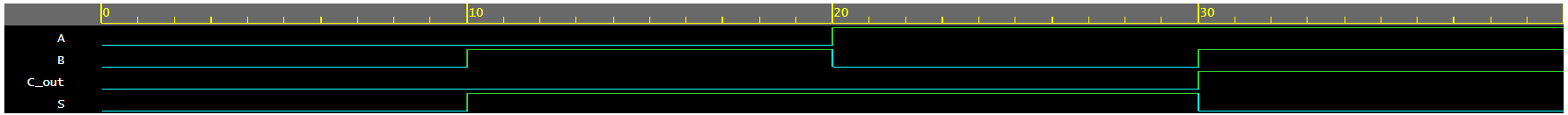
**Experiment – 3**

**Verilog code for designing a Half Adder.**

**design.sv** module half\_adder(A, B, S, C\_out);  
 input A, B;  
 output S, C\_out;  
  
 xor(S, A, B);  
 and(C\_out, A, B);  
endmodule

**testbench.sv** module half\_adder\_test();  
 reg A, B;  
 wire S, C\_out;  
  
 half\_adder half\_adder\_dut(A, B, S, C\_out);  
   
 initial begin  
 A = 0; B = 0; #10;  
 A = 0; B = 1; #10;  
 A = 1; B = 0; #10;  
 A = 1; B = 1; #10;  
 $finish;  
 end  
  
 initial begin  
 $dumpfile("test.vcd");  
 $dumpvars(1, half\_adder\_test);  
 end  
endmodule

**Output Waveform**

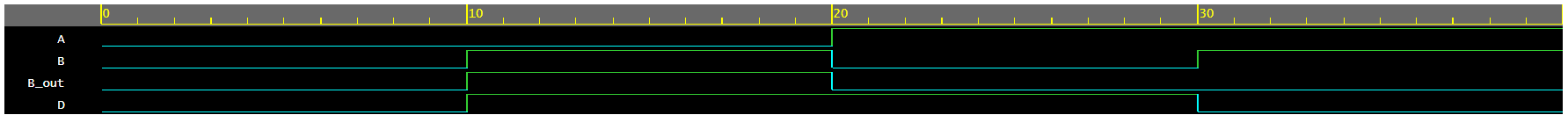


**Verilog code for designing a Half Subtracter.**

**design.sv** module half\_subtracter(A, B, D, B\_out);  
 input A, B;  
 output D, B\_out;  
 wire a\_not;  
  
 xor(D, A, B);  
 not(a\_not, A);  
 and(B\_out, a\_not, B);  
endmodule

**testbench.sv** module half\_subtracter\_test();  
 reg A, B;  
 wire D, B\_out;  
  
 half\_subtracter half\_subtracter\_dut(A, B, D, B\_out);  
   
 initial begin  
 A = 0; B = 0; #10;  
 A = 0; B = 1; #10;  
 A = 1; B = 0; #10;  
 A = 1; B = 1; #10;  
 $finish;  
 end  
  
 initial begin  
 $dumpfile("test.vcd");  
 $dumpvars(1, half\_subtracter\_test);  
 end  
endmodule

**Output Waveform**

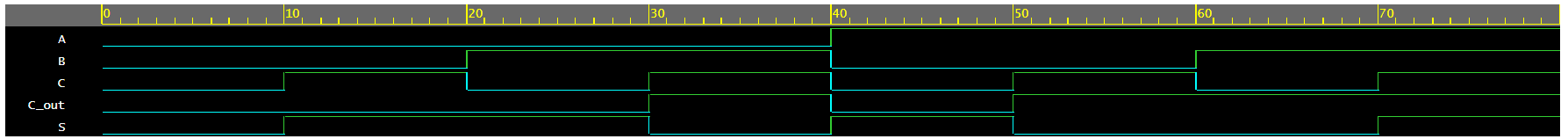


**Verilog code for designing a Full Adder.**

**design.sv**  module full\_adder(A, B, C, S, C\_out);  
 input A, B, C;  
 output S, C\_out;  
 wire a\_xor\_b, or1, or2;  
  
 xor(a\_xor\_b, A, B);  
 xor(S, a\_xor\_b, C);  
 and(or1, a\_xor\_b, C);  
 and(or2, A, B);  
 or(C\_out, or1, or2);  
endmodule

**testbench.sv**  module full\_adder\_test();  
 reg A, B, C;  
 wire S, C\_out;  
  
 full\_adder full\_adder\_dut(A, B, C, S, C\_out);  
   
 initial begin  
 A = 0; B = 0; C = 0; #10;  
 A = 0; B = 0; C = 1; #10;  
 A = 0; B = 1; C = 0; #10;  
 A = 0; B = 1; C = 1; #10;  
 A = 1; B = 0; C = 0; #10;  
 A = 1; B = 0; C = 1; #10;  
 A = 1; B = 1; C = 0; #10;  
 A = 1; B = 1; C = 1; #10;  
 $finish;  
 end  
  
 initial begin  
 $dumpfile("test.vcd");  
 $dumpvars(1, full\_adder\_test);  
 end  
endmodule

**Output Waveform**



**Verilog code for designing a Full Subtracter.**

**design.sv**  module full\_subtractor(A, B, C, D, B\_out);  
 input A, B, C;  
 output D, B\_out;  
 wire a\_xor\_b,not\_a\_xor\_b, not\_a, or1, or2;  
  
 xor(a\_xor\_b, A, B);  
 xor(D, a\_xor\_b, C);  
 not(not\_a\_xor\_b, a\_xor\_b);  
 and(or1, not\_a\_xor\_b, C);  
 not(not\_a, A);  
 and(or2, not\_a, B);  
 or(B\_out, or1, or2);  
endmodule

**testbench.sv**  module full\_subtracter\_test();  
 reg A, B, C;  
 wire D, B\_out;  
  
 full\_subtracter full\_subtracter\_dut(A, B, C, D, B\_out);  
   
 initial begin  
 A = 0; B = 0; C = 0; #10;  
 A = 0; B = 0; C = 1; #10;  
 A = 0; B = 1; C = 0; #10;  
 A = 0; B = 1; C = 1; #10;  
 A = 1; B = 0; C = 0; #10;  
 A = 1; B = 0; C = 1; #10;  
 A = 1; B = 1; C = 0; #10;  
 A = 1; B = 1; C = 1; #10;  
 $finish;  
 end  
  
 initial begin  
 $dumpfile("test.vcd");  
 $dumpvars(1, full\_subtracter\_test);  
 end  
endmodule

**Output Waveform**

