Assignment 3 Report

Group member: Deependra Shukla

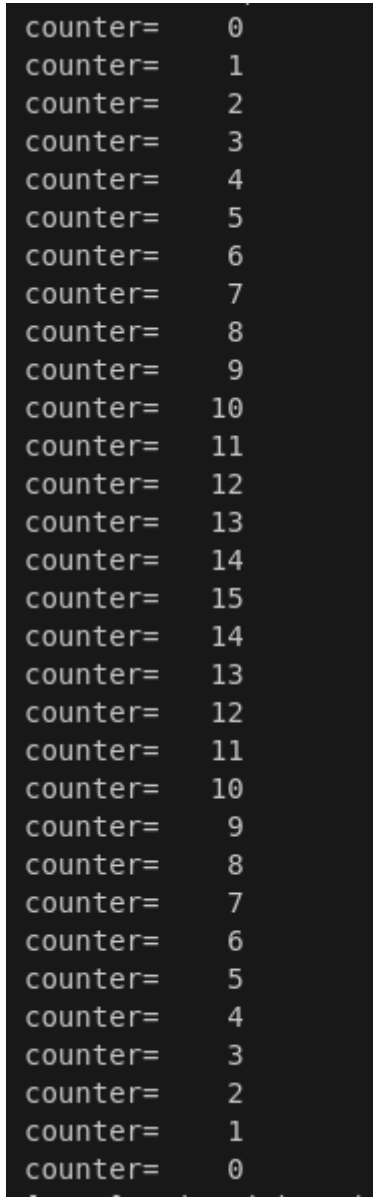
Group member: Suraj Kumar Prajapati

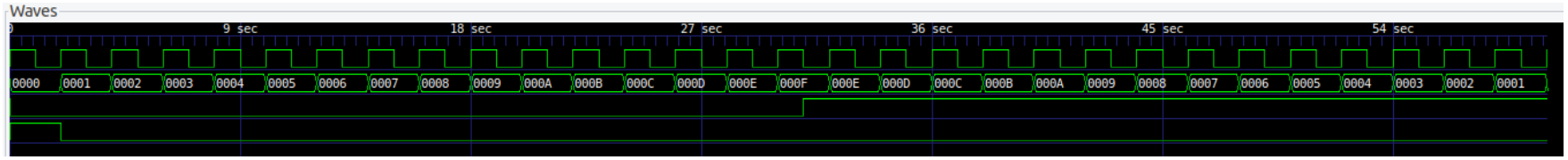
1. Write Verilog code for a mod 16 synchronous counter along with the test bench.

```
`include "tff.v"
module mode_counter(clk, reset, mode, counter);
parameter size=16;
input clk,reset,mode;
output wire [size-1:0]counter;
wire [size-1:0]qb;
wire [size-1:0] t;
wire inv_mode;
wire [size-2:0]adv1, adv2, xr;
assign t[0] =1'd1;

tff tff_inst1(clk,reset,t[0],counter[0],qb[0]);

genvar i;
for (i=1; i<size; i=i+1) begin

if (i==1) begin
  assign adv1[i-1] = counter[i-1] & ~mode;
  assign adv2[i-1] = qb[i-1] & mode;
  assign xr[i-1] = adv1[i-1] ^ adv2[i-1];
end

else begin
    assign adv1[i-1] = counter[i-1] & adv1[i-2];
    assign adv2[i-1] = qb[i-1] & adv2[i-2];
    assign xr[i-1] = adv1[i-1] ^ adv2[i-1];
end
    tff tff_inst2(clk,reset,xr[i-1],counter[i],qb[i]);
end

endmodule
```

```
`include "mode_counter.v"

module tb_model_counter();
reg clk, reset,mode;
wire [15:0]counter;

mode_counter #(.size(16)) c1(clk, reset,mode,  counter);

initial begin
clk <= 1;
forever begin
#1 clk <= ~clk;
end
end

initial begin
$monitor("counter=%d", counter);

reset = 1'b1;
mode=1'd0;
#2 reset = 1'b0;
#29 mode=1'd1;
#29 $finish;

end

initial begin
    $dumpfile("out.vcd");
    $dumpvars;
  end
endmodule
```

Terminal Output:





2. Write Verilog code for a 4-bit universal shift register along with the test bench. Instead of writing the entire code into a single file, you have to create modules for the flip-flops and call them into your main file.

```
module univ_shift_reg (
    input wire clk, rst,
    input wire [1:0] ctrl,
    input wire [3:0] d,
    output wire [3:0] q
);

reg [3:0] r_current, r_next;
always @(posedge clk or posedge rst) begin
    if (rst)
        r_current <= 4'b0000;
    else
        r_current <= r_next;
end
always @* begin
    case (ctrl)
        2'b00: r_next = r_current;
        2'b01: r_next = {r_current[2:0], d[0]};
        2'b10: r_next = {d[3], r_current[3:1]};
        default: r_next = d;
    endcase
end
assign q = r_current;
endmodule
```

```
VCD info: dumpfile out.vcd opened for output.
ctrl = 11, input = 1010, output = xxxx
ctrl = 11, input = 1010, output = 1010
ctrl = 01, input = 1010, output = 1010
ctrl = 01, input = 1010, output = 0100
ctrl = 10, input = 1010, output = 0100
ctrl = 10, input = 1010, output = 1010
ctrl = 00, input = 1010, output = 1010
ctrl = 11, input = 1010, output = 1010
ctrl = 01, input = 1010, output = 1010
ctrl = 01, input = 1010, output = 0100
ctrl = 01, input = 1010, output = 0100
ctrl = 01, input = 1010, output = 1000
ctrl = 01, input = 1010, output = 1000
ctrl = 01, input = 1010, output = 0000
ctrl = 11, input = 1010, output = 0000
ctrl = 11, input = 1010, output = 1010
ctrl = 10, input = 1010, output = 1010
ctrl = 10, input = 1010, output = 1101
ctrl = 10, input = 1010, output = 1101
ctrl = 10, input = 1010, output = 1110
ctrl = 10, input = 1010, output = 1110
ctrl = 10, input = 1010, output = 1111
ctrl = 10, input = 1010, output = 1111
[Done] exit with code=0 in 0.042 seconds
```