DIGITAL DESIGN CS223

SECTION 02

LAB 05

Sümeyye ACAR

22103640

22.05.2023

1.0 SystemVerilog module for synchronously resettable D flip-flop

```
// D Flip-Flop
module d_ff( input logic clk, reset, cin,
output logic cout );

always @(posedge clk) begin
if (reset) begin
cout = 0;
end
else begin
cout <= cin;
end
end
endmodule
```

2.0 Structural SystemVerilog module for the shift register using the D flip-flop module

```
// Shift Register
module s_reg ( input logic clk, reset, par_load, shift, [7:0] in,
               input logic in2,
               output logic [7:0] cout );
   always @(posedge clk) begin
      if (reset) begin
         cout <= 8'b00000000;
      end
      else if (shift) begin
         cout <= cout >> 1;
         cout[7] <= in2;
      end
      else if (par_load) begin
         cout <= in;
      end
   end
endmodule
```

3.0 Structural SystemVerilog module for the serial adder using the shift register, full adder, and D flip-flop

```
module s_adder( input logic clk, reset, parload, shift, [7:0] a, [7:0] b,
                output logic [7:0] sum,
                output logic outt );

    logic [7:0] arr, arr2;
    logic temp, temp2, cout, temp_sum, sgn;
    integer count = 0;

    s_reg sr1(clk, reset, parload, sgn, a, 1'b0, arr);
    s_reg sr2(clk, reset, parload, sgn, b, 1'b0, arr2);
    s_reg sr3(clk, reset, parload, sgn, 8'b00000000, temp_sum, sum);

    d_ff dff(clk, reset, temp2, temp);
    f_adder fa(arr[0], arr2[0], temp, temp_sum, cout);

    assign sgn = ~outt & shift;
    assign temp2 = sgn & cout | ~sgn & temp;

    always @(posedge clk) begin
        if (shift) begin
            count = count + 1;
        end
        if (count == 8) begin
            outt = 1;
        end
        if (reset | parload) begin
            outt = 0;
            count = 0;
        end
    end

endmodule
```