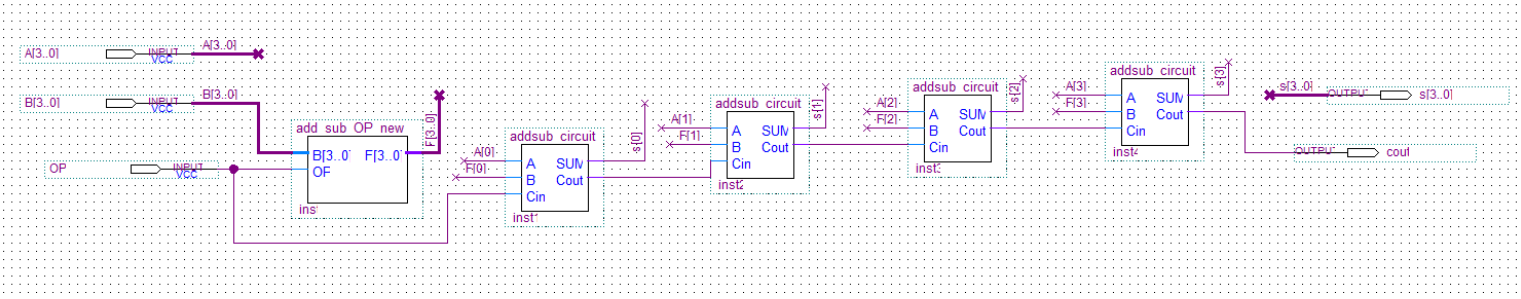
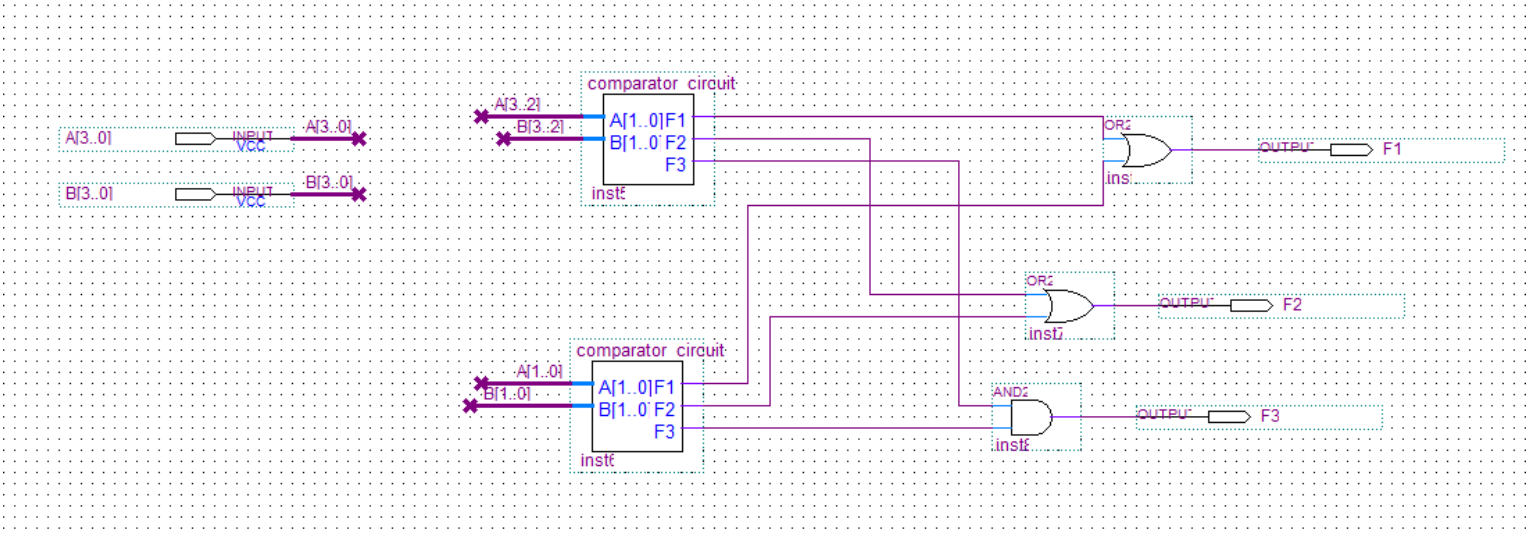
LAB 3 PRE-WORK

2.2.5 ARITHMETIC UNIT

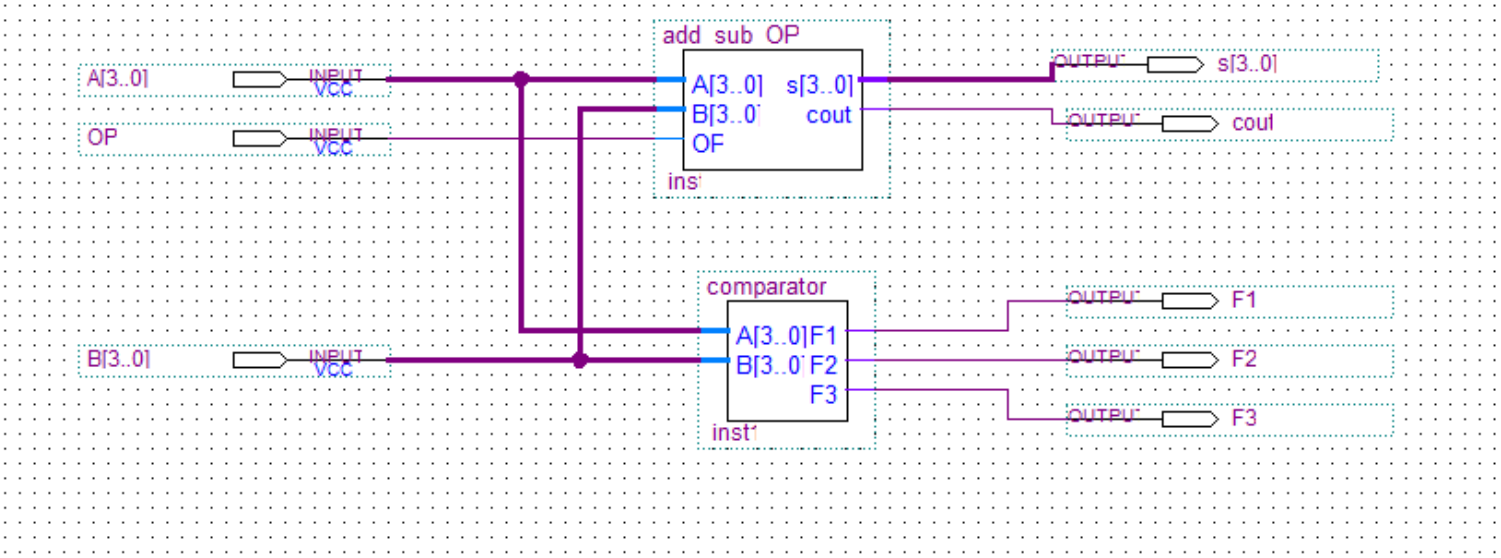
Adder/Subtractor:



Comparator:



Arithmetic Unit:



Verilog code for adder/subtractor:

```
module add_sub_OP (A, B, OP, s, cout);
  parameter size = 4 ;

  input [size-1:0] A, B;
  input OP;
  output [size-1:0] s;
  output cout;
  wire [size+2:0] w;

  genvar i;
  generate
  for (i=0 ; i<size; i=i+1) begin: add_sub_OP
    if (i==0) begin
       OP_signal_new U2(B, OP, w[i]);
       add_sub_circuit U1(A[i], w[i], OP, s[i], w[i+4]);
     end

    else if(i==size-1) begin
      OP_signal_new U2(B, OP, w[i]);
      add_sub_circuit U1(A[i], w[i], w[2*i], s[i], cout);
    end

    else begin
       OP_signal_new U2(B, OP, w[i]);
       add_sub_circuit U1(A[i], w[i] , w[i+3], s[i], w[i+4]);
    end

end
endgenerate
endmodule

```

Verilog code for comparator:

```
module comparator(A, B, F1, F2, F3);
  parameter size = 4;

  input [size-1:0] A, B;
  output F1, F2, F3;
  wire w1, w2, w3, w4, w5, w6;

  or (F1, w1, w4);
  or (F2, w2, w5);
  and (F3, w3, w6);

  genvar i;
  generate
    for (i=0 ; i<size ; i=i+1) begin: comparator
      if (i==0 || i==1)
        comparator_circuit U1(A[i], B[i], w4, w5, w6);
    else
        comparator_circuit U1(A[i], B[i], w1, w2, w3);
    end
  endgenerate
 endmodule
```

Verilog code and testbench for the arithmetic unit:

Verilog:

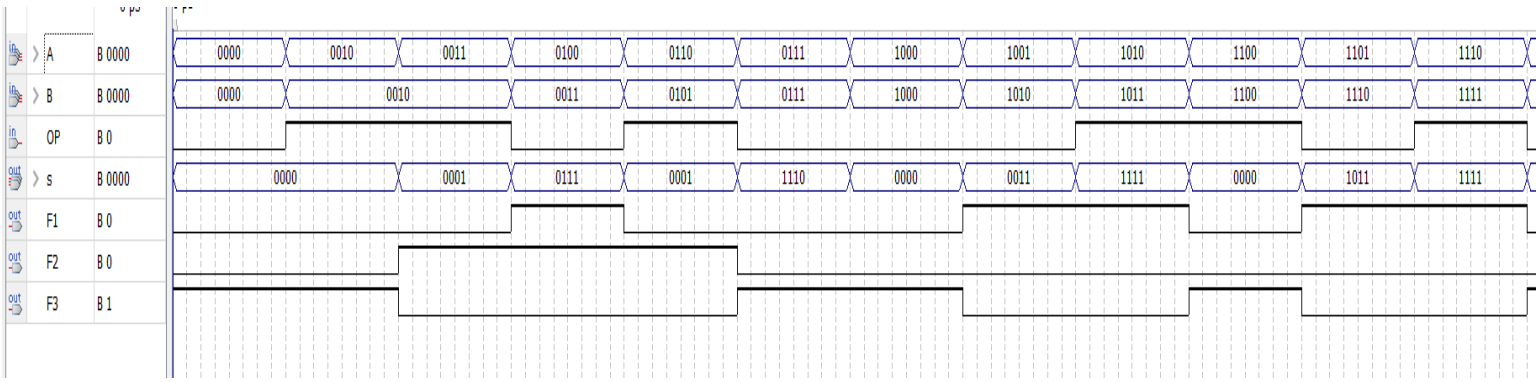
```
module arithmetic_unit (A, B, OP, s, cout, F1, F2, F3);
  parameter size = 4;

  input [size-1:0] A, B;
  input OP;
  output [size-1:0] s;
  output cout, F1, F2, F3;

  genvar i;
  generate
    for (i=0 ; i<size ; i=i+1) begin: arithmetic_unit
        add_sub_OP U3(A, B, OP, s, cout);
        comparator U4(A, B, F1, F2, F3);
    end
  endgenerate
endmodule

```

Testbench:

```
module arithmetic_unit_tb();

parameter size = 4 ;

 reg [size-1:0] A, B;

 reg OP;

wire [size-1:0] s;

wire cout, F1, F2, F3;

arithmetic_unit DUT(A, B, OP, s, cout, F1, F2, F3);


always

 begin

 A=4'b0000; B=4'b0000; OP=1; #100;

 A=4'b0001; B=4'b0001; OP=0; #100;

A=4'b0010; B=4'b0001; OP=1; #100;

A=4'b0011; B=4'b0010; OP=1; #100;

A=4'b0011; B=4'b0100; OP=1; #100;

A=4'b0100; B=4'b0101; OP=1; #100;

A=4'b0101; B=4'b0110; OP=1; #100;

A=4'b0111; B=4'b0111; OP=1; #100;

A=4'b1000; B=4'b0111; OP=1; #100;

A=4'b1001; B=4'b1000; OP=1; #100;

A=4'b1010; B=4'b1001; OP=1; #100;

A=4'b1011; B=4'b1011; OP=1; #100;
```

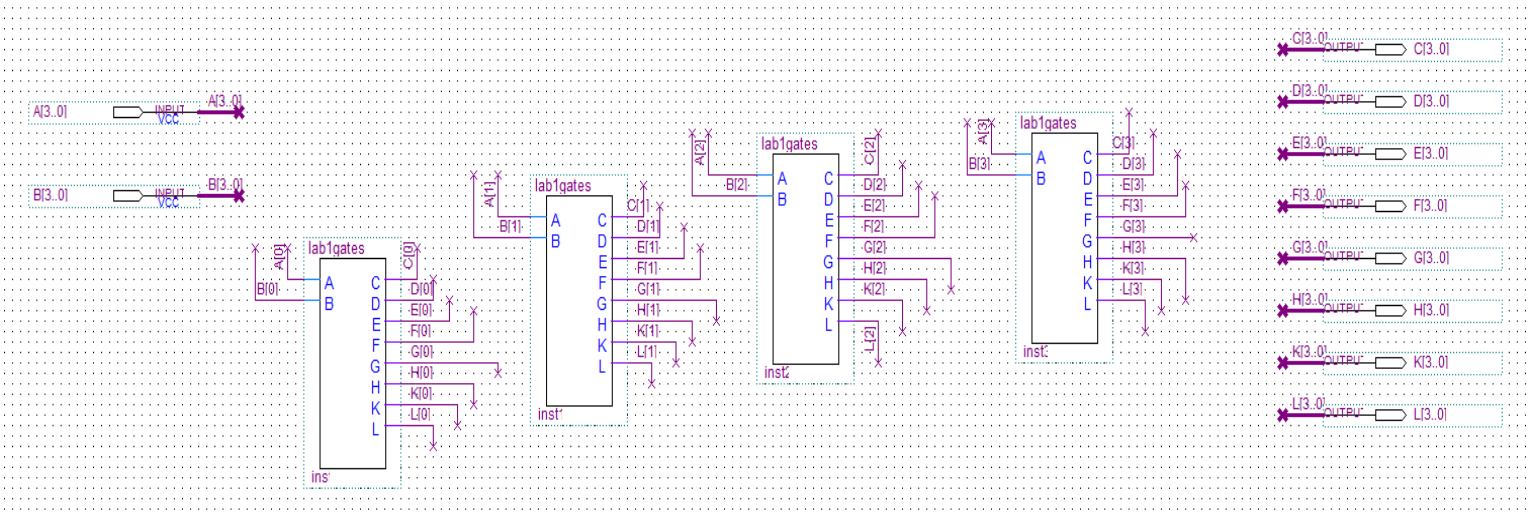
```
A=4'b1100; B=4'b1100; OP=1; #100;
A=4'b1101; B=4'b1100; OP=1; #100;
A=4'b1110; B=4'b1111; OP=1; #100;
A=4'b1111; B=4'b1111; OP=1; #100;
 A=4'b0000; B=4'b0000; OP=0; #100;
A=4'b0001; B=4'b0001; OP=1; #100;
A=4'b0010; B=4'b0001; OP=0; #100;
A=4'b0011; B=4'b0010; OP=0; #100;
A=4'b0011; B=4'b0100; OP=0; #100;
A=4'b0100; B=4'b0101; OP=0; #100;
A=4'b0101; B=4'b0110; OP=0; #100;
 A=4'b0111; B=4'b0111; OP=0; #100;
 A=4'b1000; B=4'b0111; OP=0; #100;
 A=4'b1001; B=4'b1000; OP=0; #100;
A=4'b1010; B=4'b1001; OP=0; #100;
 A=4'b1011; B=4'b1011; OP=0; #100;
 A=4'b1100; B=4'b1100; OP=0; #100;
A=4'b1101; B=4'b1100; OP=0; #100;
 A=4'b1110; B=4'b1111; OP=0; #100;
  A=4'b1111; B=4'b1111; OP=0; #100;
 end
endmodule
```

The waveform:

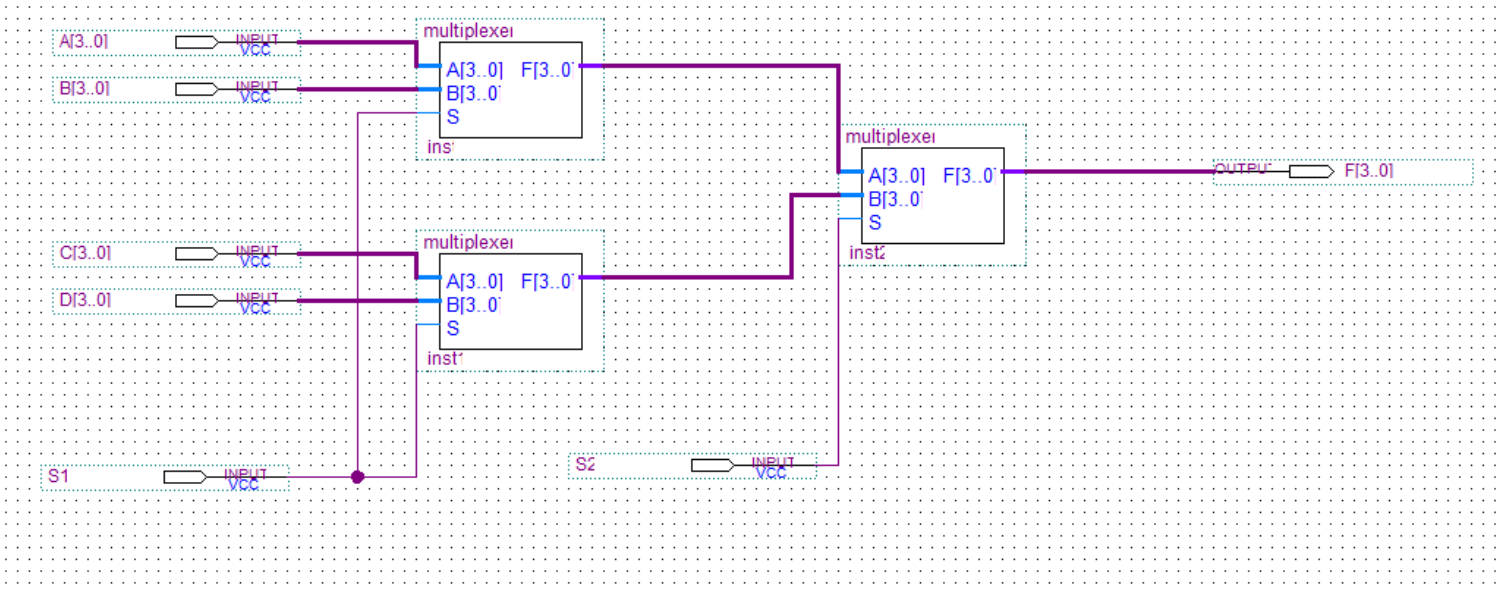


2.2.7 LOGIC UNIT WITH MULTIPLEXERS

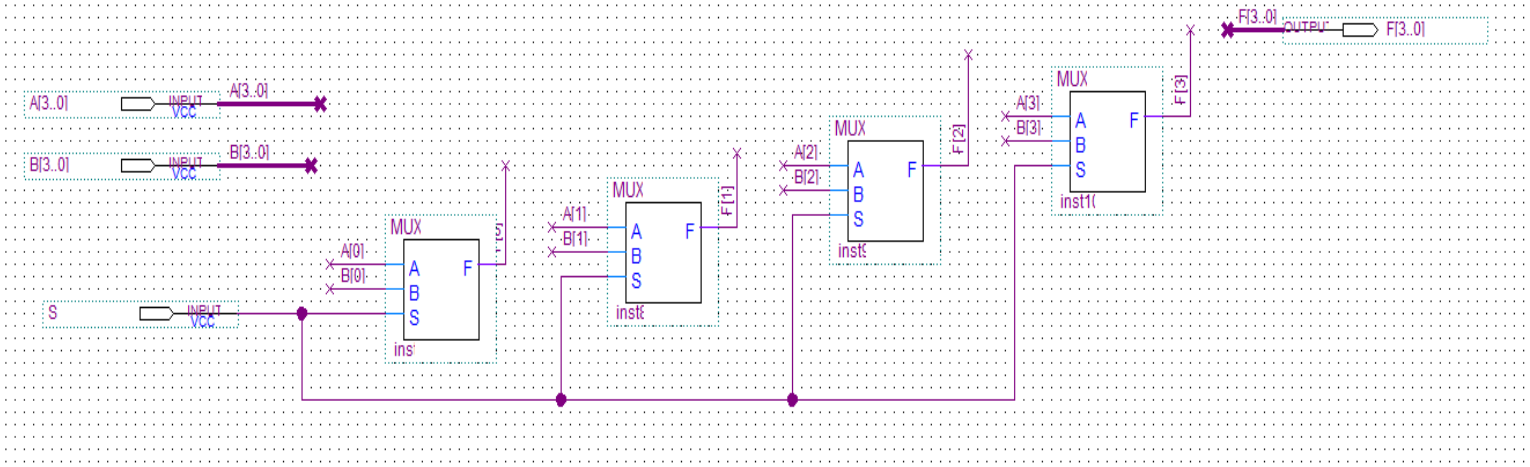
Gates:



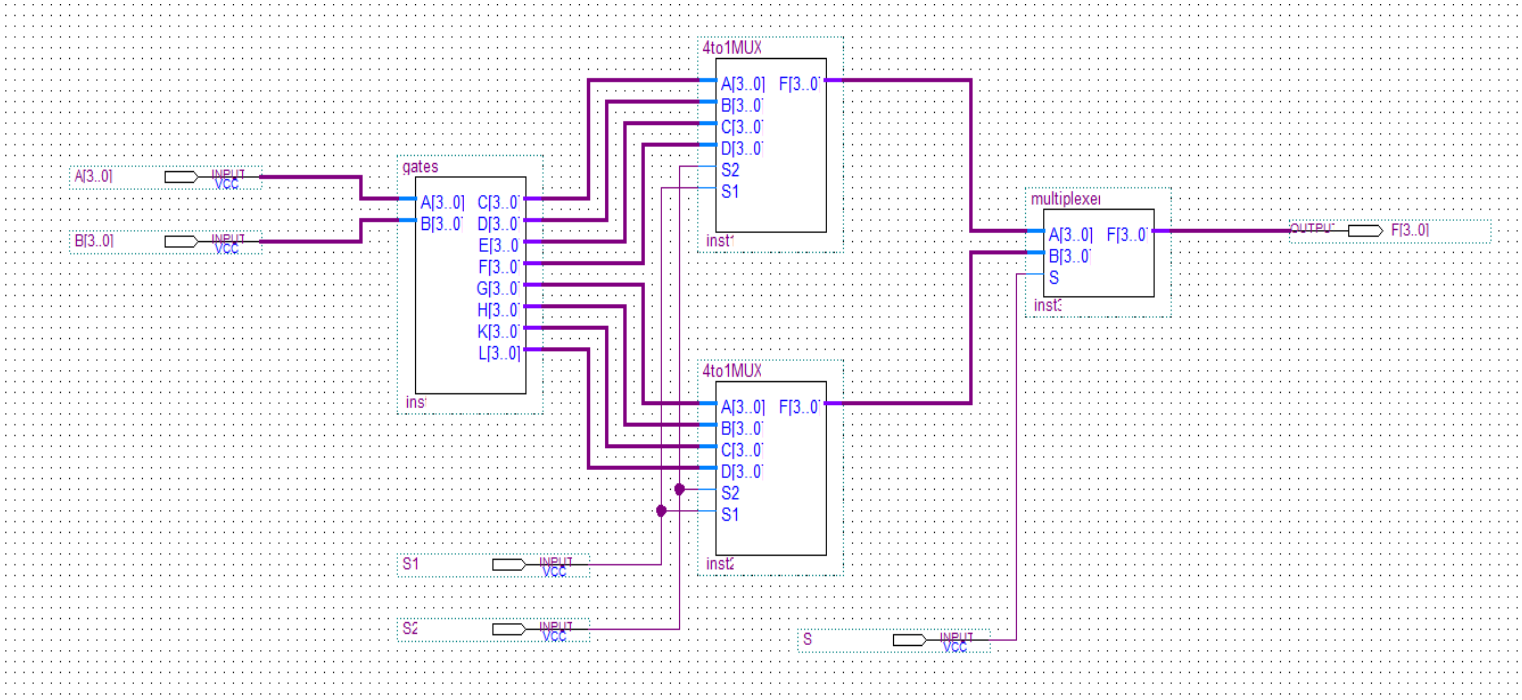
4to1MUX:



Multiplexer:



Logic Unit:



Verilog code for gates:

```
1   module gates(A, B, C, D, E, F, G, H, K, L);
2     parameter size =4 ;
3
4     input [size-1:0] A, B;
5     output [size-1:0] C, D, E, F, G, H, K, L;
6
7     always @(A, B);
8       lablgates U5(A, B, C, D, E, F, G, H, K, L);
9
10  endmodule
11
```

Verilog code for 4to1MUX:

```
1   module x4to1MUX(A, B, C, D, S1, S2, F);
2     parameter size = 4;
3
4     input [size-1:0] A, B, C, D;
5     input S1, S2;
6     output [size-1:0] F;
7     wire [size-1:0] w1, w2;
8
9
10        always @(A, B, C, D, S1, S2);
11         multiplexer U2(A, B, S1, w1);
12         multiplexer U3(C, D, S1, w2);
13         multiplexer U4(w1, w2, S2, F);
14
15  endmodule
16
```

Verilog code for multiplexer:

```
module multiplexer(A, B, S, F);

  parameter size = 4;

  input [size-1:0] A, B;
  input S;
  output [size-1:0] F;

  genvar i;
   generate
     for (i=0 ; i<size; i=i+1) begin: multiplexer

      MUX U1(A[i], B[i], S, F[i]);

     end
   endgenerate
endmodule

```

Verilog code and testbench for the logic unit:

Verilog:

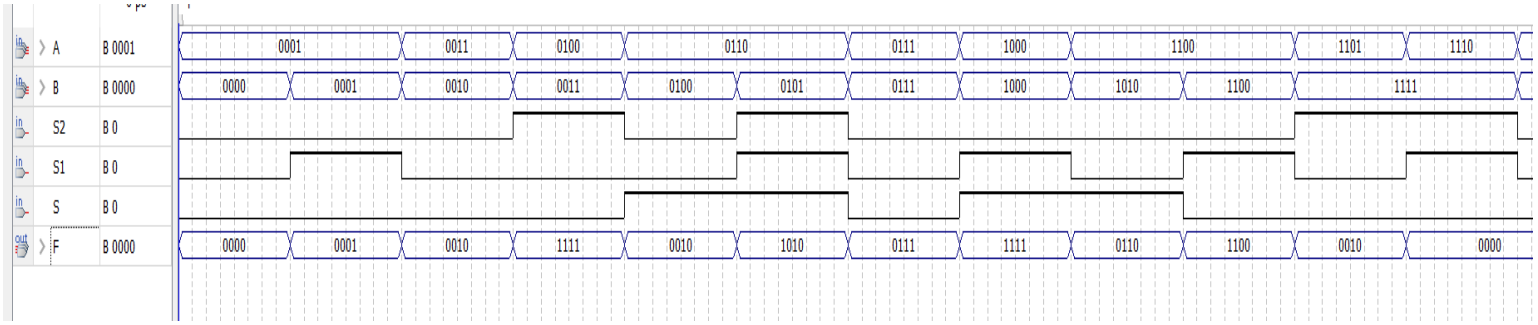
```
module logic_unit(A, B, S1, S2, S, F);
  parameter size = 4 ;

  input [size-1:0] A, B;
  input S1, S2, S;
  output [size-1:0] F;
  wire [size-1:0] w1, w2, w3, w4, w5, w6, w7, w8, w9, w10;


  genvar i;
   generate
     for (i=0 ; i<size; i=i+1) begin: logic_unit

        gates U6(A, B, w1, w2, w3, w4, w5, w6, w7, w8);
        x4to1MUX U7(w1, w2, w3, w4, S1, S2, w9);
        x4to1MUX U8(w5, w6, w7, w8, S1, S2, w10);
        multiplexer U9(w9, w10, S, F);

      end
    endgenerate
endmodule
```

Testbench:

```
module logic_unit_tb();
        parameter size = 4 ;
        reg [size-1:0] A, B;
        reg S1, S2, S;
        wire [size-1:0] F;
```

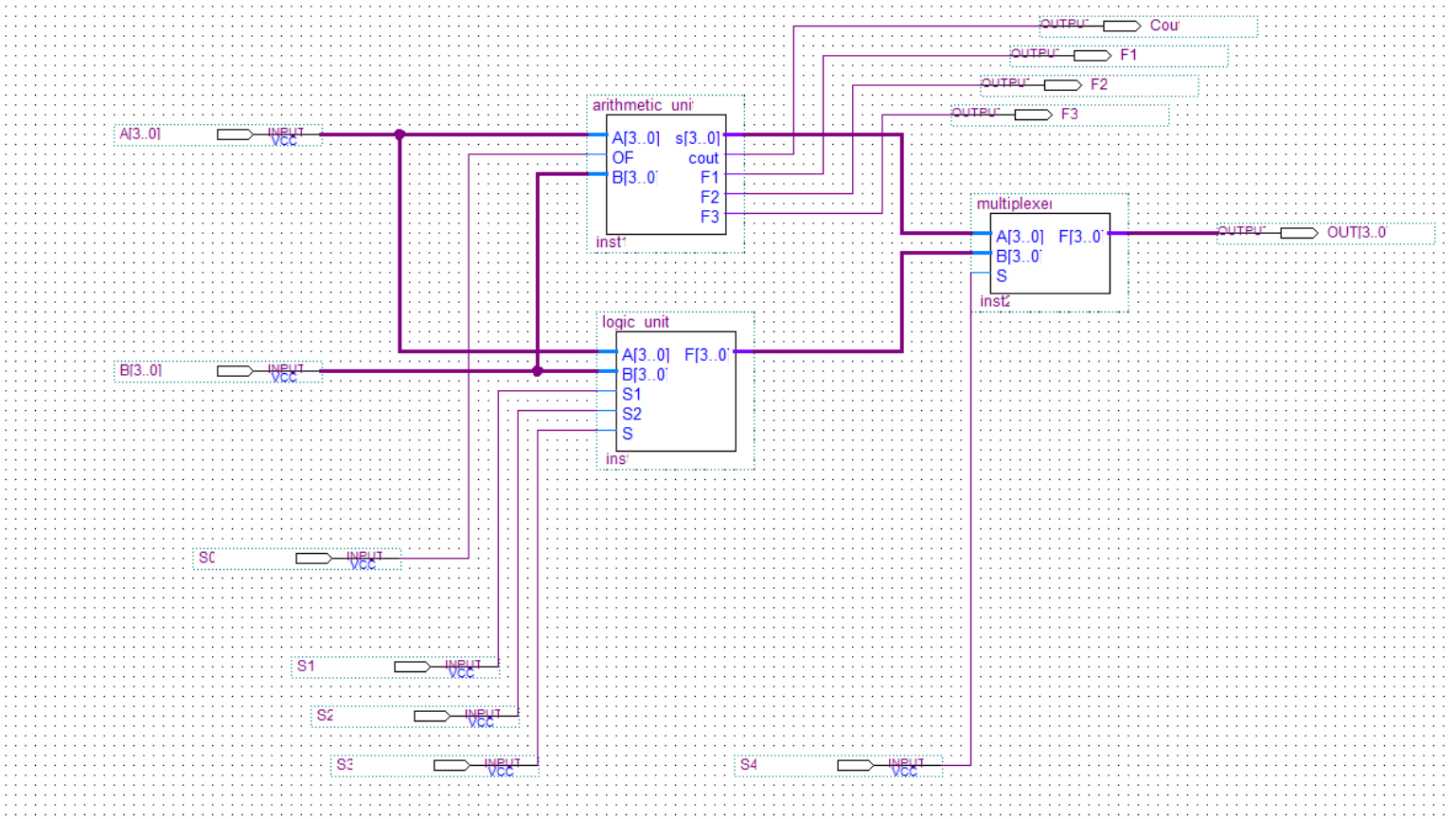
```
        logic_unit DUT(A, B, S1, S2, S, F);
always
  begin
    A=4'b0000; B=4'b0000; S1=1'b0; S2=1'b0; S=1'b0; #100;
    A=4'b0001; B=4'b0001; S1=1'b0; S2=1'b1; S=1'b1; #100;
    A=4'b0010; B=4'b0001; S1=1'b0; S2=1'b1; S=1'b0; #100;
    A=4'b0011; B=4'b0010; S1=1'b0; S2=1'b1; S=1'b1; #100;
    A=4'b0011; B=4'b0100; S1=1'b1; S2=1'b0; S=1'b0; #100;
    A=4'b0100; B=4'b0101; S1=1'b1; S2=1'b0; S=1'b1; #100;
    A=4'b0101; B=4'b0110; S1=1'b1; S2=1'b0; S=1'b1; #100;
    A=4'b0111; B=4'b0111; S1=1'b0; S2=1'b1; S=1'b0; #100;
    A=4'b1000; B=4'b0111; S1=1'b0; S2=1'b0; S=1'b0; #100;
    A=4'b1001; B=4'b1000; S1=1'b0; S2=1'b1; S=1'b1; #100;
    A=4'b1010; B=4'b1001; S1=1'b1; S2=1'b0; S=1'b0; #100;
    A=4'b1011; B=4'b1011; S1=1'b1; S2=1'b0; S=1'b0; #100;
    A=4'b1100; B=4'b1100; S1=1'b1; S2=1'b0; S=1'b1; #100;
    A=4'b1101; B=4'b1100; S1=1'b1; S2=1'b1; S=1'b0; #100;
    A=4'b1110; B=4'b1111; S1=1'b0; S2=1'b0; S=1'b1; #100;
    A=4'b1111; B=4'b1111; S1=1'b0; S2=1'b0; S=1'b1; #100;
    A=4'b0000; B=4'b0000; S1=1'b1; S2=1'b1; S=1'b1; #100;
    A=4'b0001; B=4'b0001; S1=1'b1; S2=1'b0; S=1'b0; #100;
    A=4'b0010; B=4'b0001; S1=1'b1; S2=1'b0; S=1'b1; #100;
    A=4'b0011; B=4'b0010; S1=1'b1; S2=1'b0; S=1'b0; #100;
    A=4'b0011; B=4'b0100; S1=1'b0; S2=1'b1; S=1'b1; #100;
    A=4'b0100; B=4'b0101; S1=1'b0; S2=1'b1; S=1'b0; #100;
    A=4'b0101; B=4'b0110; S1=1'b0; S2=1'b1; S=1'b0; #100;
    A=4'b0111; B=4'b0111; S1=1'b1; S2=1'b0; S=1'b1; #100;
    A=4'b1000; B=4'b0111; S1=1'b1; S2=1'b1; S=1'b1; #100;
    A=4'b1001; B=4'b1000; S1=1'b1; S2=1'b0; S=1'b0; #100;
    A=4'b1010; B=4'b1001; S1=1'b0; S2=1'b1; S=1'b1; #100;
    A=4'b1011; B=4'b1011; S1=1'b0; S2=1'b1; S=1'b1; #100;
    A=4'b1100; B=4'b1100; S1=1'b0; S2=1'b1; S=1'b0; #100;
    A=4'b1101; B=4'b1100; S1=1'b0; S2=1'b0; S=1'b1; #100;
    A=4'b1110; B=4'b1111; S1=1'b1; S2=1'b1; S=1'b0; #100;
    A=4'b1111; B=4'b1111; S1=1'b1; S2=1'b1; S=1'b0; #100;
  end
endmodule
```

The waveform:



2.2.8 ARITHMETIC LOGIC UNIT (ALU) TOP LEVEL DESIGN

The schematic:



Verilog code and testbench for ALU:

Verilog:

```
module ALU(A, B, S0, S1, S2, S3, S4, F1, F2, F3, Cout, OUT);
  parameter size = 4;
  input [size-1:0] A, B;
  input S0, S1, S2, S3, S4;
  output [size-1:0] OUT;
  output F1, F2, F3, Cout;
  wire [size-1:0] w1, w2;

   genvar i;
   generate
    for (i=0 ; i<size ; i=i+1) begin: ALU
      arithmetic_unit U11(A, B, S0, w1, Cout, F1, F2, F3);
      logic_unit U12(A, B, S1, S2, S3, w2);
      multiplexer U13(w1, w2, S4, OUT);
    end
  endgenerate
 endmodule

```

Testbench:

```
module ALU_tb();
 parameter size = 4 ;

 reg [size-1:0] A, B;
 reg S0, S1, S2, S3, S4;
 wire [size-1:0] OUT;
 wire Cout, F1, F2, F3;

 ALU DUT(A, B, S0, S1, S2, S3, S4, F1, F2, F3, Cout, OUT);

 always
 begin
    A=4'b0000; B=4'b0000; S1=1'b0; S2=1'b0; S0=1'b0; S3=1'b0; S4=1'b0; #100;
    A=4'b0001; B=4'b0001; S1=1'b0; S2=1'b1; S0=1'b1; S3=1'b0; S4=1'b0; #100;
    A=4'b0010; B=4'b0001; S1=1'b0; S2=1'b1; S0=1'b0; S3=1'b1; S4=1'b1; #100;
    A=4'b0011; B=4'b0010; S1=1'b0; S2=1'b1; S0=1'b1; S3=1'b0; S4=1'b0; #100;
    A=4'b0011; B=4'b0100; S1=1'b1; S2=1'b0; S0=1'b0; S3=1'b0; S4=1'b1; #100;
    A=4'b0100; B=4'b0101; S1=1'b1; S2=1'b0; S0=1'b1; S3=1'b1; S4=1'b0; #100;
    A=4'b0101; B=4'b0110; S1=1'b1; S2=1'b0; S0=1'b1; S3=1'b0; S4=1'b1; #100;
    A=4'b0111; B=4'b0111; S1=1'b0; S2=1'b1; S0=1'b0; S3=1'b1; S4=1'b0; #100;
    A=4'b1000; B=4'b0111; S1=1'b0; S2=1'b0; S0=1'b0; S3=1'b0; S4=1'b1; #100;
    A=4'b1001; B=4'b1000; S1=1'b0; S2=1'b1; S0=1'b1; S3=1'b0; S4=1'b0; #100;
    A=4'b1010; B=4'b1001; S1=1'b1; S2=1'b0; S0=1'b0; S3=1'b1; S4=1'b0; #100;
    A=4'b1011; B=4'b1011; S1=1'b1; S2=1'b0; S0=1'b0; S3=1'b1; S4=1'b1; #100;
    A=4'b1100; B=4'b1100; S1=1'b1; S2=1'b0; S0=1'b1; S3=1'b0; S4=1'b1; #100;
    A=4'b1101; B=4'b1100; S1=1'b1; S2=1'b1; S0=1'b0; S3=1'b1; S4=1'b0; #100;
    A=4'b1110; B=4'b1111; S1=1'b0; S2=1'b0; S0=1'b1; S3=1'b0; S4=1'b1; #100;
    A=4'b1111; B=4'b1111; S1=1'b0; S2=1'b0; S0=1'b1; S3=1'b1; S4=1'b0; #100;

    A=4'b0000; B=4'b0000; S1=1'b1; S2=1'b1; S0=1'b1; S3=1'b1; S4=1'b0; #100;
    A=4'b0001; B=4'b0001; S1=1'b1; S2=1'b0; S0=1'b0; S3=1'b0; S4=1'b0; #100;
    A=4'b0010; B=4'b0001; S1=1'b1; S2=1'b0; S0=1'b1; S3=1'b0; S4=1'b1; #100;
```

```
    A=4'b0011; B=4'b0010; S1=1'b1; S2=1'b0; S0=1'b0; S3=1'b1; S4=1'b1; #100;
    A=4'b0011; B=4'b0100; S1=1'b0; S2=1'b1; S0=1'b1; S3=1'b1; S4=1'b0; #100;
    A=4'b0100; B=4'b0101; S1=1'b0; S2=1'b1; S0=1'b0; S3=1'b0; S4=1'b0; #100;
    A=4'b0101; B=4'b0110; S1=1'b0; S2=1'b1; S0=1'b0; S3=1'b0; S4=1'b0; #100;
    A=4'b0111; B=4'b0111; S1=1'b1; S2=1'b0; S0=1'b1; S3=1'b1; S4=1'b0; #100;
    A=4'b1000; B=4'b0111; S1=1'b1; S2=1'b1; S0=1'b1; S3=1'b0; S4=1'b0; #100;
    A=4'b1001; B=4'b1000; S1=1'b1; S2=1'b0; S0=1'b0; S3=1'b1; S4=1'b1; #100;
    A=4'b1010; B=4'b1001; S1=1'b0; S2=1'b1; S0=1'b1; S3=1'b0; S4=1'b0; #100;
    A=4'b1011; B=4'b1011; S1=1'b0; S2=1'b1; S0=1'b1; S3=1'b1; S4=1'b0; #100;
    A=4'b1100; B=4'b1100; S1=1'b0; S2=1'b1; S0=1'b0; S3=1'b0; S4=1'b1; #100;
    A=4'b1101; B=4'b1100; S1=1'b0; S2=1'b0; S0=1'b1; S3=1'b0; S4=1'b0; #100;
    A=4'b1110; B=4'b1111; S1=1'b1; S2=1'b1; S0=1'b0; S3=1'b1; S4=1'b0; #100;
    A=4'b1111; B=4'b1111; S1=1'b1; S2=1'b1; S0=1'b0; S3=1'b0; S4=1'b1; #100;

  end
endmodule
```

The waveform: