Faculty of Engineering and Technology

Department of Electrical and Computer Engineering

Digital System

ENCS 234

Verilog Assignment

Student Name: Nsreen Tawafsha.

Student ID: 1182319.

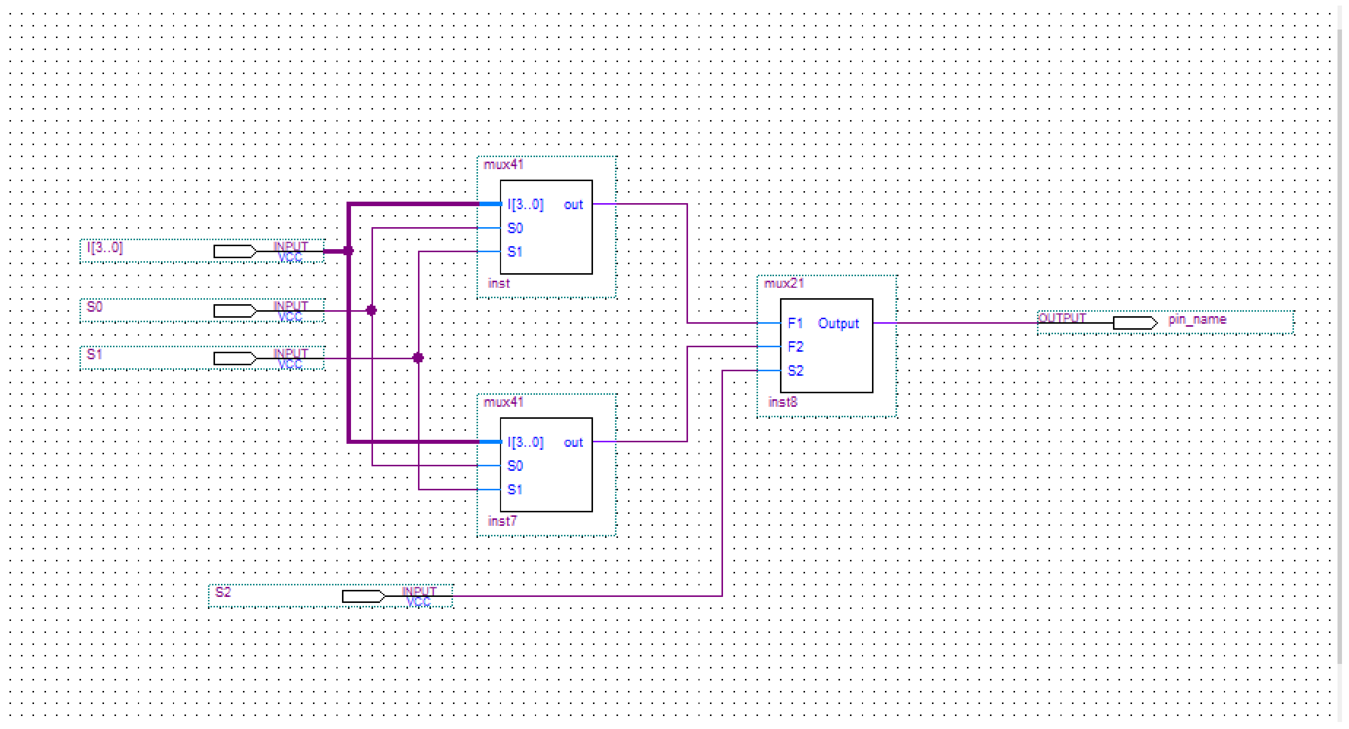
Section: 1.

Instructor: Mohammed Hussein.

Date: March 12, 2021

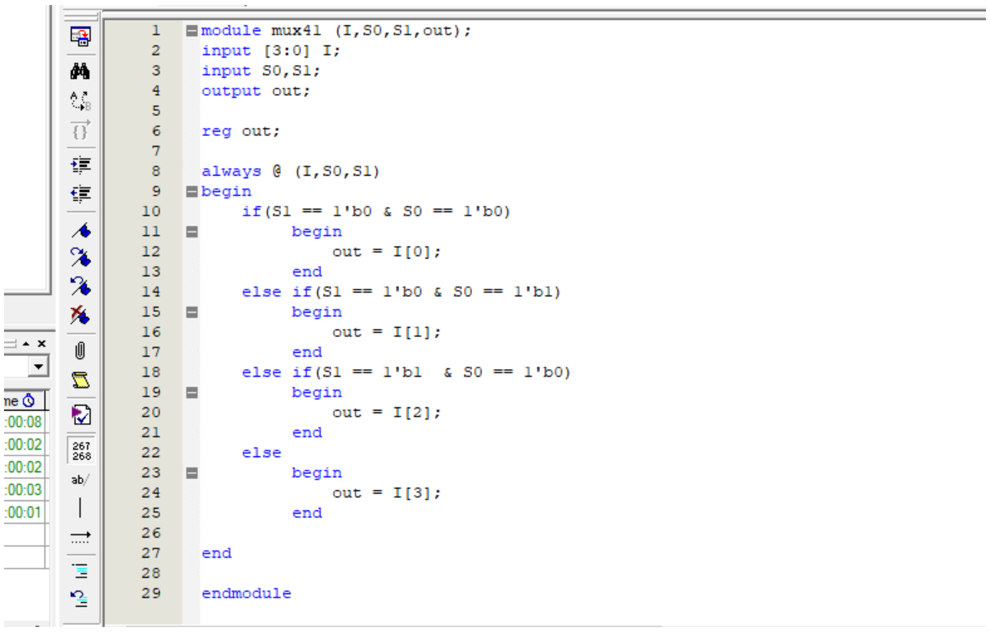
Question 1:

Whole System Design

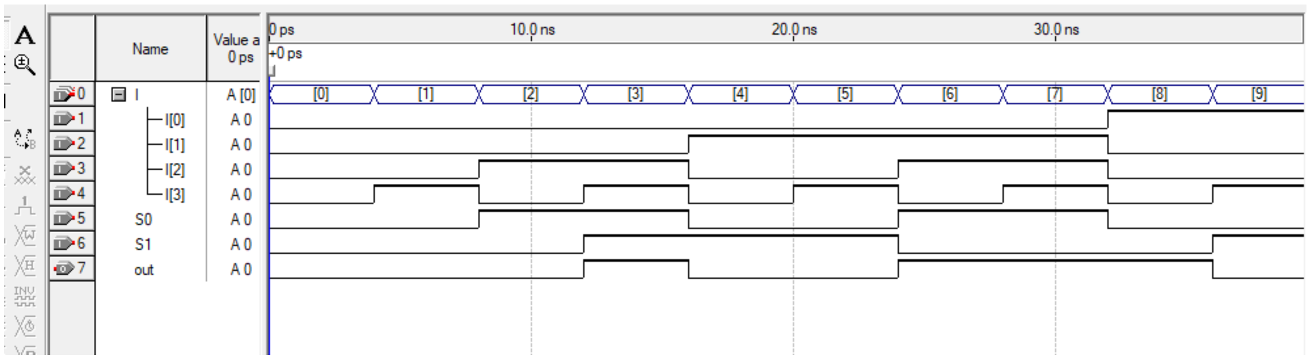


Verilog Codes and there simulations

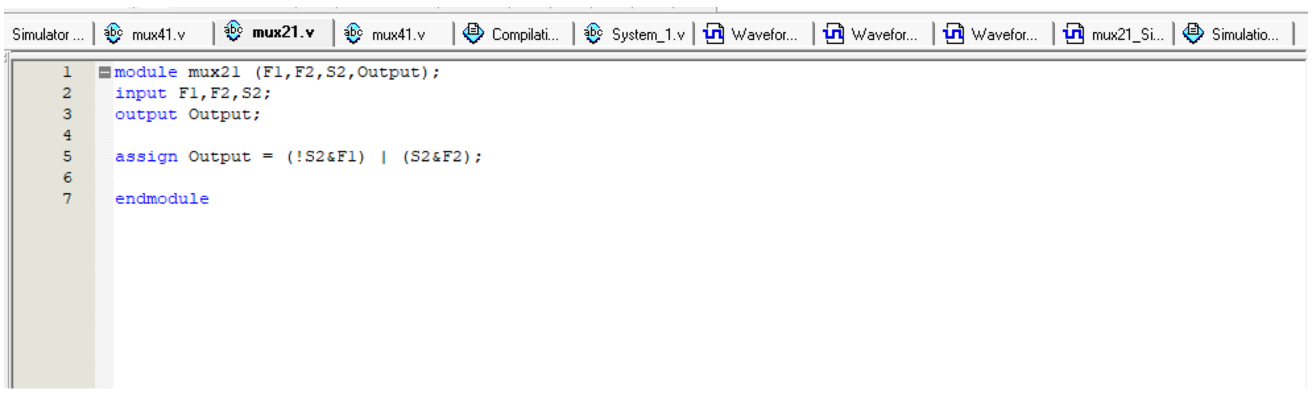
Mux4×1



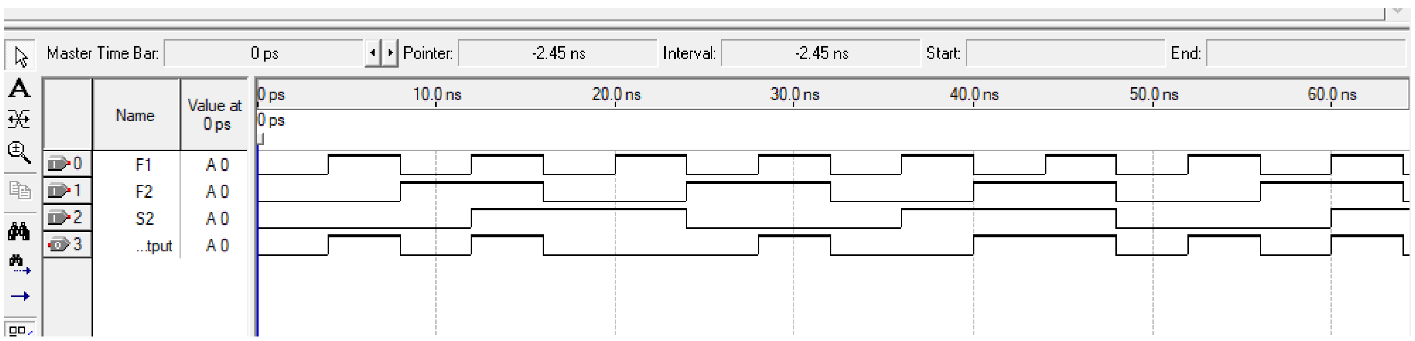
```
module mux41 (I,S0,S1,out);
input [3:0] I;
input S0,S1;
output out;

reg out;

always @ (I,S0,S1)
begin
     if(S1 == 1'b0 & S0 == 1'b0)
          begin
               out = I[0];
          end
     else if(S1 == 1'b0 & S0 == 1'b1)
          begin
               out = I[1];
          end
     else if(S1 == 1'b1  & S0 == 1'b0)
          begin
               out = I[2];
          end
     else
          begin
               out = I[3];
          end

end

endmodule
```



Mux2×1

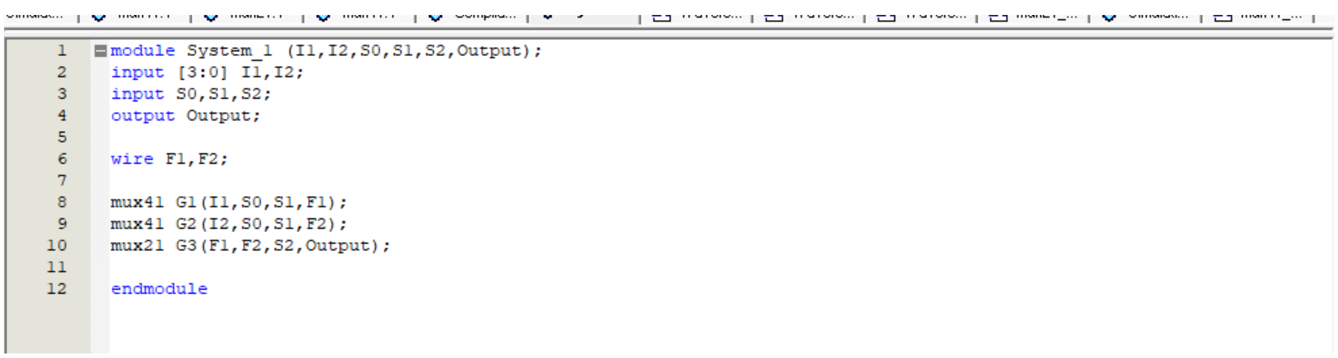


```
module mux21 (F1,F2,S2,Output);
input F1,F2,S2;
output Output;

assign Output = (!S2&F1) | (S2&F2);

endmodule
```



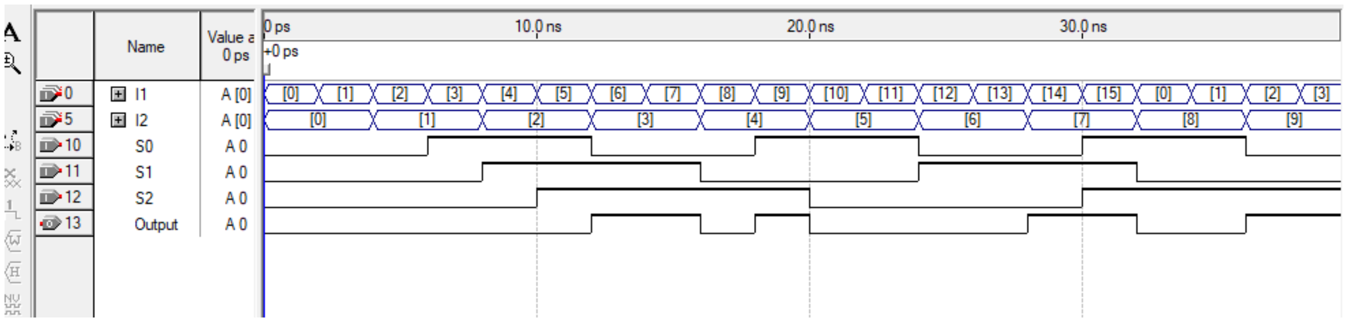
	Name	Value at 0 ps
0	F1	A 0
1	F2	A 0
2	S2	A 0
3	...tput	A 0

Whole System code



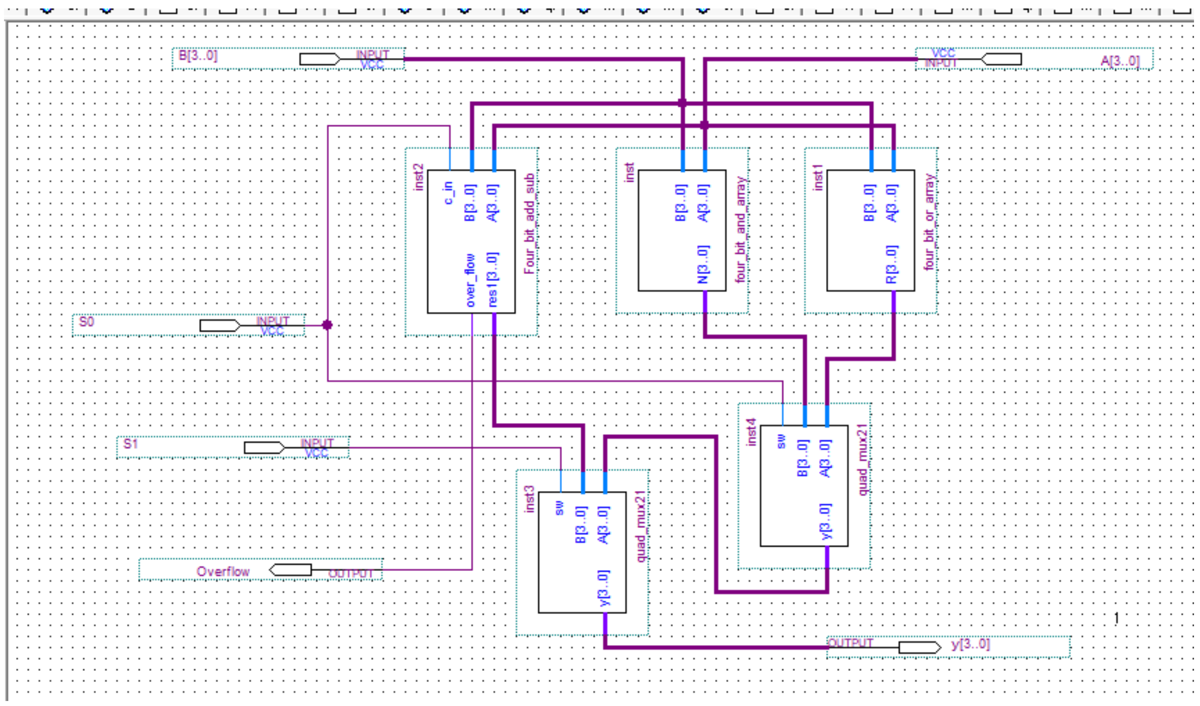
```
module System_1 (I1,I2,S0,S1,S2,Output);
input [3:0] I1,I2;
input S0,S1,S2;
output Output;

wire F1,F2;

mux41 G1(I1,S0,S1,F1);
mux41 G2(I2,S0,S1,F2);
mux21 G3(F1,F2,S2,Output);

endmodule
```

Whole Diagram Simulation



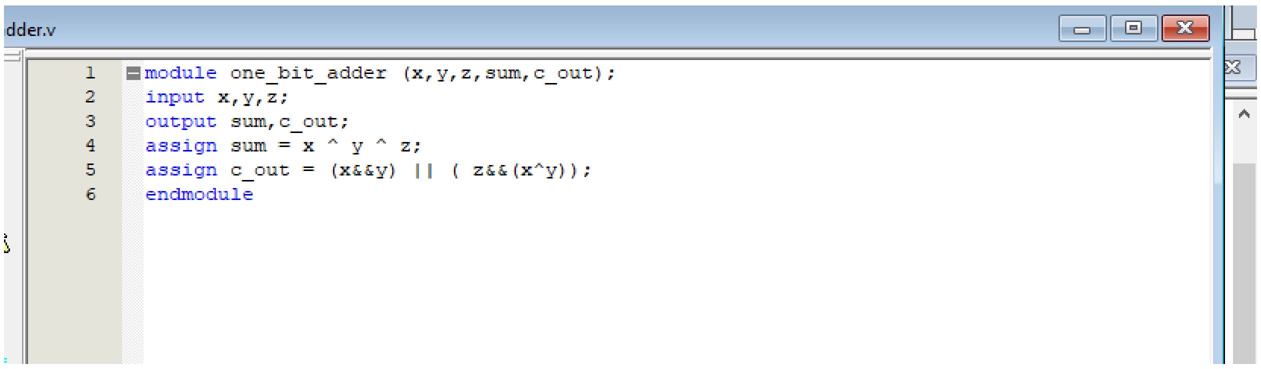
Question 2:

Whole System Design



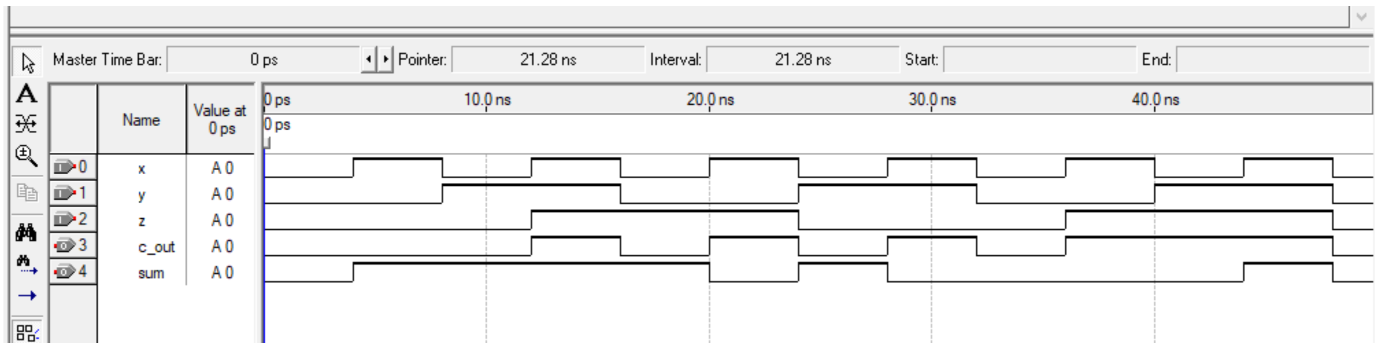
Verilog Codes and there simulations

One-Bit Adder



dder.v

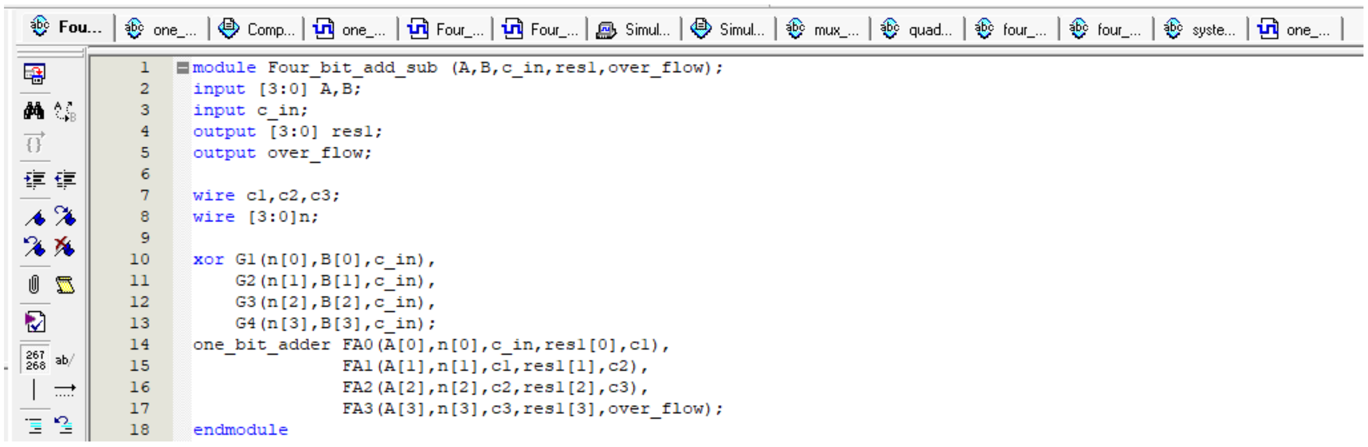
```
module one_bit_adder (x,y,z,sum,c_out);
input x,y,z;
output sum,c_out;
assign sum = x ^ y ^ z;
assign c_out = (x&&y) || ( z&&(x^y));
endmodule
```

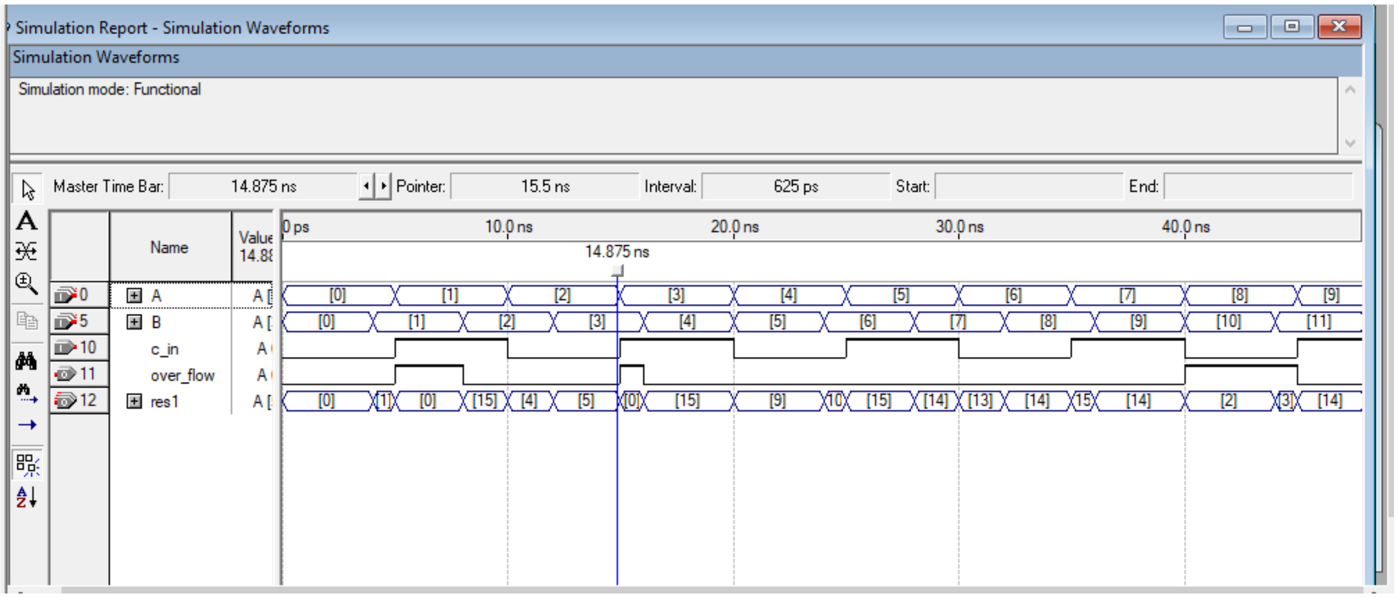


Master Time Bar: 0 ps Pointer: 21.28 ns Interval: 21.28 ns Start: End:

	Name	Value at 0 ps
0	x	A 0
1	y	A 0
2	z	A 0
3	c_out	A 0
4	sum	A 0

Four-Bit Adder


```
module Four_bit_add_sub (A,B,c_in,res1,over_flow);
input [3:0] A,B;
input c_in;
output [3:0] res1;
output over_flow;

wire c1,c2,c3;
wire [3:0]n;

xor G1(n[0],B[0],c_in),
    G2(n[1],B[1],c_in),
    G3(n[2],B[2],c_in),
    G4(n[3],B[3],c_in);
one_bit_adder FA0(A[0],n[0],c_in,res1[0],c1),
              FA1(A[1],n[1],c1,res1[1],c2),
              FA2(A[2],n[2],c2,res1[2],c3),
              FA3(A[3],n[3],c3,res1[3],over_flow);
endmodule
```

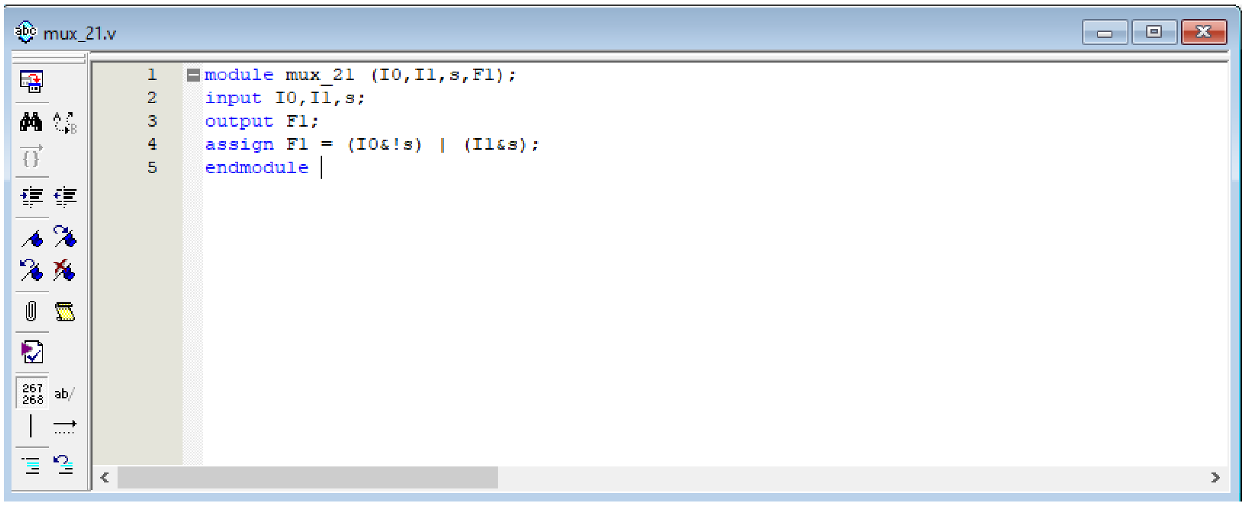

Simulation Report - Simulation Waveforms

Simulation Waveforms

Simulation mode: Functional

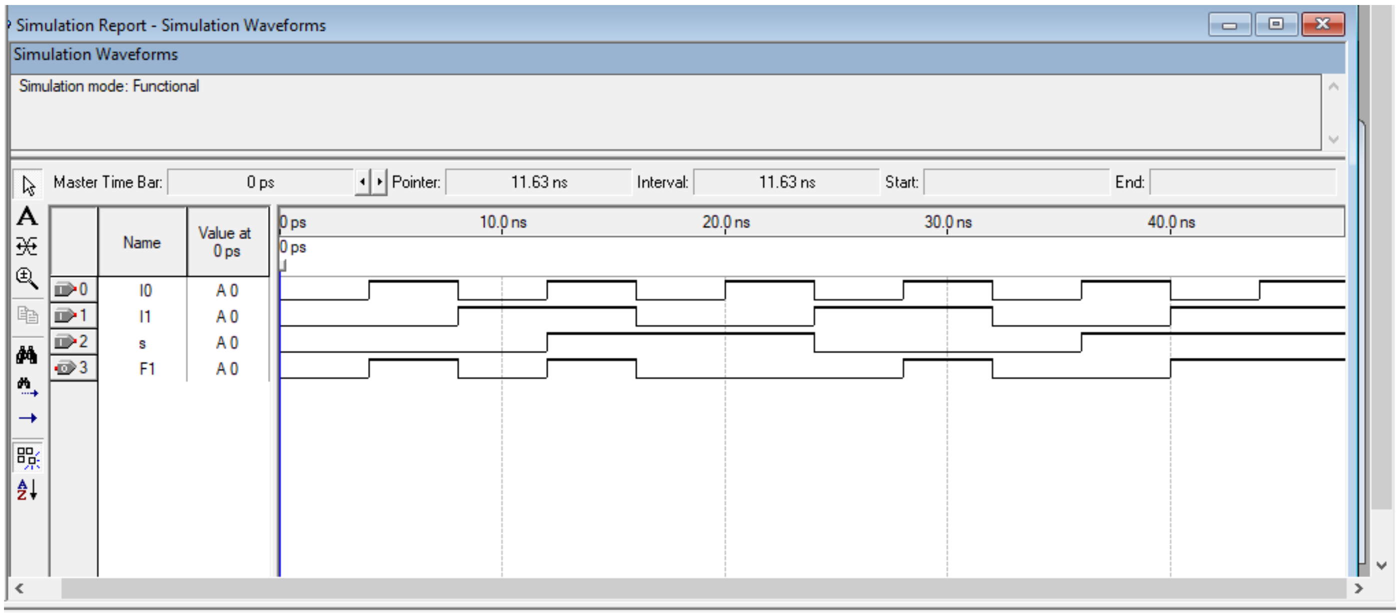
Master Time Bar: 14.875 ns Pointer: 15.5 ns Interval: 625 ps Start: End:

Mux2x1



mux_21.v

```
module mux_21 (I0,I1,s,F1);
input I0,I1,s;
output F1;
assign F1 = (I0&!s) | (I1&s);
endmodule
```



Simulation Report - Simulation Waveforms

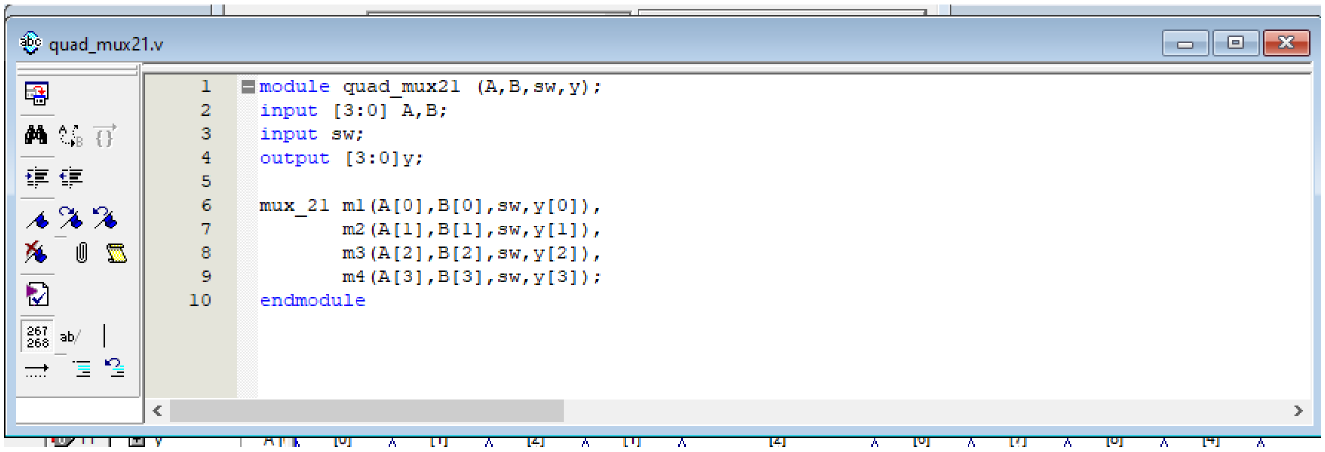
Simulation Waveforms

Simulation mode: Functional

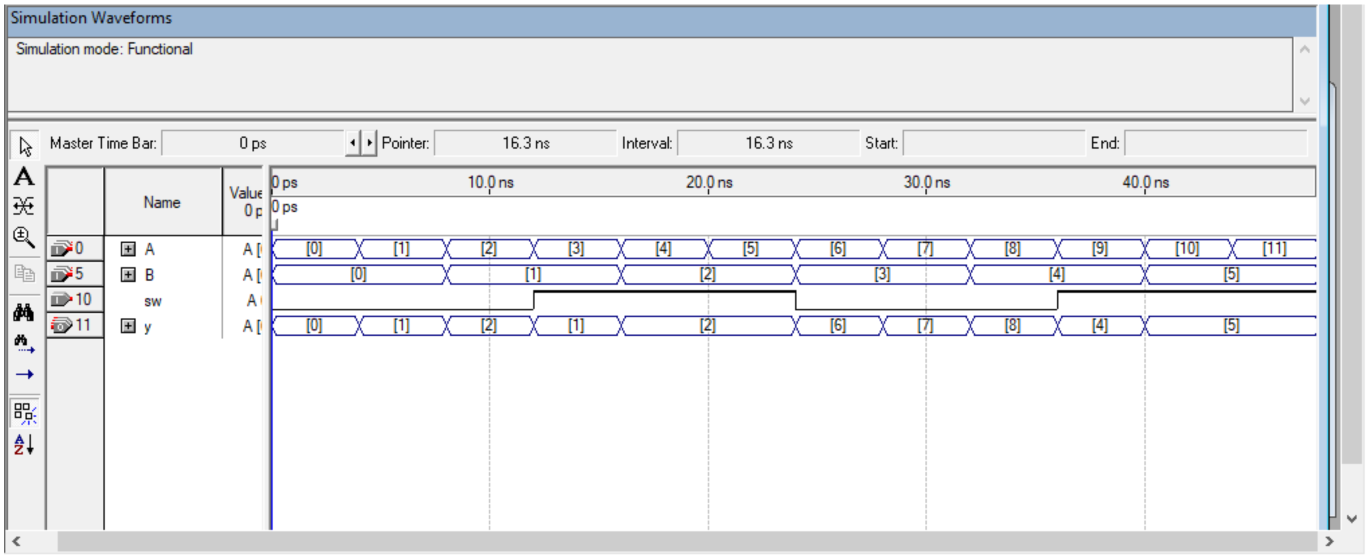
Master Time Bar: 0 ps   Pointer: 11.63 ns   Interval: 11.63 ns   Start: End:

	Name	Value at 0 ps
0	I0	A 0
1	I1	A 0
2	s	A 0
3	F1	A 0

Quad Mux 2x1



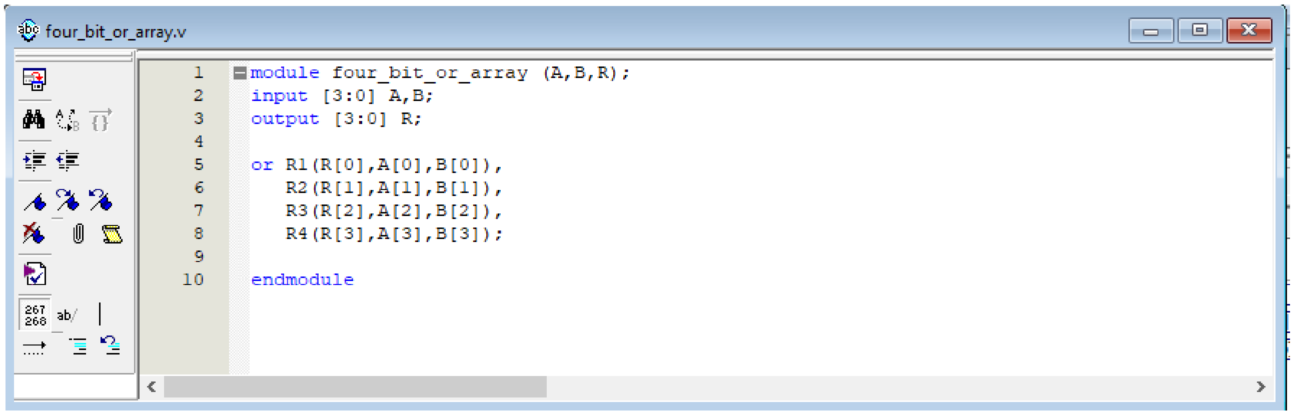
```verilog
module quad_mux21 (A,B,sw,y);
input [3:0] A,B;
input sw;
output [3:0]y;

mux_21 m1(A[0],B[0],sw,y[0]),
       m2(A[1],B[1],sw,y[1]),
       m3(A[2],B[2],sw,y[2]),
       m4(A[3],B[3],sw,y[3]);
endmodule
```


Simulation Waveforms
Name
Value
0 ps
10.0 ns
20.0 ns
30.0 ns
40.0 ns
A [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11]
B [0] [1] [2] [3] [4] [5]
sw
y [0] [1] [2] [1] [2] [6] [7] [8] [4] [5]

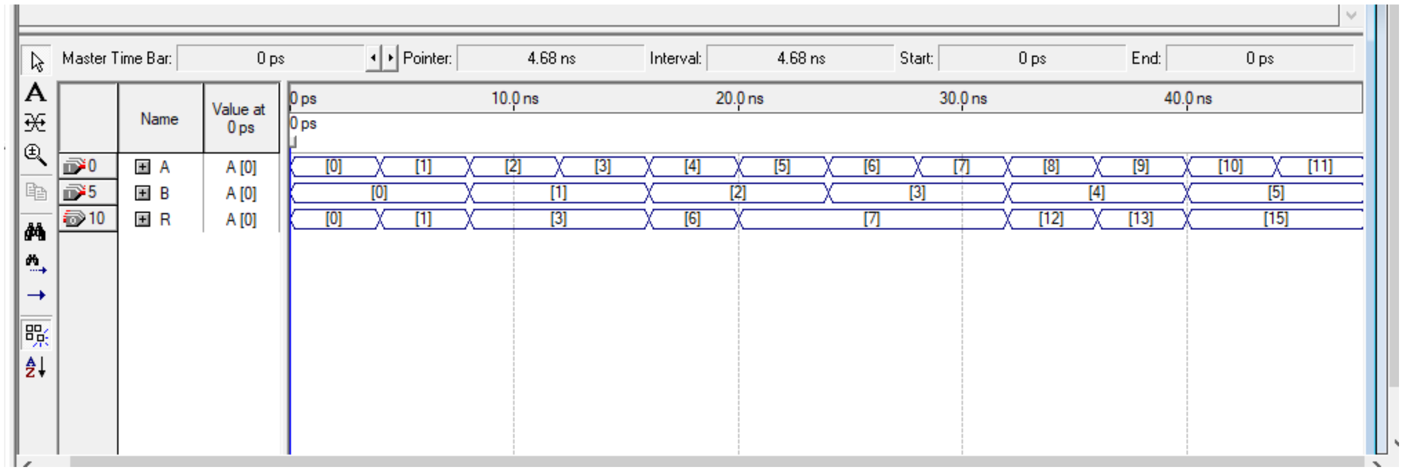
Simulation mode: Functional

Master Time Bar: 0 ps Pointer: 16.3 ns Interval: 16.3 ns Start: End:

Four-Bit or Array

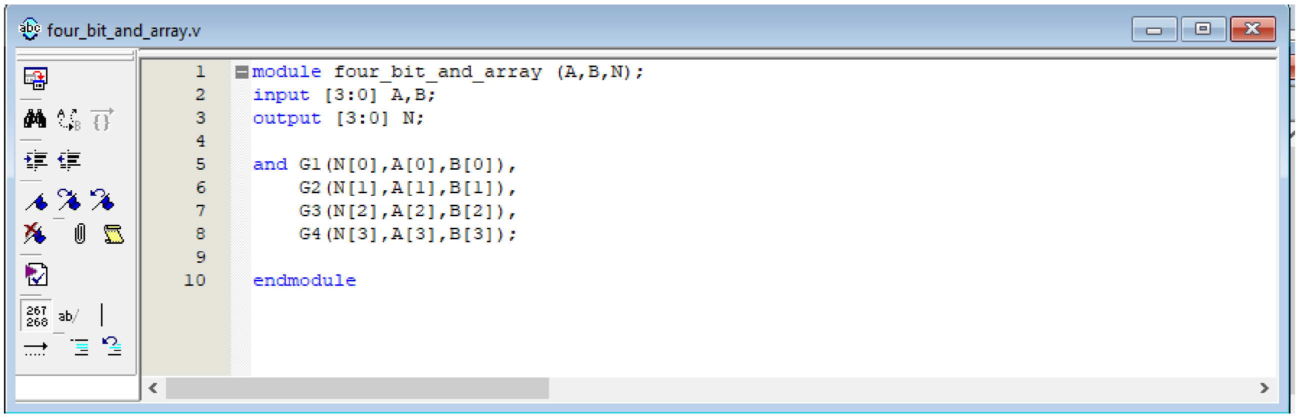


```
module four_bit_or_array (A,B,R);
input [3:0] A,B;
output [3:0] R;

or R1(R[0],A[0],B[0]),
   R2(R[1],A[1],B[1]),
   R3(R[2],A[2],B[2]),
   R4(R[3],A[3],B[3]);

endmodule
```



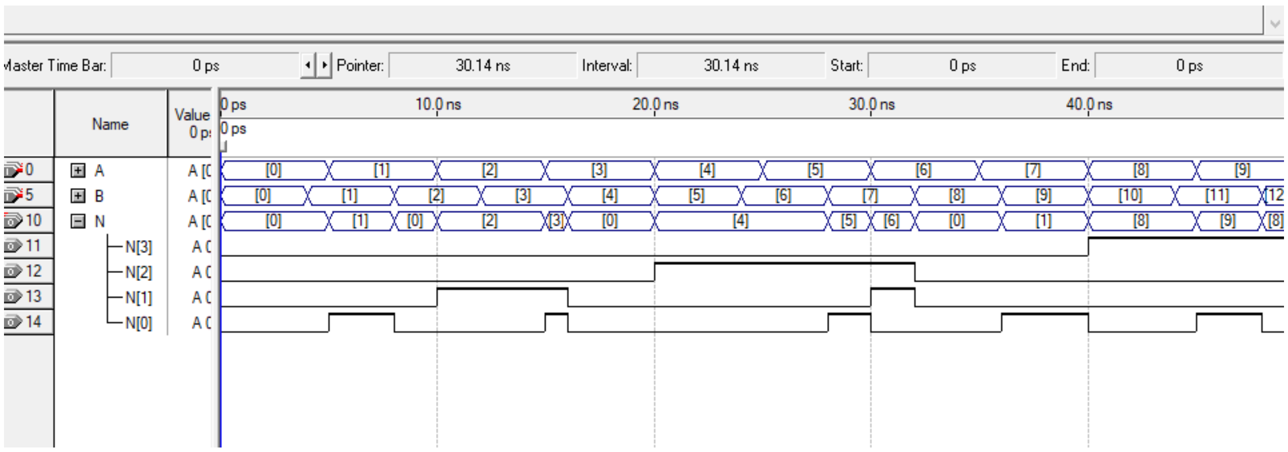
```
     Name   Value at 0 ps
0    A      A [0]   [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11]
5    B      A [0]   [0] [1] [2] [3] [4] [5]
10   R      A [0]   [0] [1] [3] [6] [7] [12] [13] [15]
```

Four-Bit and Array

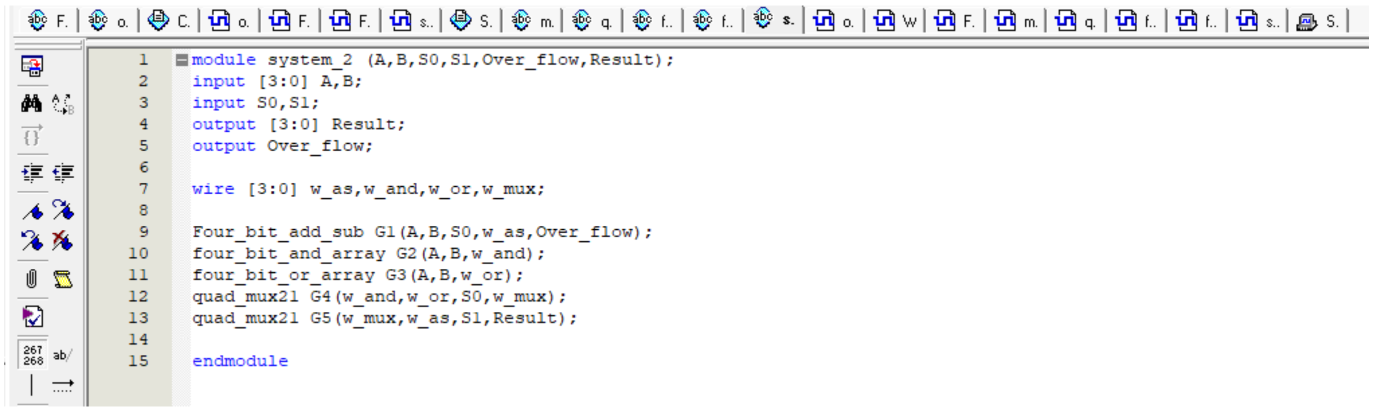


four_bit_and_array.v

```
module four_bit_and_array (A,B,N);
input [3:0] A,B;
output [3:0] N;

and G1(N[0],A[0],B[0]),
    G2(N[1],A[1],B[1]),
    G3(N[2],A[2],B[2]),
    G4(N[3],A[3],B[3]);

endmodule
```

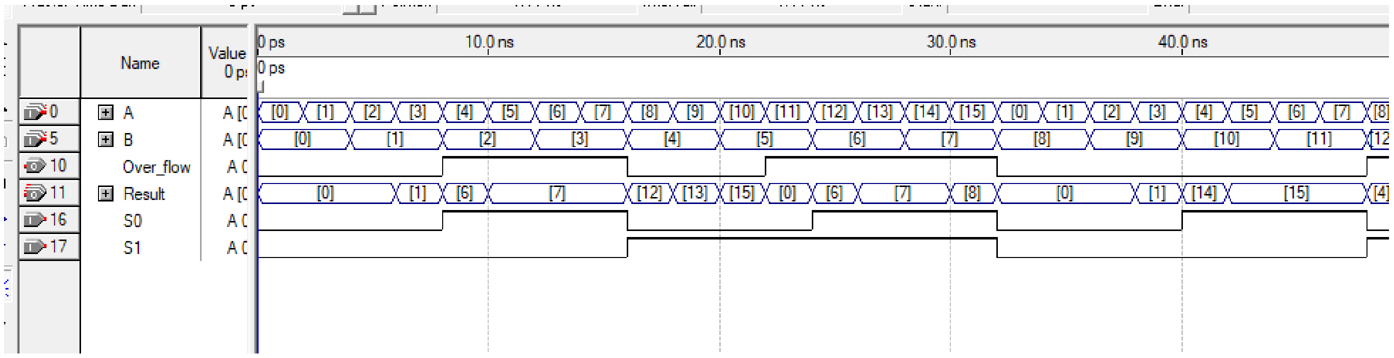


Whole System Code



```
module system_2 (A,B,S0,S1,Over_flow,Result);
input [3:0] A,B;
input S0,S1;
output [3:0] Result;
output Over_flow;

wire [3:0] w_as,w_and,w_or,w_mux;

Four_bit_add_sub G1(A,B,S0,w_as,Over_flow);
four_bit_and_array G2(A,B,w_and);
four_bit_or_array G3(A,B,w_or);
quad_mux21 G4(w_and,w_or,S0,w_mux);
quad_mux21 G5(w_mux,w_as,S1,Result);

endmodule
```

Whole Diagram Simulation


Name
Value
0 ps
10.0 ns
20.0 ns
30.0 ns
40.0 ns
A
B
Over_flow
Result
S0
S1
[0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [0] [1] [2] [3] [4] [5] [6] [7] [8]
[0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12
[0] [1] [6] [7] [12] [13] [15] [0] [6] [7] [8] [0] [1] [14] [15] [4]