LAB #09

FLIP FLOP AND LATCHES IN VERILOG:

Fall 2023

CSE-304L Computer Organization & Architecture

Submitted by: MUHAMMAD SADEEQ

Registration No.: 21PWCSE2028

Section: C

"On my honor, as a student of the University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work"

Submitted to:

Dr. Bilal Habib

(28 Dec 2023)

Department of Computer systems engineering
University of Engineering and Technology,
Peshawar

Tasks 1

Code:

```
module SRLatch (
  input S,  // Set input
  input R,  // Reset input
  output Q, // Q output
  output ~Q // Inverted Q output
);

  reg Q, ~Q;

  always @(S, R)
    if (R && ~S)       // Reset has higher priority
      begin
        Q <= 0;
        ~Q <= 1;
      end
    else if (~R && S)  // Set
      begin
        Q <= 1;
        ~Q <= 0;
      end
    else               // Hold state
      begin
        Q <= Q;
        ~Q <= ~Q;
      end

endmodule
```

Tasks 2

Code:

```
module SRFlipFlop (
  input S,      // Set input
  input R,      // Reset input
  input CLK,    // Clock input
  output Q,     // Q output
  output ~Q     // Inverted Q output
);

  reg Q, ~Q;
  always @(posedge CLK or negedge S or negedge R)
    if (R)
      begin
        Q <= 0;
        ~Q <= 1;
      end
    else if (S)
      begin
        Q <= 1;
        ~Q <= 0;
      end;

endmodule
```

Tasks 3

Code:

```
module JKFlipFlop (
  input J,      // Set input
  input K,      // Reset input
  input CLK,    // Clock input
  output Q,     // Q output
  output ~Q     // Inverted Q output
);

  reg Q, ~Q;

  always @(posedge CLK)
    if (J && ~K)
      begin
        Q <= 1;
        ~Q <= 0;
      end
    else if (~J && K)
      begin
        Q <= 0;
        ~Q <= 1;
      end
    else if (J && K)
      begin
        Q <= ~Q;
        ~Q <= Q;
      end;
```

```
endmodule
```

Tasks 4

Code:

```
module DFlipFlop (
  input D,      // Data input
  input CLK,    // Clock input
  output Q,     // Q output
  output ~Q     // Inverted Q output
);

  reg Q, ~Q;

  always @(posedge CLK)
    begin
      Q <= D;
      ~Q <= ~D;
    end

endmodule
```

Tasks 5

Code:

```
module TFlipFlop (
  input T,      // Toggle input
  input CLK,    // Clock input
  output Q,     // Q output
  output ~Q     // Inverted Q output
);

  reg D;
  wire Q, ~Q;

  // D flip-flop instantiation
  DFlipFlop uut (
    .D(T),
    .CLK(CLK),
    .Q(Q),
    .~Q(~Q)
  );

endmodule
```