stack5  = M1.stack [5];
33 assign   stack6  = M1.stack [6];
34 assign   stack7  = M1.stack [7];
35 assign  read_ptr = M1.read_ptr;
36 assign write_ptr = M1.write_ptr;
37 assign  ptr_diff = M1.ptr_diff;
38
39 always begin clk = 0; forever #5 clk = ~clk; end  // create clock
40 initial  #1500 $stop;
41
42 initial begin
43    #10   rst = 1;   #40 rst = 0;    #420  rst = 1;   # 460 rst = 0;  end  // cycle reset
44
45 initial begin
46    $readmemb ("FIFO_Input_Data.txt", Data_in_array); // read data to Data_in_array from
   text file saved in simulation path
47    end
48
49 initial begin
50 //#80 Data_in = 1; forever #10 Data_in = Data_in + 1;  //manual method for creating Data_in
51    #80 Data_in = 1;
52    for (j=0; j<stack_height; j=j+1)
53      begin
54        Data_in = Data_in_array[j]; // data in coming from text file
55        $display("Data_out: %u, Data_in: %u", Data_out, Data_in );
56      end //for
57 end
58
59 initial fork  // cycling read/write inputs
60    # 80 write_to_stack  = 1;
61    #180 write_to_stack  = 0;
62    #250 read_from_stack = 1;
63    #350 read_from_stack = 0;
64    #420 write_to_stack  = 1;
65    #480 write_to_stack  = 0;
66   join
```

```
67
68    //display results
69    always @(posedge clk)
70    begin
71      $display("Data_out: %u, Data_in: %u, Data_in_array: %u, write_to_stack: %b,
      read_from_stack: %b, stack_empty: %b, stack_full: %b",
72                Data_out, Data_in, Data_in_array, write_to_stack, read_from_stack, stack_empty,
      stack_full );
73    end
74
75    endmodule
```