CS223 Digital design

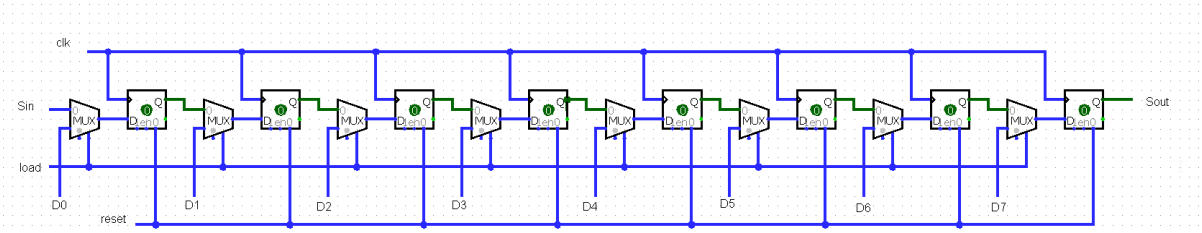
Lab 5 Preliminary Report

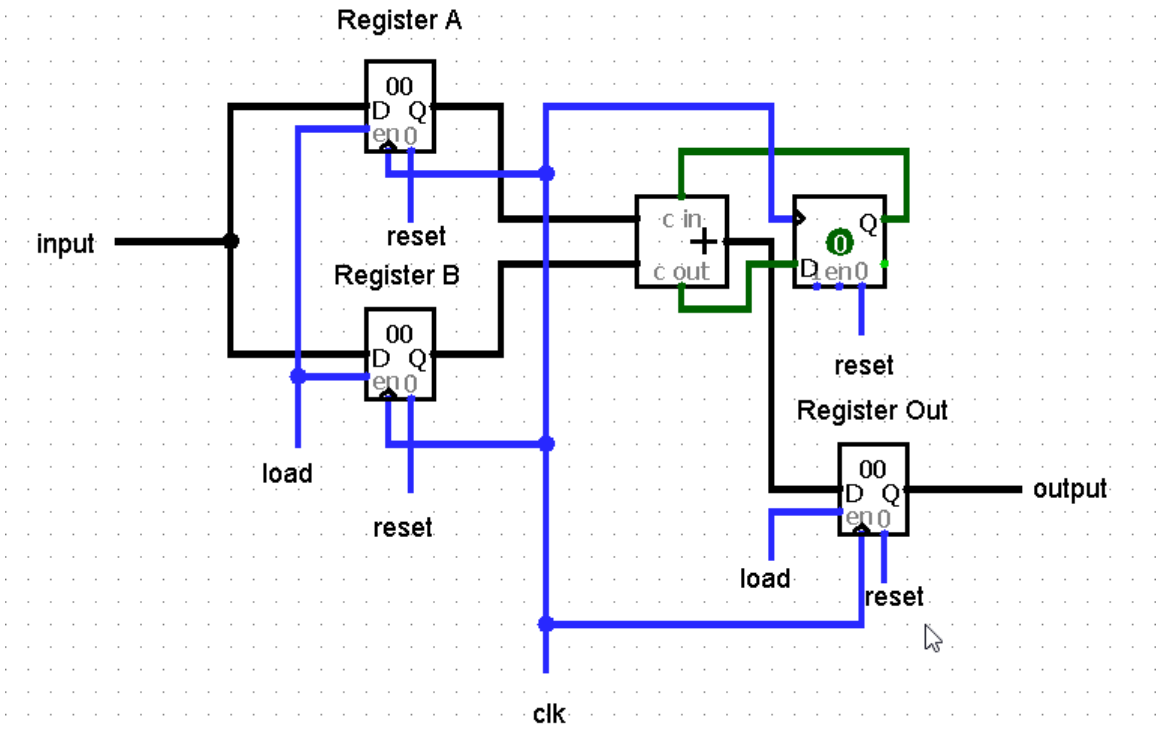
Ege Ipekci

Section 1

21902333

22.05.2023


clk
Sin
load
reset
D0
D1
D2
D3
D4
D5
D6
D7
Sout


Register A
input
reset
Register B
load
reset
c in
c out
reset
Register Out
load
reset
output
clk

```
module multiplexer(
   input logic a, b, select,
   output logic o
   );

      always_comb
      begin
         if( !select )
            o = a;
         else
            o = b;
      end
endmodule

module D_ff(
   input logic D, clk, reset,
   output logic Q
);
   always_ff @( posedge clk )
   begin
   if( reset )
      Q <= 0;
   else
      Q <= D;
   end
endmodule
```

```
module full_adder(
   input logic A, B, Cin,
   output logic Sum, Cout
);

   assign Sum = Cin ^ ( A ^ B );
   assign Cout = ( ( A ^ B ) & Cin ) | ( A & B );
endmodule

module shift_register(
   input logic [7:0] D, I, clk, load, reset,
   output logic [7:0]O2

);

   logic [7:0] O;

   multiplexer mux1( I, D[7],load, O[7] );
   D_ff ff1( O[7], clk, reset, O2[7] );
   multiplexer mux2( O2[7], D[6], load, O[6] );
   D_ff ff2( O[6], clk, reset, O2[6] );
   multiplexer mux3( O2[6], D[5], load, O[5] );
   D_ff ff3( O[5], clk, reset, O2[5] );
   multiplexer mux4( O2[5], D[4], load, O[4] );
   D_ff ff4( O[4], clk, reset, O2[4] );
   multiplexer mux5( O2[4], D[3], load, O[3] );
   D_ff ff5( O[3], clk, reset, O2[3] );
   multiplexer mux6( O2[3], D[2], load, O[2] );
   D_ff ff6( O[2], clk, reset, O2[2] );
```

```
    multiplexer mux7( O2[2], D[1], load, O[1] );
    D_ff ff7( O[1], clk, reset, O2[1] );
    multiplexer mux8( O2[1], D[0], load, O[0] );
    D_ff ff8( O[0], clk, reset, O2[0] );

endmodule



module serial_adder(
    input logic [7:0] A, [7:0] B, shift, load, reset,
    output logic [7:0] Sum
    );
    logic loadx, Q, Sumx, Cout;
    logic [7:0] OA;
    logic [7:0] OB;
    logic [7:0] OSum;
    logic I = 0;
    logic [7:0] Summx = 8'b00000000;

    always_comb
    begin
        if( shift == 1 & load == 0 )  loadx = 0;
        else if( load == 1 & shift == 0 ) loadx = 1;
    end

    full_adder fa( OA[0], OB[0], Q, Sumx, Cout );
    D_ff ff(Cout, clk, reset, Q);
    shift_register regA( A, I, clk, loadx, reset, OA );
```

```
    shift_register regB( B, I, clk, loadx, reset, OB );
    shift_register regOut( Summx, Sum, clk, loadx, reset, OSum );
endmodule

module testbench(

    );
 logic [7:0] D;
 logic I, clk, load, reset;
 logic [7:0]Q;

shift_register uut( D, I, clk, load, reset ,Q );

always begin
clk = 0; #2.5;
clk =1; #2.5;
end
initial begin

 I =0; #5;
 reset = 1;#5;
 reset = 0; #5;
  D = 8'b01010101;
 load = 1; #5;
 load = 0; #5;



end
```

```
module testbench(

   );
    logic [7:0] A, B;
    logic shift, load, reset, clk;
    logic [7:0] Sum;

serial_adder uut( A, B, shift, clk, load, reset , Sum );

always begin
clk = 0; #2.5;
clk =1; #2.5;
end
initial begin

 reset = 1;#5;
 reset = 0; #5;
  A = 8'b01010101;
  B = 8'b01010101;
 load = 1; #5;
 load = 0; #5;



end



endmodule
```