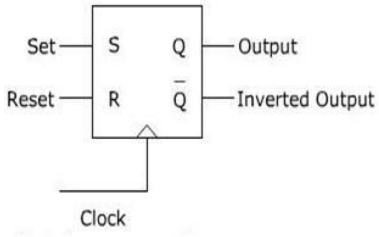
Experiment-8

Design Verilog program for implementing various types of Flip-Flops such as SR, JK and D.

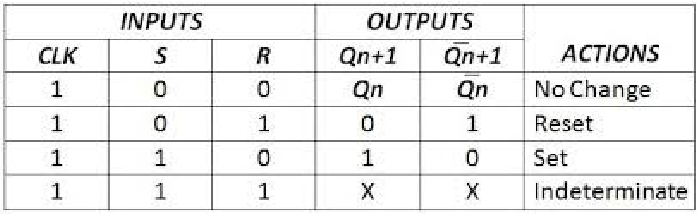
Aim: To develop Verilog code to implement types of flip-flops.

a) SR Flip-flop

Symbol

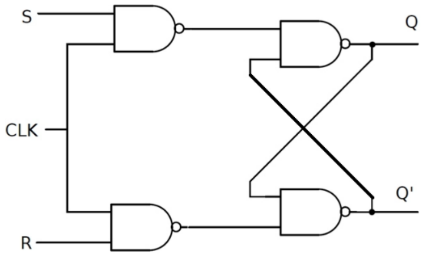


Truth table



INPUTS			OUTPUTS		ACTIONS
CLK	S	R	Qn+1	$\overline{Q}n+1$	
1	0	0	Qn	$\overline{Q}n$	No Change
1	0	1	0	1	Reset
1	1	0	1	0	Set
1	1	1	X	X	Indeterminate

Logic Diagram



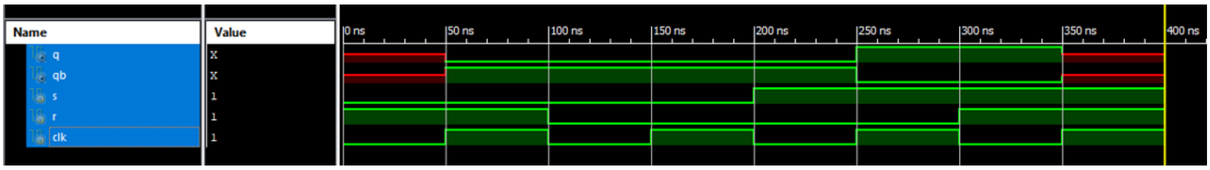
Verilog code

```
module sr_ff (s, r, clk, q, qb);
input s, r, clk;
output reg q, qb;
always@(posedge clk)
begin
  case({s,r})
     2'b00: begin q = q; qb=qb; end
     2'b01: begin q = 1‘b0; qb=1’b1; end
     2'b10: begin q = 1‘b1; qb=1’b0; end
     2'b11: begin q = 1‘bx; qb=1’bx; end
  endcase
 end
endmodule
```

Testbench code

```
module sr_tb;
reg s,r,clk;
wire q,qb;
sr_ff uut(.s(s), .r(r), .clk(.clk), .q(q), .qb(qb));
initial begin
   clk=0;
   forever #50 clk=~clk;
end
initial begin
  s=0; r=1; #100;
  s=0; r=0; #100;
  s=1; r=0; #100;
  s=1; r=1; #100;
  $finish();
end
endmodule
```

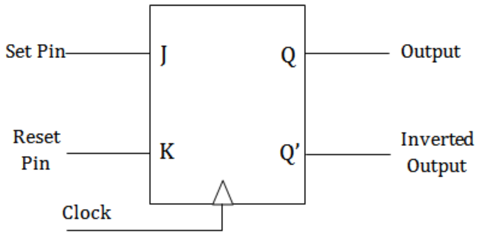
Timing Diagram



Name	Value
q	x
qb	x
s	1
r	1
clk	1

b) JK Flip-flop

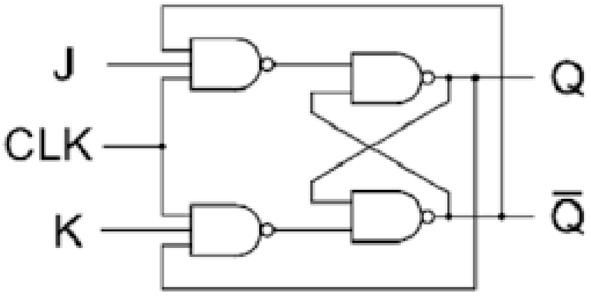
Symbol



Truth table

Clock	J	K	Q_{n+1}	State
0	x	x	Q_n	
1	0	0	Q_n	Hold
1	0	1	0	Reset
1	1	1	1	Set
1	1	1	$\overline{Q}_n$	Toggle

Logic Diagram



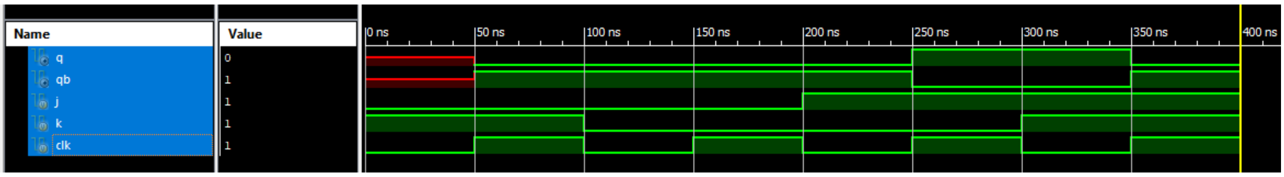
Verilog code

```
module jk_ff (j, k, clk, q, qb);
input j, k, clk;
output reg q, qb;
always @ (posedge clk)
begin
      case ({j,k})
        2'b00 :  begin q = q;  qb=qb; end
        2'b01 :  begin q = 1’b0;  qb=1’b1; end
        2'b10 :  begin q = 1’b1;  qb=1’b0; end
        2'b11 :  begin q = ~q;  qb=~qb; end
      endcase
endmodule
```

Testbench code

```
module jk_tb;
reg j, k, clk;
wire q, qb;
jk_ff uut(.j(j), .k(k), .clk(.clk), .q(q), .qb(qb));
initial begin
  clk=0;
  forever #50 clk=~clk;
end
initial begin
  j=0; k=1; #100;
  j=0; k=0; #100;
  j=1; k=0; #100;
  j=1; k=1; #100;
  $finish();
end
endmodule
```

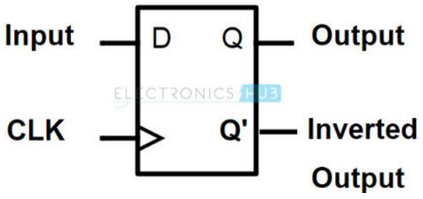
Timing Diagram



Name	Value
q	0
qb	1
j	1
k	1
clk	1

c) D Fllip-flop

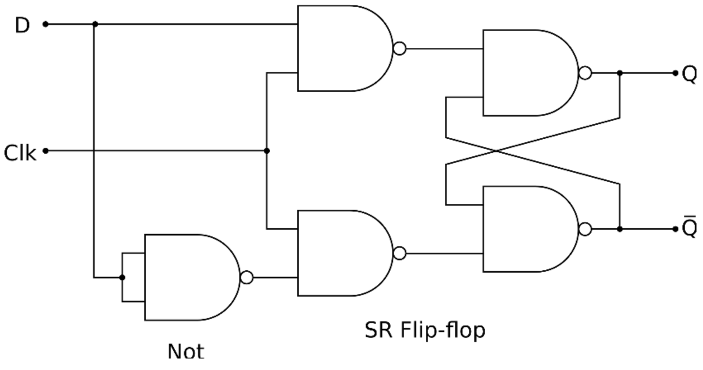
Symbol



Truth table

CLK	*Input D*	*Output Qn+1*
1	0	0
1	1	1
0	X	No Change

Logic Diagram



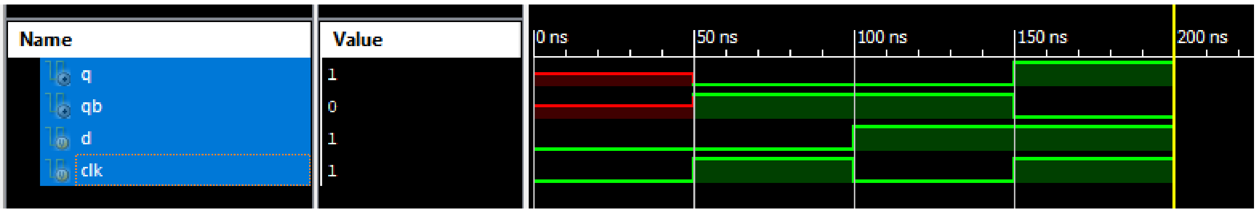
Verilog code

```
module d_ff(clk, d, q, qb);
input clk, d;
output reg q;
always @ (posedge clk)
  begin
    q = d; qb=~q;
  end
endmodule
```

Testbench code

```
module d_tb;
reg d, clk;
wire q, qb;
d_ff UUT(.d(d), .clk(clk), .q(q), .qb(qb));
initial begin
  clk = 0;
  forever #50 clk=~clk;
end
initial begin
  d = 1'b0; #100
  d = 1'b1; #100
  $finish();
end
endmodule
```

Timing Diagram



Name	Value
q	1
qb	0
d	1
clk	1

Result

Verilog programs for implementing SR flip-flop, JK flip-flop and D flip-flop are developed.