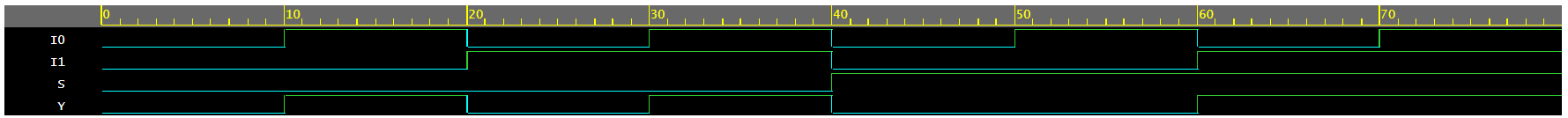
**Experiment – 6-A**

**Verilog code for designing a 2:1 Multiplexer.**

**design.sv** module mux\_2\_to\_1(S, I1, I0, Y);  
 input S, I1, I0;  
 output Y;  
  
 assign Y = ~S ? I0 : I1;  
endmodule

**testbench.sv** module mux\_2\_to\_1\_test();  
 reg S, I1, I0;  
 wire Y;  
  
 mux\_2\_to\_1 mux\_2\_to\_1\_dut(S, I1, I0, Y);  
  
 initial begin  
 S = 0; I1 = 0; I0 = 0; #10;  
 S = 0; I1 = 0; I0 = 1; #10;  
 S = 0; I1 = 1; I0 = 0; #10;  
 S = 0; I1 = 1; I0 = 1; #10;  
 S = 1; I1 = 0; I0 = 0; #10;  
 S = 1; I1 = 0; I0 = 1; #10;  
 S = 1; I1 = 1; I0 = 0; #10;  
 S = 1; I1 = 1; I0 = 1; #10;  
 $finish;  
 end  
  
 initial begin  
 $dumpfile("test.vcd");  
 $dumpvars(1, mux\_2\_to\_1\_test);  
 end  
endmodule

**Output Waveform**

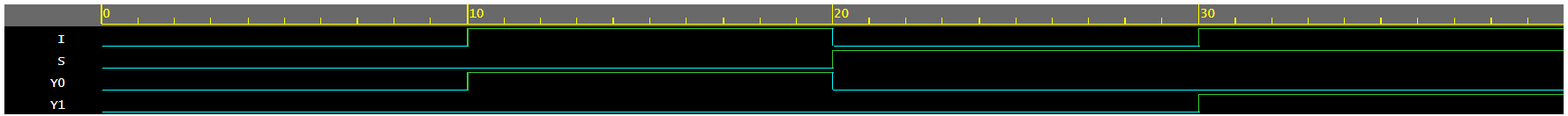


**Verilog code for designing a 2:1 De-Multiplexer.**

**design.sv** module demux\_2\_to\_1(S, I, Y1, Y0);  
 input S, I;  
 output Y1, Y0;  
  
 assign Y1 = S ? I : 0;  
 assign Y0 = ~S ? I : 0;  
endmodule

**testbench.sv** module demux\_2\_to\_1\_test();  
 reg S, I;  
 wire Y1, Y0;  
  
 demux\_2\_to\_1 demux\_2\_to\_1\_dut(S, I, Y1, Y0);  
  
 initial begin  
 S = 0; I = 0; #10;  
 S = 0; I = 1; #10;  
 S = 1; I = 0; #10;  
 S = 1; I = 1; #10;  
 $finish;  
 end  
  
 initial begin  
 $dumpfile("test.vcd");  
 $dumpvars(1, demux\_2\_to\_1\_test);  
 end  
endmodule

**Output Waveform**

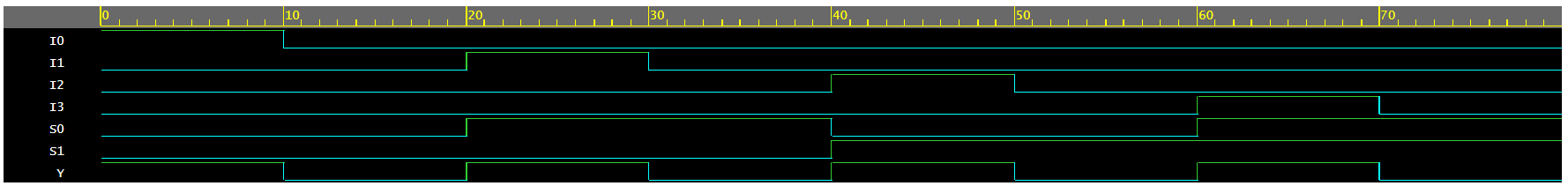


**Verilog code for designing a 4:1 Multiplexer.**

**design.sv** module mux\_4\_to\_1(S1, S0, I3, I2, I1, I0, Y);  
 input S1, S0, I3, I2, I1, I0;  
 output Y;  
  
 assign Y = ~S1 ? (~S0 ? I0 : I1) : (~S0 ? I2 : I3);  
endmodule

**testbench.sv** module mux\_4\_to\_1\_test();  
 reg S1, S0, I3, I2, I1, I0;  
 wire Y;  
  
 mux\_4\_to\_1 mux\_4\_to\_1\_dut(S1, S0, I3, I2, I1, I0, Y);  
  
 initial begin  
 S1 = 0; S0 = 0; I3 = 0; I2 = 0; I1 = 0; I0 = 1; #10;  
 S1 = 0; S0 = 0; I3 = 0; I2 = 0; I1 = 0; I0 = 0; #10;  
  
 S1 = 0; S0 = 1; I3 = 0; I2 = 0; I1 = 1; I0 = 0; #10;  
 S1 = 0; S0 = 1; I3 = 0; I2 = 0; I1 = 0; I0 = 0; #10;  
  
 S1 = 1; S0 = 0; I3 = 0; I2 = 1; I1 = 0; I0 = 0; #10;  
 S1 = 1; S0 = 0; I3 = 0; I2 = 0; I1 = 0; I0 = 0; #10;  
  
 S1 = 1; S0 = 1; I3 = 1; I2 = 0; I1 = 0; I0 = 0; #10;  
 S1 = 1; S0 = 1; I3 = 0; I2 = 0; I1 = 0; I0 = 0; #10;  
 $finish;  
 end  
  
 initial begin  
 $dumpfile("test.vcd");  
 $dumpvars(1, mux\_4\_to\_1\_test);  
 end  
endmodule

**Output Waveform**



**Verilog code for designing a 4:1 De-Multiplexer.**

**design.sv** module demux\_4\_to\_1(S1, S0, I, Y3, Y2, Y1, Y0);  
 input S1, S0, I;  
 output Y3, Y2, Y1, Y0;  
   
 assign Y0 = ~S1 & ~S0 ? I : 0;  
 assign Y1 = ~S1 & S0 ? I : 0;  
 assign Y2 = S1 & ~S0 ? I : 0;  
 assign Y3 = S1 & S0 ? I : 0;  
endmodule

**testbench.sv** module demux\_4\_to\_1\_test();  
 reg S1, S0, I;  
 wire Y3, Y2, Y1, Y0;  
  
 demux\_4\_to\_1 demux\_4\_to\_1\_dut(S1, S0, I, Y3, Y2, Y1, Y0);  
  
 initial begin  
 S1 = 0; S0 = 0; I = 0; #10;  
 S1 = 0; S0 = 0; I = 1; #10;  
 S1 = 0; S0 = 1; I = 0; #10;  
 S1 = 0; S0 = 1; I = 1; #10;  
 S1 = 1; S0 = 0; I = 0; #10;  
 S1 = 1; S0 = 0; I = 1; #10;  
 S1 = 1; S0 = 1; I = 0; #10;  
 S1 = 1; S0 = 1; I = 1; #10;  
 $finish;  
 end  
  
 initial begin  
 $dumpfile("test.vcd");  
 $dumpvars(1, demux\_4\_to\_1\_test);  
 end  
endmodule

**Output Waveform**

