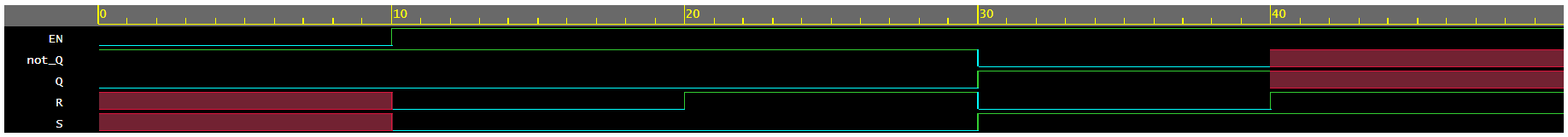
**Experiment – 7**

**Verilog code for designing a SR Latch.**

**design.sv**module sr\_latch (EN, S, R, Q, not\_Q);  
 input EN, S, R;  
 output reg Q, not\_Q;  
  
 always @(EN or S or R) begin  
 if (EN) begin  
 case ({S, R})  
 2'b00:  
 {Q, not\_Q} <= {Q, not\_Q};  
  
 2'b01:  
 {Q, not\_Q} <= 2'b01;  
  
 2'b10:  
 {Q, not\_Q} <= 2'b10;  
  
 2'b11:  
 {Q, not\_Q} <= 2'bxx;  
 endcase  
 end  
 else begin  
 {Q, not\_Q} <= {Q, not\_Q};  
 end  
 end  
endmodule

**testbench.sv**module sr\_latch\_test;  
 reg EN, S, R;  
 wire Q, not\_Q;  
  
 sr\_latch sr\_latch\_dut (EN, S, R, Q, not\_Q);  
  
 initial begin  
 sr\_latch\_dut.Q = 0;  
 sr\_latch\_dut.not\_Q = 1;  
 {EN, S, R} = 3'b0xx; #10;  
 {EN, S, R} = 3'b100; #10;  
 {EN, S, R} = 3'b101; #10;  
 {EN, S, R} = 3'b110; #10;  
 {EN, S, R} = 3'b111; #10;  
 $finish;  
 end  
  
 initial begin  
 $dumpfile("test.vcd");  
 $dumpvars(1, sr\_latch\_test);  
 end  
endmodule

**Output Waveform**

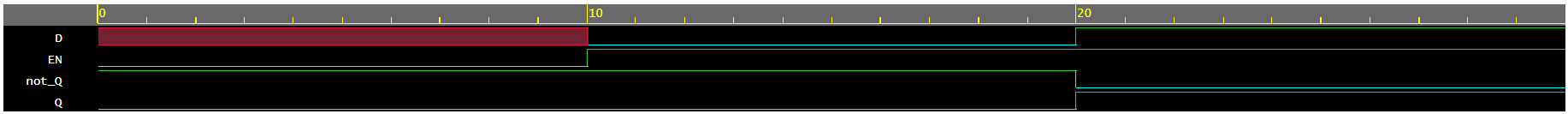


**Verilog code for designing a D Latch.**

**design.sv**module d\_latch(EN, D, Q, not\_Q);  
 input EN, D;  
 output reg Q, not\_Q;  
  
 always @(EN or D) begin  
 if (EN) begin  
 {Q, not\_Q} <= {D, ~D};  
 end  
 else begin  
 {Q, not\_Q} <= {Q, not\_Q};  
 end  
 end   
endmodule

**testbench.sv**module d\_latch\_test;  
 reg EN, D;  
 wire Q, not\_Q;  
  
 d\_latch d\_latch\_dut (EN, D, Q, not\_Q);  
  
 initial begin  
 d\_latch\_dut.Q = 0;  
 d\_latch\_dut.not\_Q = 1;  
 {EN, D} = 2'b0x; #10;  
 {EN, D} = 2'b10; #10;  
 {EN, D} = 2'b11; #10;  
 $finish;  
 end  
  
 initial begin  
 $dumpfile("test.vcd");  
 $dumpvars(1, d\_latch\_test);  
 end  
endmodule

**Output Waveform**

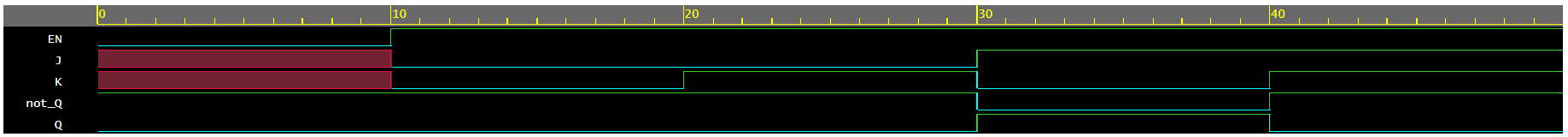


**Verilog code for designing a JK Latch.**

**design.sv**module jk\_latch (EN, J, K, Q, not\_Q);  
 input EN, J, K;  
 output reg Q, not\_Q;  
  
 always @(EN or J or K) begin  
 if (EN) begin  
 case ({J, K})  
 2'b00:  
 {Q, not\_Q} <= {Q, not\_Q};  
  
 2'b01:  
 {Q, not\_Q} <= 2'b01;  
  
 2'b10:  
 {Q, not\_Q} <= 2'b10;  
  
 2'b11:  
 {Q, not\_Q} <= {not\_Q, Q};  
 endcase  
 end  
 else begin  
 {Q, not\_Q} <= {Q, not\_Q};  
 end  
 end  
endmodule

**testbench.sv** module jk\_latch\_test;  
 reg EN, J, K;  
 wire Q, not\_Q;  
  
 jk\_latch jk\_latch\_dut (EN, J, K, Q, not\_Q);  
  
 initial begin  
 jk\_latch\_dut.Q = 0;  
 jk\_latch\_dut.not\_Q = 1;  
 {EN, J, K} = 3'b0xx; #10;  
 {EN, J, K} = 3'b100; #10;  
 {EN, J, K} = 3'b101; #10;  
 {EN, J, K} = 3'b110; #10;  
 {EN, J, K} = 3'b111; #10;  
 $finish;  
 end  
  
 initial begin  
 $dumpfile("test.vcd");  
 $dumpvars(1, jk\_latch\_test);  
 end  
endmodule

**Output Waveform**



**Verilog code for designing a T Latch.**

**design.sv**module t\_latch(EN, T, Q, not\_Q);  
 input EN, T;  
 output reg Q, not\_Q;  
  
 always @(EN or T) begin  
 if (EN) begin  
 if (T) begin  
 {Q, not\_Q} <= {not\_Q, Q};  
 end  
 else begin  
 {Q, not\_Q} <= {Q, not\_Q};  
 end  
 end  
 else begin  
 {Q, not\_Q} <= {Q, not\_Q};  
 end  
 end  
endmodule

**testbench.sv**module t\_latch\_test;  
 reg EN, T;  
 wire Q, not\_Q;  
  
 t\_latch t\_latch\_dut (EN, T, Q, not\_Q);  
  
 initial begin  
 t\_latch\_dut.Q = 0;  
 t\_latch\_dut.not\_Q = 1;  
 {EN, T} = 2'b0x; #10;  
 {EN, T} = 2'b10; #10;  
 {EN, T} = 2'b11; #10;  
 $finish;  
 end  
  
 initial begin  
 $dumpfile("test.vcd");  
 $dumpvars(1, t\_latch\_test);  
 end  
endmodule

**Output Waveform**

