Date .....

Expt. No. .....

Page No. .....

D Flip Flop

```
// D Flip Flop without reset
module DFF(Q, D, clk, rst);
    input D, clk, rst;
    output reg Q;
    always @(posedge clk)
      Q <= D;
    endmodule.

// D flip-flop with asynchronous reset (V2001, V2005)
module Dff (output reg Q, input D, clk, rst);
  always @(posedge clk, negedge clk)
    if (!rst) Q <= 1'b0;        // Same as : if (rst == 0)
    else Q <= D;
  endmodule
// Test Bench
module testbench_Dff ( );
    reg clk, data;
    wire out;
  initial begin
  clk = 0;
  data = 0;
  #1000 $finish;
  end
 always #100 clk = ~clk;
 always #333 data = ~data;
  Dff_d_DFF0 (out, data, clk);
endmodule
```

Expt. No.....

Date.....

Page No.....

T - Flip Flop

```
module Tff (Q, Qb, T, clk);
 input T, clk;
 output reg Q, Qb;
   initial
     begin
       Q = 1'b1;
       Qb = 1'b0;
   end
   always @(clk)
     begin
      if (clk)
        begin
          if (T == 1'b0) begin Q = Q; Qb = Qb; end
          else begin Q = ~Q; Qb = ~Qb; end
        end
     end
  endmodule

module Tff_TB
   wire Q, Qb;
   reg T, clk;
   Tff T1 (Q, Qb, T, clk);
     initial
       begin
         T = 1'b0; clk = 1'b0;
       always
         begin
         #100 clk = ~clk;
         #200 T = ~T;
         end
       initial
       #1000 $finish
   endmodule
```

Teacher's Signature : .....

Date

Expt. No. Page No.

SR Flip Flop

```
module SRff(Q,Qb,S,R,clk);
input S,R,clk;
  output reg Q,Qb;
  always @(posedge clk);
    case({S,R})
     2'b00 : Q<=Q;
     2'b01 : Q<=1'b0;
     2'b10 : Q<=1'b1;
     2'b11 : Q<=1'bx;
    endcase
  endmodule
module SRff_TB;
wire Q,Qb;
reg clk,S,R;
SRff SR1(Q,Qb,S,R,clk);
 Initial
  begin
    S=1'b0; R=1'b0; clk=1'b0;
  end
 always
  #100 clk=~clk;
  Initial
    begin
    #50 S=1'b0; R=1'b0;
    #50 S=1'b0; R=1'b1;
    #50 S=1'b1; R=1'b0;
    #50 S=1'b1; R=1'b1;
    #1000 $finish
    end
endmodule
```

Teacher's Signature :

Date .....

Expt. No. .....

Page No. .....

SISO

```
module SISO(clk, rst, a, q);
 input a;
 input clk, rst;
 output reg q;
 always @(posedge clk, posedge rst)
   begin
     if (rst == 1'b1)
       q <= 1'b0;
     else
       q <= a;
   end
 endmodule
module SISO_TB
 wire q;
 reg a, clk, rst;
 SISO s1(clk, rst, a, q);
  initial
    clk = 1'b1;
  always #100 clk = ~clk;
  initial begin
    a = 1'b0; rst = 1'b1;
    #200 rst = 1'b0;
    #200 a = 1'b1;
    #200 rst = 1'b1;
    #200 rst = 1'b0;
      end
  initial
  #2000 $finish
   endmodule
```

Teacher's Signature : _____

Date .....

Expt. No. .....

Page No. .....

SIPO

```
module SIPO (clk, rst, a, q);
input a;
input clk, rst;
output [3:0] q;
reg [3:0] temp;
always @(posedge clk, posedge rst)
  begin
   if (rst = 1'b1)
    temp <= 4'b0000;
   else
    begin
     temp <= temp << 1'b1;
     temp [0] <= a;
    end
  end
assign q = temp;
endmodule
module SIPO_TB;
wire [3:0] q;
reg a, clk, rst;
SIPO S1(clk, rst, a, q);
initial
  clk = 1'b0;
always #100 clk = ~clk;
  Initial begin
  a = 1'b1; rst = 1'b1;
  #500 rst = 1'b0;
  #200 a = 1'b0;
  #200 a = 1'b1;
```

Teacher's Signature : _____

```
#200  a=1'b0;
#200  a=1'b0;
#200  a=1'b1;
#200  a=1'b0;
 end
#2000 $finish;
endmodule
```

PISO

```
module PISO (clk, rst, a, q);
input clk, rst;
input [3:0] a;
output reg q;
 reg [3:0] temp;
always @(posedge clk, posedge rst)
 begin
    if (rst == 1'b1)
    begin
     q <= 1'b0;
     temp <= a;
     end
    else
    begin
     q <= temp[0];
     temp <= temp >> 1'b1;
     end
    end
endmodule
```

Teacher's Signature : _____

Date .....

Expt. No. ..... Page No. .....

```
module PISO_TB;
wire q;
reg [3:0] a;
reg clk, rst;
PISO P1 (clk, rst, a, q);
initial
  clk = 1'b1;
always #100 clk = ~clk;
 Initial begin
   a = 4'b1101 ; rst = 1'b1;
  #300 rst = 1'b0;
  #200 rst = 1'b1;
  #200 rst = 1'b0;
    end
   initial
  #1000 $finish
  endmodule
```

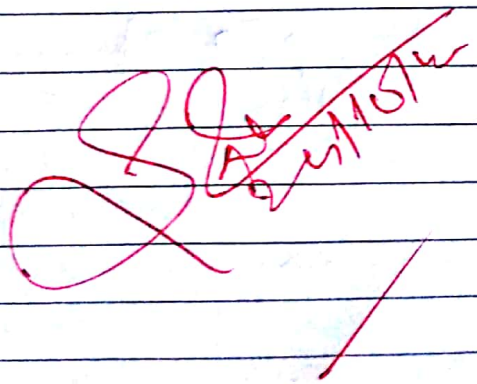
PIPO

```
module PIPO (clk, rst, a, q);
input clk, rst;
input [3:0] a;
output reg [3:0] q;
always @(posedge clk)
   begin
    if (rst = 1'b1)
      q <= 4'b0000;
     else
     q <= a;
```

Teacher's Signature : _____

```
end
endmodule

module PIPO_TB;
wire [3:0]q;
reg clk, rst;
reg [3:0] a;
PIPO P1 (clk, rst, a, q);
 Initial
  clk = 1'b1;
 always #100 clk = ~clk;
 Initial begin
    a = 4'b1101 ; rst = 1'b1;
 #200 rst = 1'b0;
 #200 a = 4'b1000;
 #200 rst = 1'b1;
 #200 rst = 1'b0;
  end
 initial
 #2000 $finish;
endmodule
```



Teacher's Signature : _____

Expt. No. .....

Date .....

Page No. .....

FSM

```
module fsm_mealy_101 (clk, rst, in, out);
 input clk, rst, in;
 output reg out;
 reg [1:0] pre_state, next_state;
 always @ (posedge clk)
  if (rst)    // if in reset condition.
   pre_state <= 2'b00;
  else
   pre_state <= next_state;
  always @ (posedge clk or in)
   begin
   case ({pre_state, in})
    3'b000 : begin
                next_state <= 2'b00;
                out <= 1'b0;
                end
    3'b001 : begin
                next_state <= 2'b01;
                out <= 1'b0;
                end
    3'b010 : begin
                next_state <= 2'b10;
                out <= 1'b0;
                end
    3'b011 : begin
                next_state <= 2'b01;
                out <= 1'b0;
                end
```

Teacher's Signature : _____

Date .....

Expt. No. .....

Page No. .....

```
3'b100  : begin
            next_state <= 2'b00;
            out <= 1'b0;
            end

3'b101  : begin
            next_state <= 2'b01;
            out <= 1'b1;
            end
endcase
end
endmodule
```

Test Bench

```
module fsm_101_tst;
wire out;
reg clk, rst, in;
fsm_melay_101 uut (clk, rst, in, out);
initial
  begin
  clk = 0;
  rst = 1;
  in = 0;
#10 rst = 0;
#10 in = 1;
#10 in = 0;
#10 in = 1;
#10 in = 1;
#10 in = 0;
```

Teacher's Signature : _____