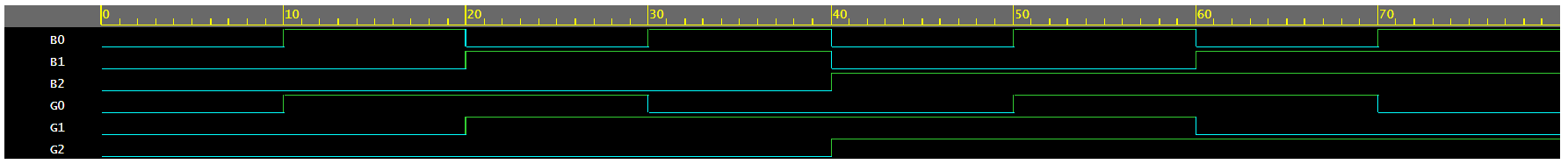
**Experiment – 5**

**Verilog code for designing a converter from 3-bit Binary to Gray code.**

**design.sv**module binary\_to\_gray(B2, B1, B0, G2, G1, G0);  
 input B2, B1, B0;  
 output G2, G1, G0;  
   
 xor(G0, B0, B1);  
 xor(G1, B1, B2);  
 buf(G2, B2);  
endmodule

**testbench.sv**module binary\_to\_gray\_test();  
 reg B2, B1, B0;  
 wire G2, G1, G0;  
  
 binary\_to\_gray binary\_to\_gray\_dut(B2, B1, B0, G2, G1, G0);  
   
 initial begin  
 B2 = 0; B1 = 0; B0 = 0; #10;  
 B2 = 0; B1 = 0; B0 = 1; #10;  
 B2 = 0; B1 = 1; B0 = 0; #10;  
 B2 = 0; B1 = 1; B0 = 1; #10;  
 B2 = 1; B1 = 0; B0 = 0; #10;  
 B2 = 1; B1 = 0; B0 = 1; #10;  
 B2 = 1; B1 = 1; B0 = 0; #10;  
 B2 = 1; B1 = 1; B0 = 1; #10;  
 $finish;  
 end  
  
 initial begin  
 $dumpfile("test.vcd");  
 $dumpvars(1, binary\_to\_gray\_test);  
 end  
endmodule

**Output Waveform**



**Verilog code for designing a converter from 3-bit Gray code to Binary.**

**design.sv**module gray\_to\_binary(G2, G1, G0, B2, B1, B0);  
 input G2, G1, G0;  
 output B2, B1, B0;  
 wire g0\_xor\_g1;  
   
 xor(g0\_xor\_g1, G0, G1);  
 xor(B0, g0\_xor\_g1, G2);  
 xor(B1, G1, G2);  
 buf(B2, G2);  
endmodule

**testbench.sv**module gray\_to\_binary\_test();  
 reg G2, G1, G0;  
 wire B2, B1, B0;  
  
 gray\_to\_binary gray\_to\_binary\_dut(G2, G1, G0, B2, B1, B0);  
  
 initial begin  
 G2 = 0; G1 = 0; G0 = 0; #10;  
 G2 = 0; G1 = 0; G0 = 1; #10;  
 G2 = 0; G1 = 1; G0 = 0; #10;  
 G2 = 0; G1 = 1; G0 = 1; #10;  
 G2 = 1; G1 = 0; G0 = 0; #10;  
 G2 = 1; G1 = 0; G0 = 1; #10;  
 G2 = 1; G1 = 1; G0 = 0; #10;  
 G2 = 1; G1 = 1; G0 = 1; #10;  
 $finish;  
 end  
  
 initial begin  
 $dumpfile("test.vcd");  
 $dumpvars(1, gray\_to\_binary\_test);  
 end  
endmodule

**Output Waveform**

