Lab 1

Question 4

Gate Modelling Code

```
module decoder_2x4_gates_q4 (D, A, B, enable);
  output [3:0] D;
  input A, B;
  input enable;
  wire A_not, B_not, enable_not;

  not G1 (A_not, A),
      G2 (B_not, B),
      G3 (enable_not, enable);

  nand G4 (D[0], A_not, B_not, enable_not),
       G5 (D[1], A_not, B, enable_not),
       G6 (D[2], A, B_not, enable_not),
       G7 (D[3], A, B, enable_not);

endmodule
```

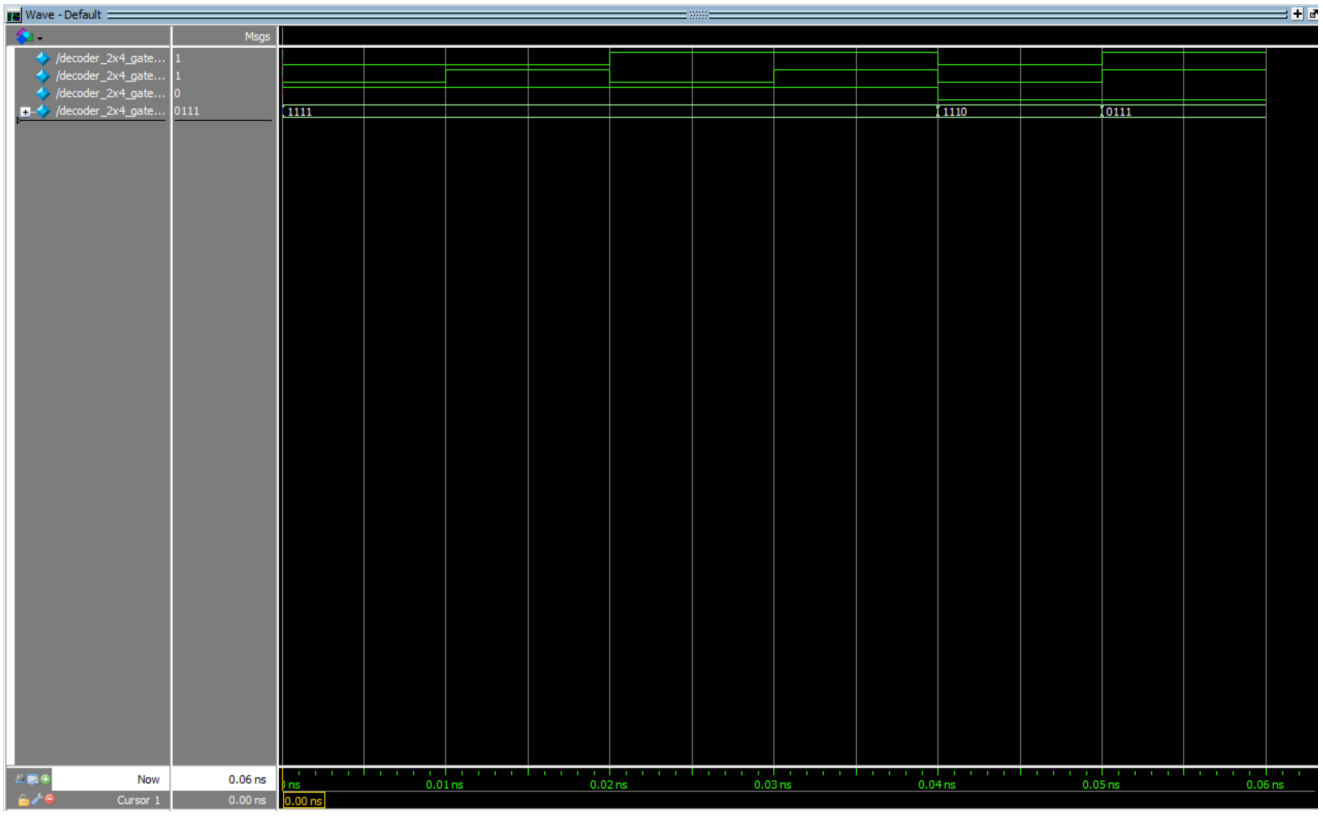
Dataflow Modelling Code

```
module decoder_2x4_dataflow_Q4 (D, A, B, enable);
  output [3:0] D;
  input A, B, enable;

  assign D[0] = enable & ~A & ~B;
  assign D[1] = enable & ~A &  B;
  assign D[2] = enable &  A & ~B;
  assign D[3] = enable &  A &  B;
endmodule
```

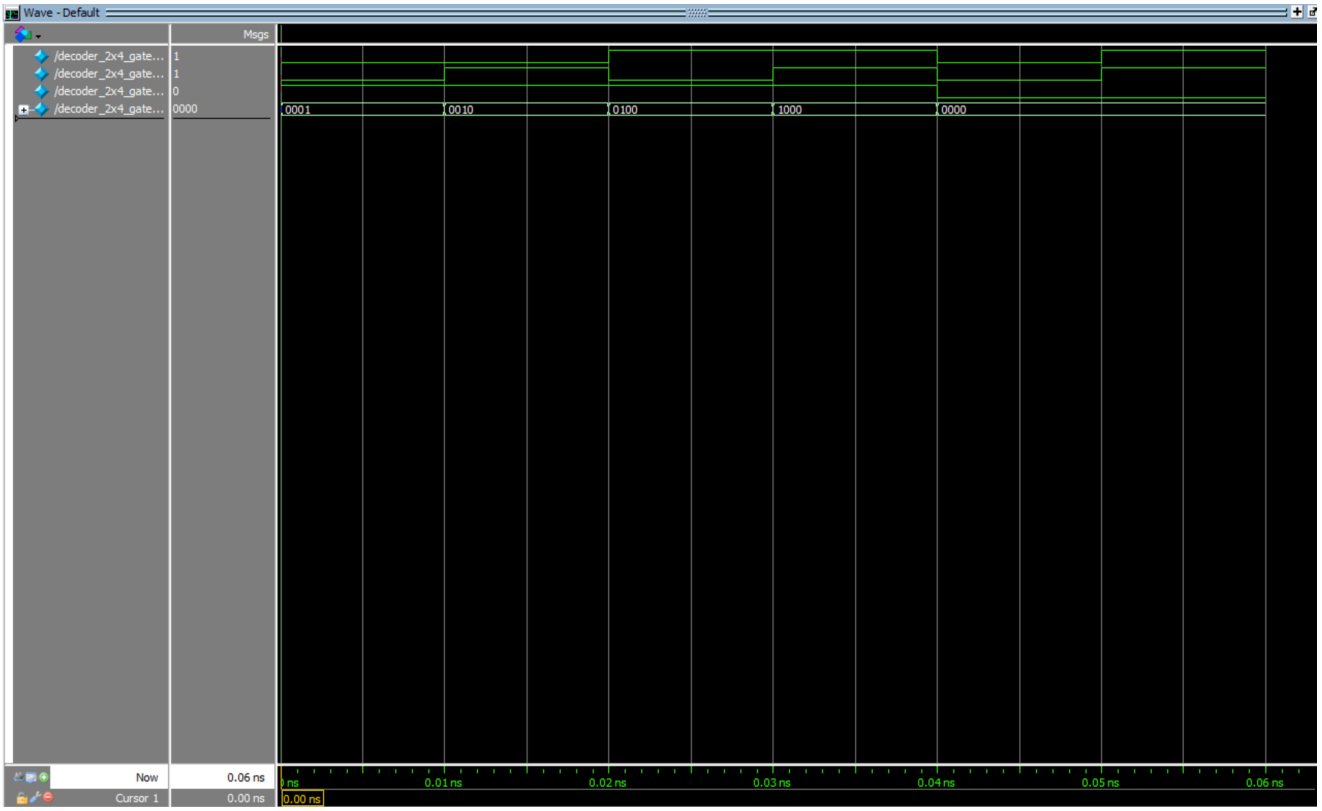
Testbench

```
module decoder_2x4_gates_q4_tb;
  reg A, B, enable;
  wire [0:3] D;

  // Instantiate DUT
  //decoder_2x4_gates_q4 uut (D, A, B, enable);
  decoder_2x4_dataflow_Q4 uut (D, A, B, enable);

  initial begin
    $monitor("Time=%0t | Enable=%b A=%b B=%b | D=%b", $time, enable, A, B,
D);

    // Test all combinations
    enable = 1; A = 0; B = 0; #10;
    enable = 1; A = 0; B = 1; #10;
    enable = 1; A = 1; B = 0; #10;
    enable = 1; A = 1; B = 1; #10;

    // Disable case
    enable = 0; A = 0; B = 0; #10;
    enable = 0; A = 1; B = 1; #10;

    $stop;
  end
endmodule
```

Gate Modelling Output



Dataflow Modelling Output



Question 5

Dataflow Modelling Code

```
module Q5_adder4_dataflow (
  input  [3:0] A, B,
  input        Cin,
  output [3:0] Sum,
  output       Cout
);
  assign {Cout, Sum} = A + B + Cin;
endmodule
```

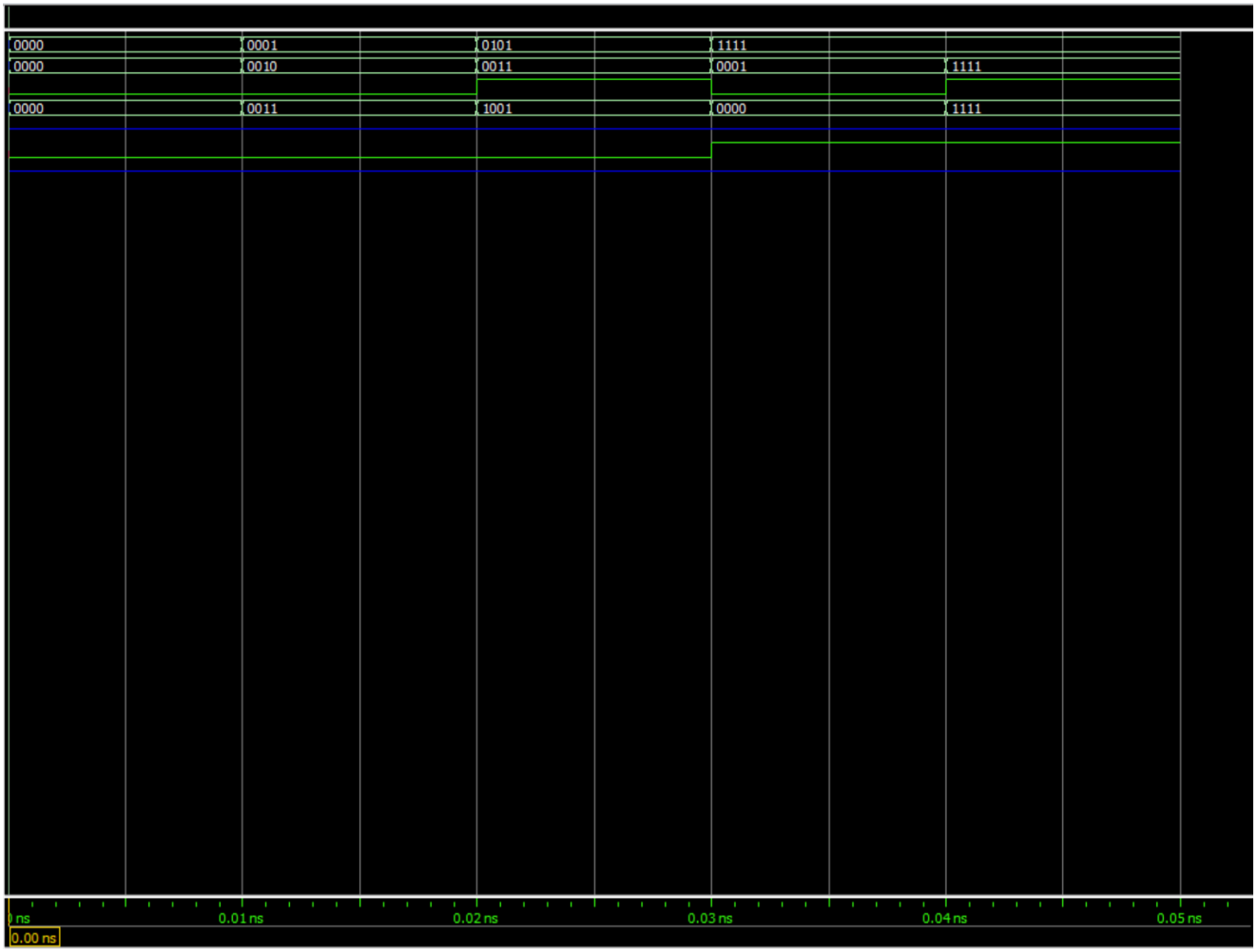
Behavioral Modelling Code

```
module Q5_adder4_behavioral (
  input  [3:0] A, B,
  input        Cin,
  output reg [3:0] Sum,
  output reg       Cout
);
  always @ (A, B, Cin) begin
    {Cout, Sum} = A + B + Cin;
  end
endmodule
```

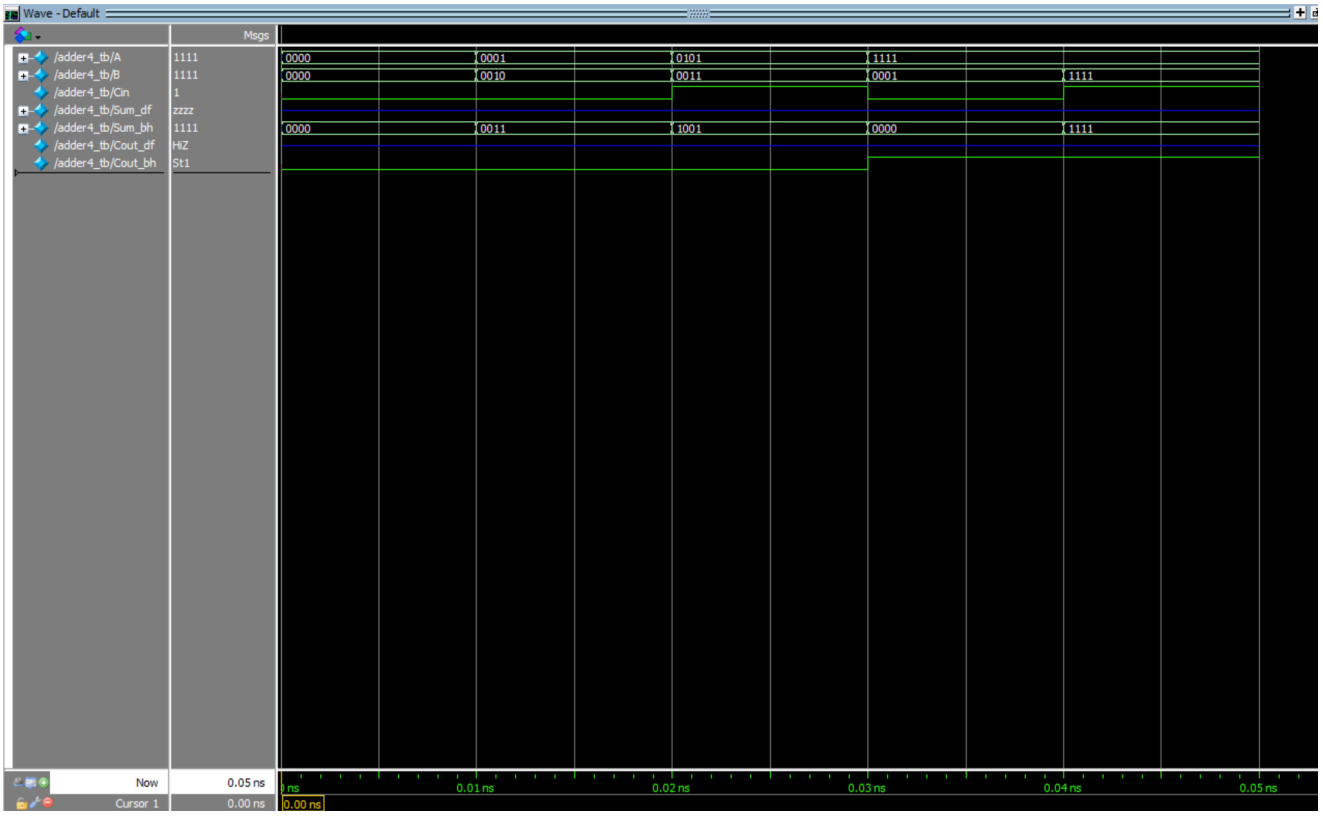
Testbench

```
module adder4_tb;
  reg  [3:0] A, B;
  reg        Cin;
  wire [3:0] Sum_df, Sum_bh;
  wire       Cout_df, Cout_bh;

  // Instantiate both versions
  //Q5_adder4_dataflow   DF (A, B, Cin, Sum_df, Cout_df);
  Q5_adder4_behavioral BH (A, B, Cin, Sum_bh, Cout_bh);

  initial begin
    $monitor("Time=%0t | A=%b B=%b Cin=%b | DF: Sum=%b Cout=%b | BH: Sum=%b
Cout=%b",
             $time, A, B, Cin, Sum_df, Cout_df, Sum_bh, Cout_bh);

    // Test cases
    A=4'b0000; B=4'b0000; Cin=0; #10;
    A=4'b0001; B=4'b0010; Cin=0; #10;
    A=4'b0101; B=4'b0011; Cin=1; #10;
    A=4'b1111; B=4'b0001; Cin=0; #10;
    A=4'b1111; B=4'b1111; Cin=1; #10;

    $stop;
  end
endmodule
```

Dataflow Modelling Output



Behavioral Modelling Output



Question 6

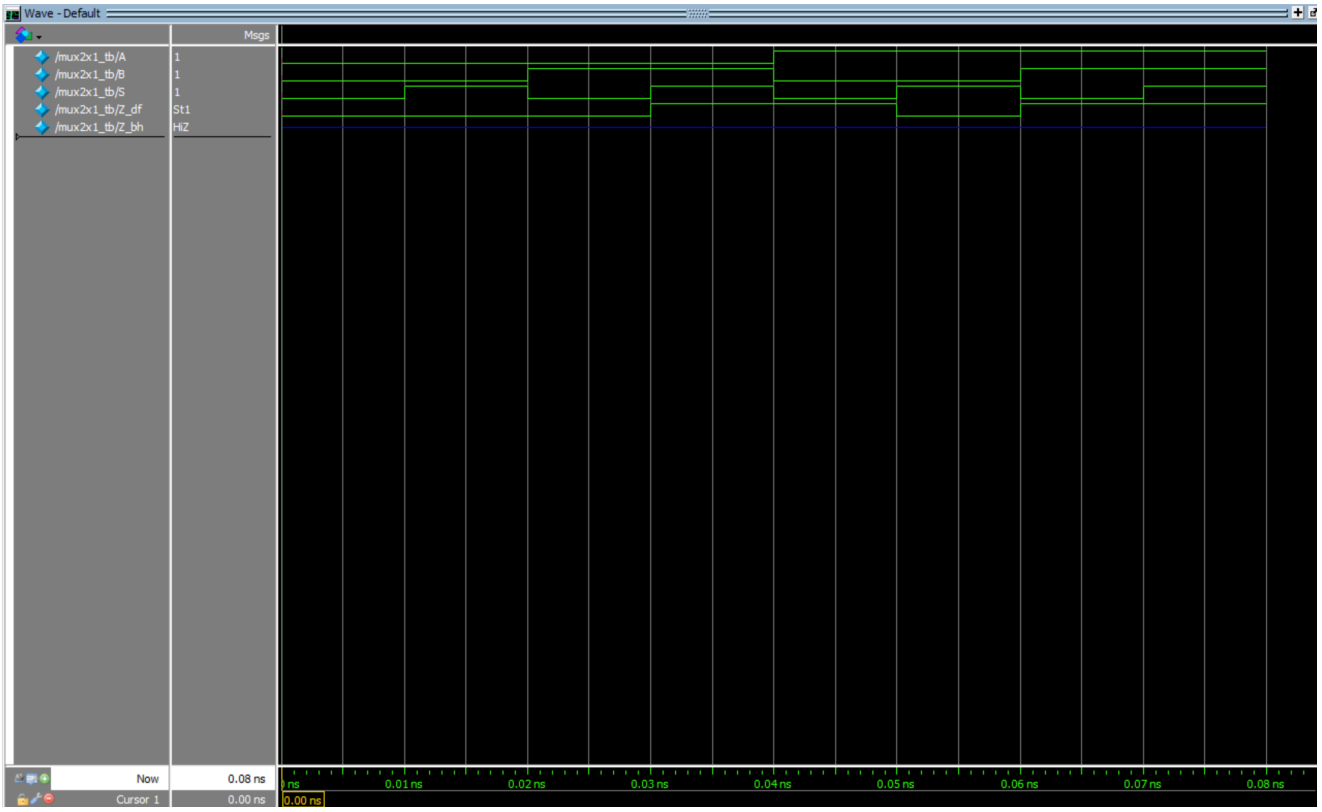
Dataflow Modelling Code

```
module Q6_mux2x1_dataflow (
  input A, B, S,
  output Z
);
  assign Z = (~S & A) | (S & B);
endmodule
## Behavioral Modelling Code
module Q6_mux2x1_behavioral (
  input A, B, S,
  output reg Z
);
  always @(*) begin
    if (S == 0)
      Z = A;
    else
      Z = B;
  end
endmodule
```

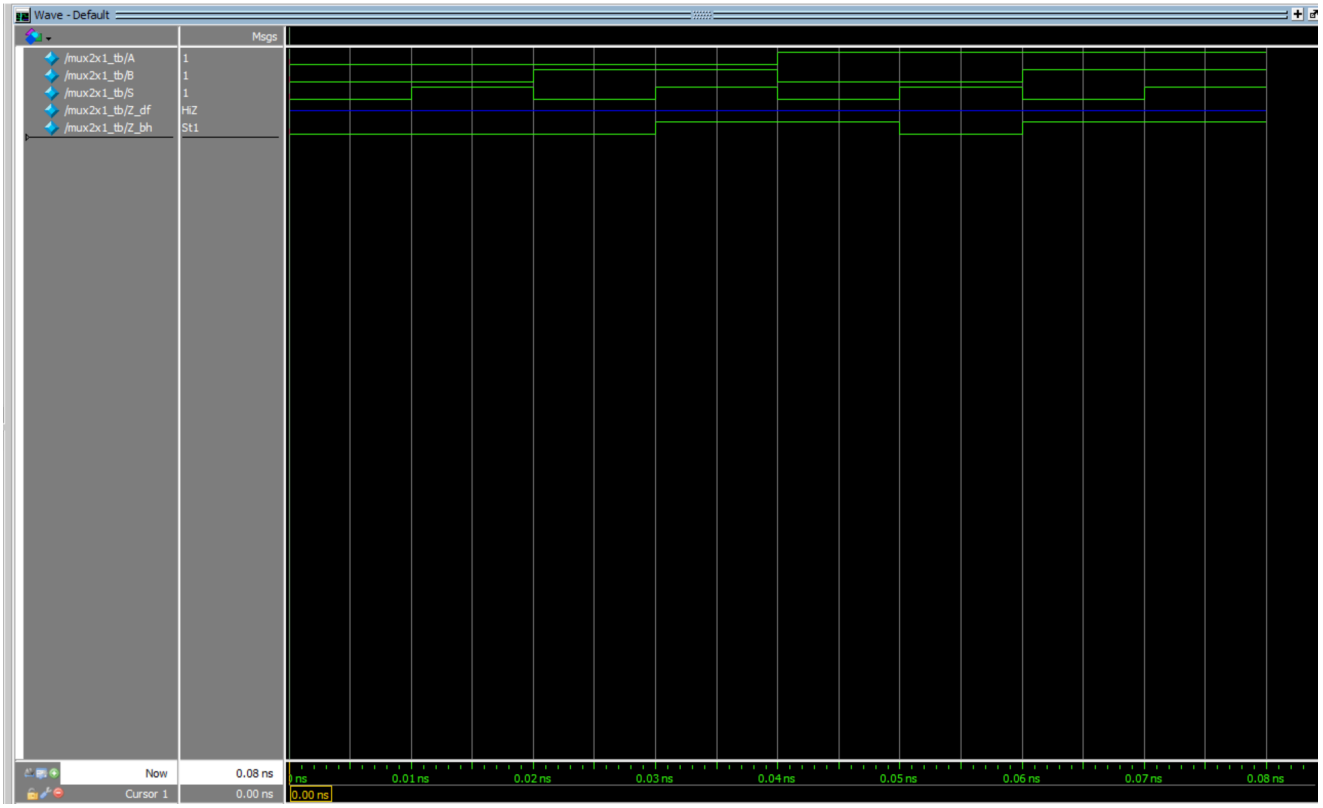
Testbench

```
module mux2x1_tb;
  reg A, B, S;
  wire Z_df, Z_bh;

  // Instantiate both versions
  //Q6_mux2x1_dataflow   DF (A, B, S, Z_df);
  Q6_mux2x1_behavioral BH (A, B, S, Z_bh);

  initial begin
    $monitor("Time=%0t | A=%b B=%b S=%b | DF=%b BH=%b",
             $time, A, B, S, Z_df, Z_bh);

    // Test all possible input combinations (2^3 = 8)
    A=0; B=0; S=0; #10;
    A=0; B=0; S=1; #10;
    A=0; B=1; S=0; #10;
    A=0; B=1; S=1; #10;
    A=1; B=0; S=0; #10;
    A=1; B=0; S=1; #10;
    A=1; B=1; S=0; #10;
    A=1; B=1; S=1; #10;

    $stop;
  end
endmodule
```

Dataflow Modelling Output



Behavioral Modelling Output



Question 7

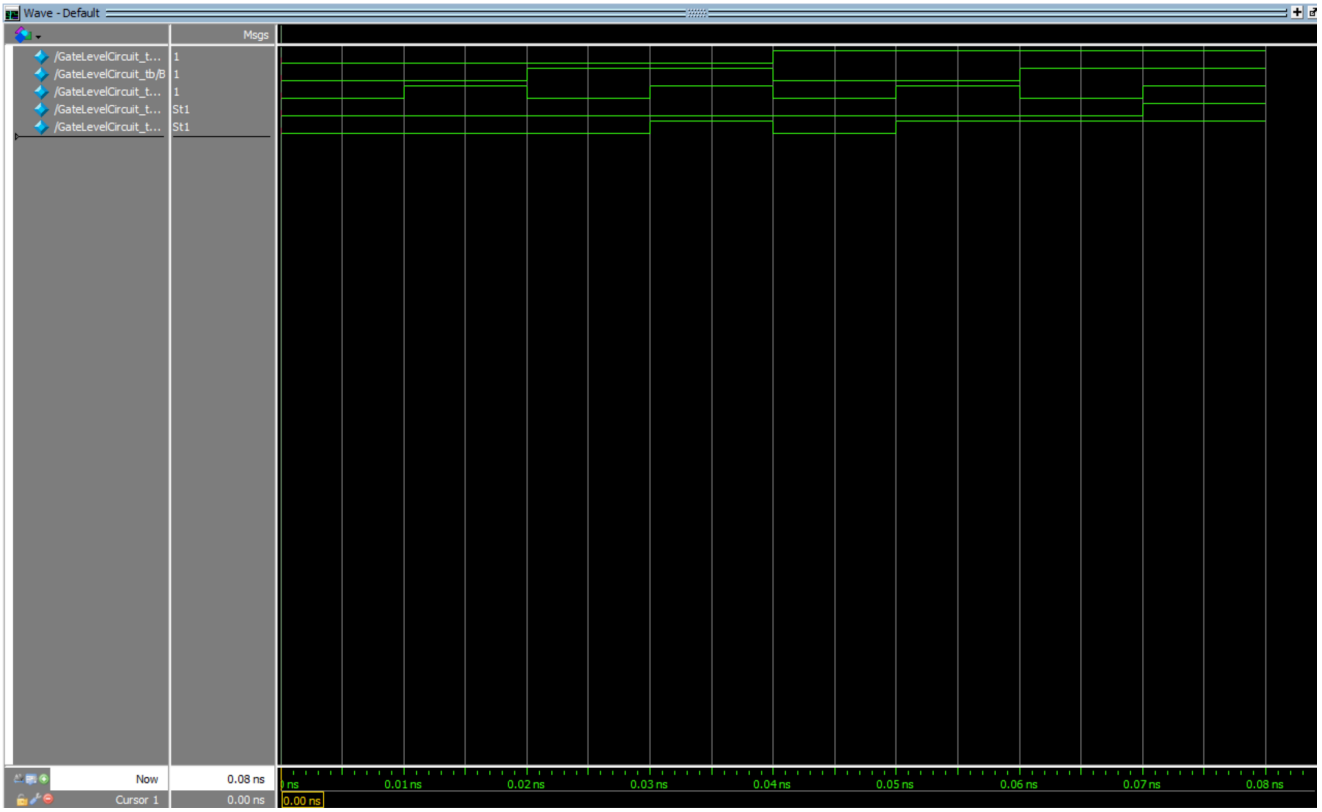
Code

```
module Q7_GateLevelCircuit (A, B, C, F1, F2);
  input A, B, C;
  output F1, F2;
  wire T1, T2, T3, F2_bar;
  wire AB, AC, BC;

  // T1 = A + B + C
  or  (T1, A, B, C);

  // T2 = A * B * C
  and (T2, A, B, C);

  // F2_bar = ~T1
  not (F2_bar, T1);

  // T3 = F2_bar * C
  and (T3, F2_bar, C);

  // F1 = T2 + T3
  or  (F1, T2, T3);

  // F2 = AB + AC + BC
  and (AB, A, B);
  and (AC, A, C);
  and (BC, B, C);
  or  (F2, AB, AC, BC);

endmodule
```

Testbench

```
module GateLevelCircuit_tb;
  reg A, B, C;
  wire F1, F2;

  // Instantiate the design
  Q7_GateLevelCircuit uut (A, B, C, F1, F2);

  initial begin
    $display("A B C | F1 F2");
    $monitor("%b %b %b | %b  %b", A, B, C, F1, F2);

    // Test all 8 combinations
    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
endmodule
```

Output


Wave - Default
Msgs
/GateLevelCircuit_t...
/GateLevelCircuit_tb/B
/GateLevelCircuit_t...
/GateLevelCircuit_t...
/GateLevelCircuit_t...
1
1
1
St1
St1
Now
0.08 ns
Cursor 1
0.00 ns
0.00 ns
0.01 ns
0.02 ns
0.03 ns
0.04 ns
0.05 ns
0.06 ns
0.07 ns
0.08 ns