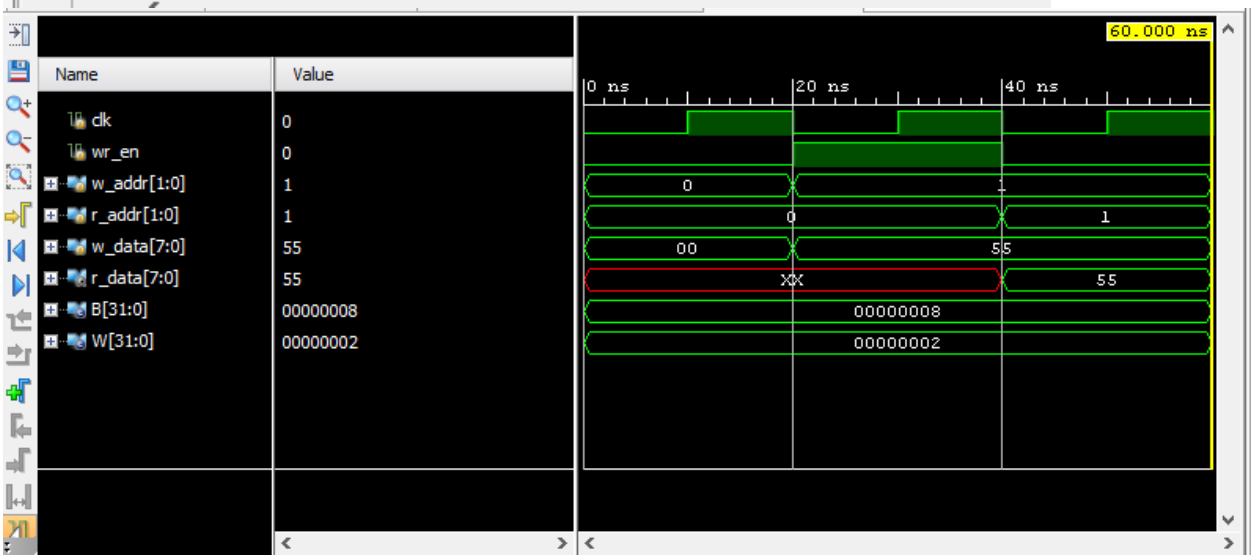
Handout#05: Test Bench Generation for Sequential Circuits

1. Write down the Verilog test benches for the following designs. Designs are available in the Lab#04. Submit the Verilog codes for testbenches and simulation timing waveforms in the form of screenshots.

a. Register File (Listing 4.5)

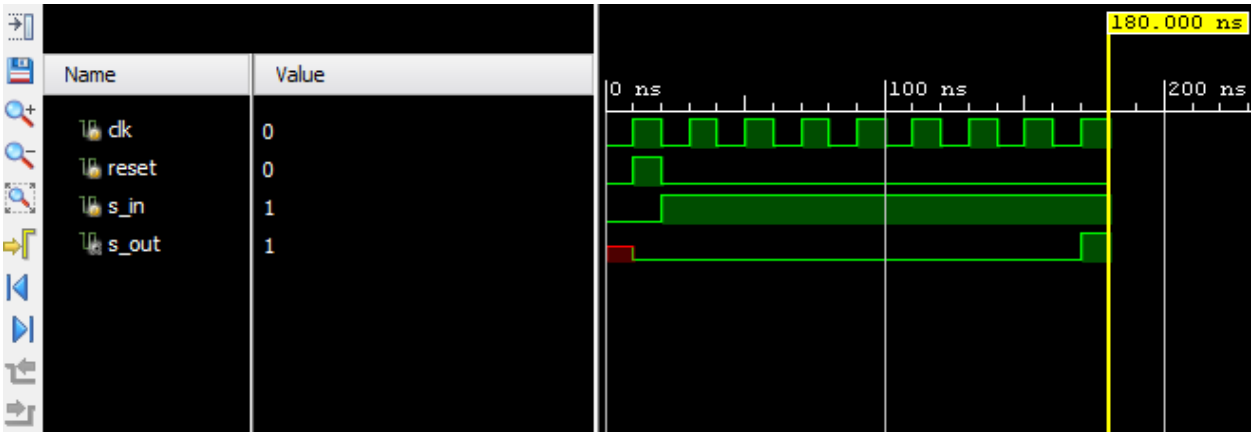
```
50  //Test bench for Register File
51  module Register_file_TB();
52  parameter B = 8,
53            W = 2;
54  reg clk;
55  reg wr_en;
56  reg [W-1:0] w_addr, r_addr;
57  reg [B-1:0] w_data;
58  wire [B-1:0] r_data;
59
60  //instantiation register file
61  Register_File RF0(clk, wr_en, w_addr, r_addr, w_data, r_data);
62
63  //clock
64  initial
65  clk = 0;
66
67  always
68  #10 clk = ~ clk;
69
70  //simulation
71  initial
72  begin
73      wr_en = 0;
```

```
71  initial
72  begin
73      wr_en = 0;
74      w_addr = 0;
75      r_addr = 0;
76      w_data = 0;
77      @(negedge clk);
78
79      wr_en = 1; w_data = 8'b01010101; w_addr = 2'b01;
80      @(negedge clk);
81
82      wr_en = 0; r_addr = 2'b01;
83      @(negedge clk);
84      $stop;
85  end
86
87  endmodule
88
```



b. Free-running shift register (Listing 4.6)

```
48  // Malak Majeed ullah khan //Test Bench for free running shift register
49  module FR_Run_SHFT();
50  reg clk, reset;
51  reg s_in;
52  wire s_out;
53
54  //istantiation
55  Free_Running_shift_reg FR0(clk, reset, s_in, s_out);
56
57  //clock
58  initial
59  clk = 0;
60
61  always
62  #10 clk = ~ clk;
63  initial   //simulation
64  begin
65  reset = 0; s_in = 0;
66  #10 reset = 1;
67  @(negedge clk) reset = 0;
68
69  s_in = 1;
70  repeat(8) @(negedge clk);
71  $stop;
```



c. Universal shift register (Listing 4.7)

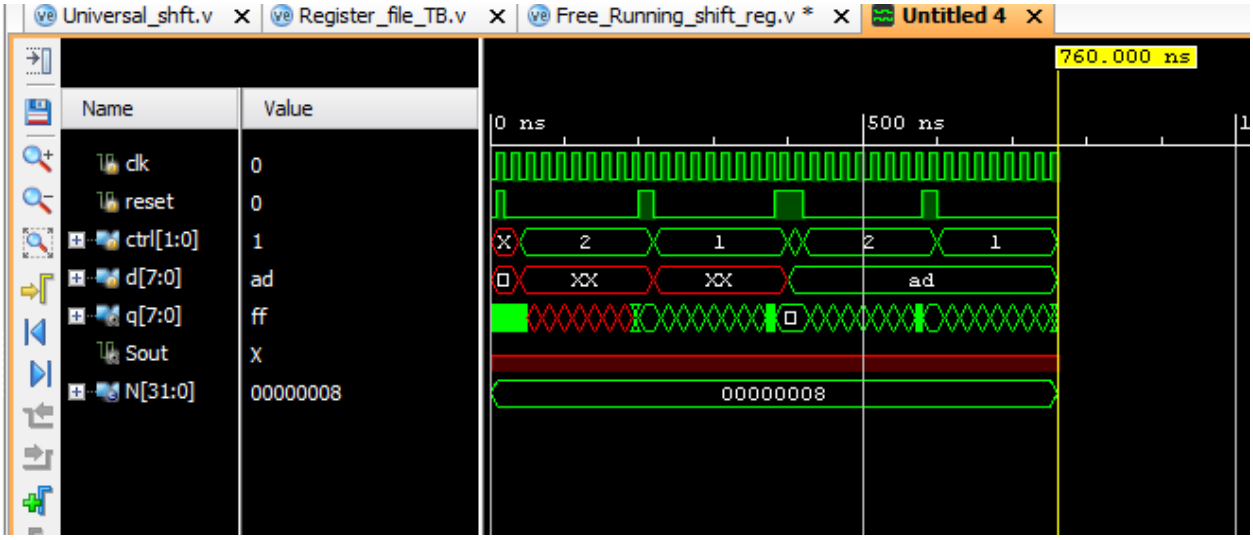
Test bench

```
// Malak Majeedullah khan
module Universal_shft_TB;
parameter N = 8;
//inputs and outputs declaration
reg clk, reset;
reg [1:0] ctrl;
reg [N-1:0] d;
wire [N-1:0] q;
wire Sout;


//instantiation
Universal_shft UNS0(clk, reset, ctrl, d, q, Sout);
//clock generation
initial
begin
    clk = 0;
end
always
#10 clk = ~clk;
initial
begin
//reset
reset = 0;
```

```
reset = 0;
#10 reset = 1;
//unreset snd test functionality
 @(negedge clk) reset = 0;
//------------------
// SIPO, left shifting
   @(negedge clk); ctrl = 2'b10; d[0] = 1'b1;
   repeat(8) @(negedge clk);
//------------------
reset = 1;
//SIPO, right shifting
 @(negedge clk); reset = 0; ctrl = 2'b01; d[7] = 1'b1;
 repeat(8) @(negedge clk);
 //-------------
reset = 1;
 //PIPO, loading condition
  @(negedge clk); reset = 0; ctrl = 2'b11; d = 8'b10101101;

 //-------------
 reset = 1;
 //PIPO, left shifting
 @(negedge clk); reset = 0;ctrl = 2'b10; d = 8'b10101101;
 repeat(8) @(negedge clk);
 //-----------------
  reset = 1;
```

```
 //PIPO, right shifting
 @(negedge clk); reset = 0;ctrl = 2'b01; d = 8'b10101101;
 repeat(8) @(negedge clk);
 $stop;
end



endmodule
```

Simulation result


Universal_shft.v
Register_file_TB.v
Free_Running_shift_reg.v *
Untitled 4
Name
Value
clk
0
reset
0
ctrl[1:0]
1
d[7:0]
ad
q[7:0]
ff
Sout
X
N[31:0]
00000008
760.000 ns
0 ns
500 ns
2
1
2
1
XX
XX
ad
00000008