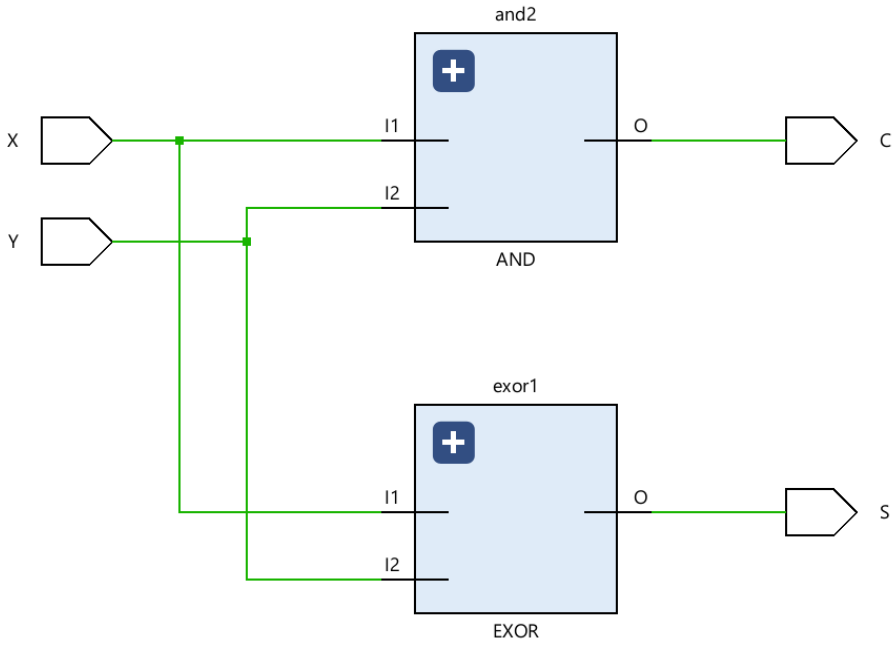
Ömer Faruk SERT

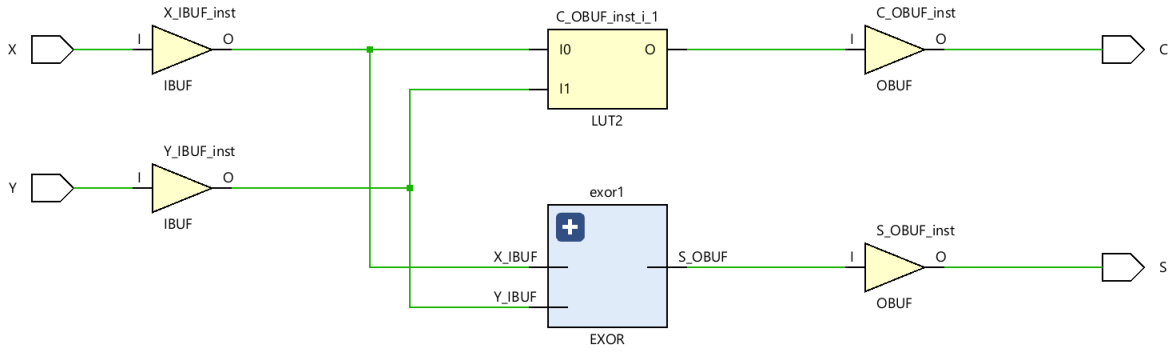
040170244

1- Half Adder

RTL Schematic

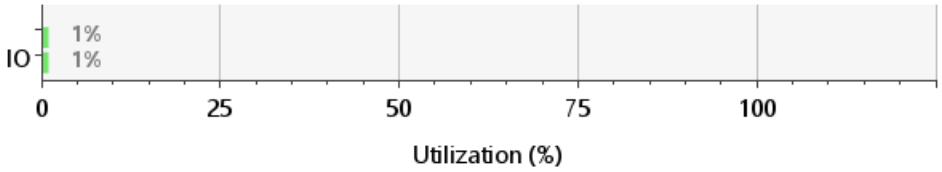


Technology Schematic



LUT Usage

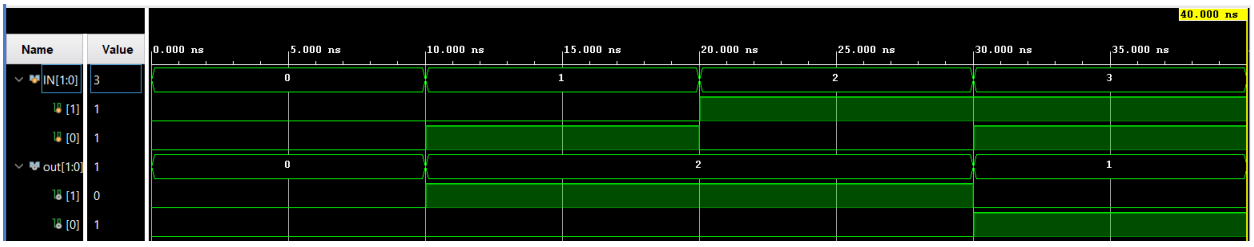
Resource	Utilization	Available	Utilization %
LUT	2	134600	0.00
IO	4	400	1.00



Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
X	C	5.592	SLOW	2.089	FAST
X	S	5.650	SLOW	2.129	FAST
Y	C	5.420	SLOW	2.018	FAST
Y	S	5.479	SLOW	2.062	FAST

Behavioral Simulation Wave Forms



Module Verilog Code

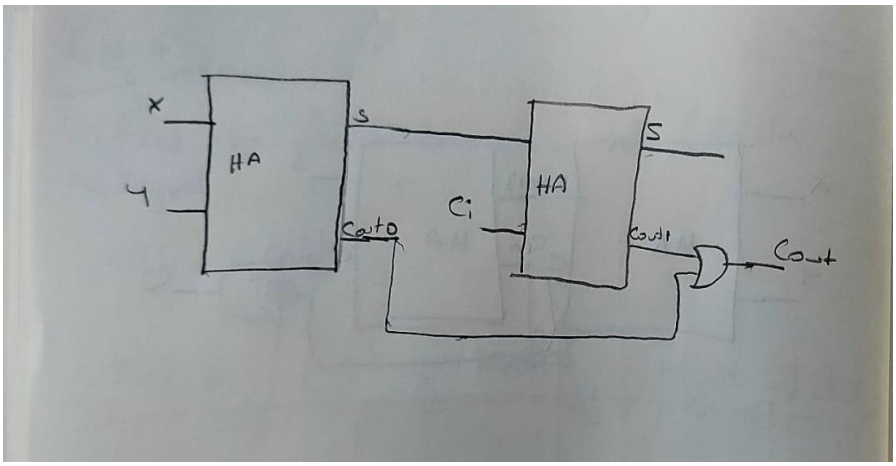
```
`timescale 1ns / 1ps

module HA(
    input X,Y,
    output Cout,S
);
    EXOR exor1(.I1(X),.I2(Y),.O(S));
    AND and2(.I1(X),.I2(Y),.O(Cout));
endmodule
```

Testbench Verilog Code

```
`timescale 1ns / 1ps

module Top_module_tb;
    reg [1:0]IN;
    wire [1:0]out;


    HA uut(.X(IN[0]),.Y(IN[1]),.Cout(out[0]),.S(out[1]));
    initial
    begin
        //inp = 1 sel = 00
        IN=2'b00;
        #10
        IN=2'b01;
        #10
        IN=2'b10;
        #10
        IN=2'b11;
        #10
        $finish;
    end
endmodule
```

2-) Full Adder

Hand Drawn Circuit



Verilog Code

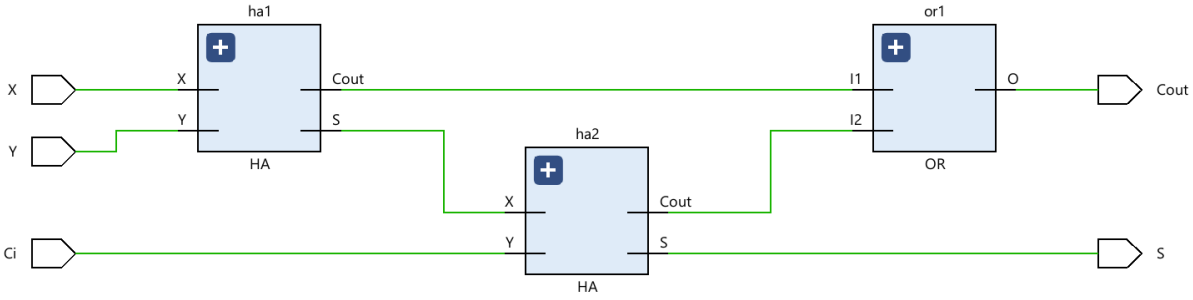
Module Code

```
`timescale 1ns / 1ps
module FA(
    input X,Y,Ci,
    output Cout,S
);
    wire [2:0]OX;
    HA ha1(.X(X),.Y(Y),.Cout(OX[0]),.S(OX[1]));
    HA ha2(.X(OX[1]),.Y(Ci),.S(S),.Cout(OX[2]));
    OR or1(.I1(OX[0]),.I2(OX[2]),.O(Cout));

endmodule
```

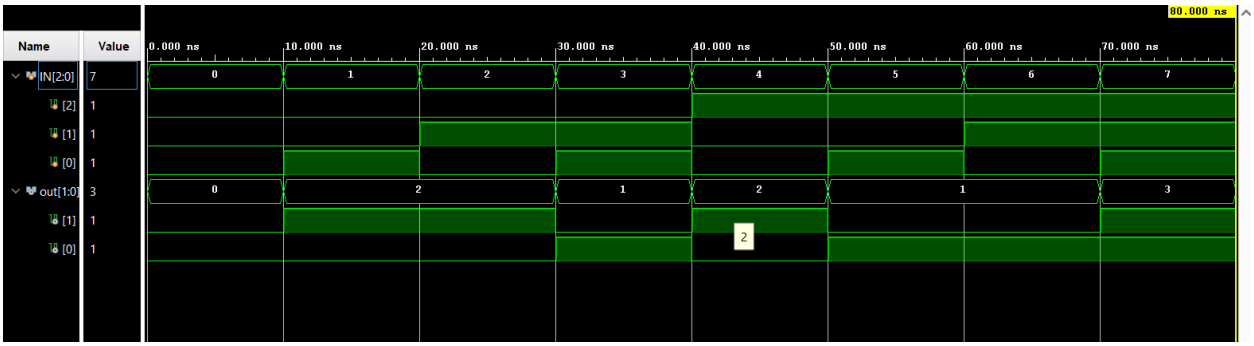
Testbench Code

```
`timescale 1ns / 1ps

module Top_module_tb;
    reg [2:0]IN;
    wire [1:0]out;


    FA uut(.X(IN[0]),.Y(IN[1]),.Ci(IN[2]),.Cout(out[0]),.S(out[1]));
    initial
    begin
        //inp = 1 sel = 00
        IN=3'b000;
        #10
        IN=3'b001;
        #10
        IN=3'b010;
        #10
        IN=3'b011;
        #10
        IN=3'b100;
        #10
        IN=3'b101;
        #10
        IN=3'b110;
        #10
        IN=3'b111;
        #10
        $finish;
    end
endmodule
```

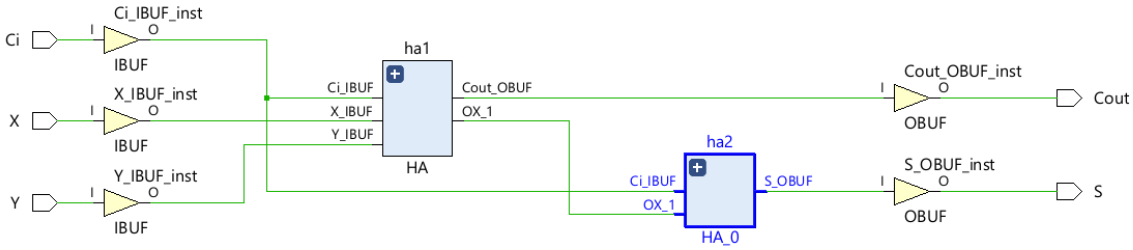
RTL Schematic



Behavioral Simulation Waveform



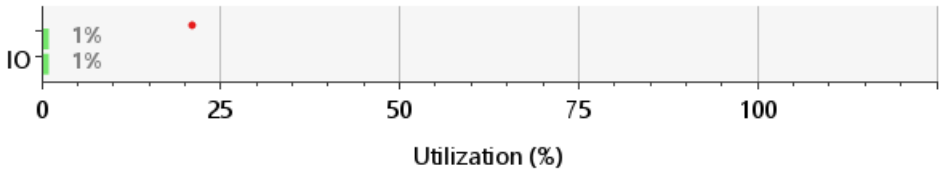
Technology Schematic



Design Summary

Summary

Resource	Utilization	Available	Utilization %
LUT	3	134600	0.00
IO	5	400	1.25



Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
Ci	Cout	4.638	SLOW	1.994	FAST
Ci	S	4.638	SLOW	1.994	FAST
X	Cout	5.101	SLOW	1.994	FAST
X	S	5.101	SLOW	2.217	FAST
Y	Cout	5.101	SLOW	1.994	FAST
Y	S	5.101	SLOW	2.217	FAST

3-) Ripple Carry Adder

Verilog Code

Module Code

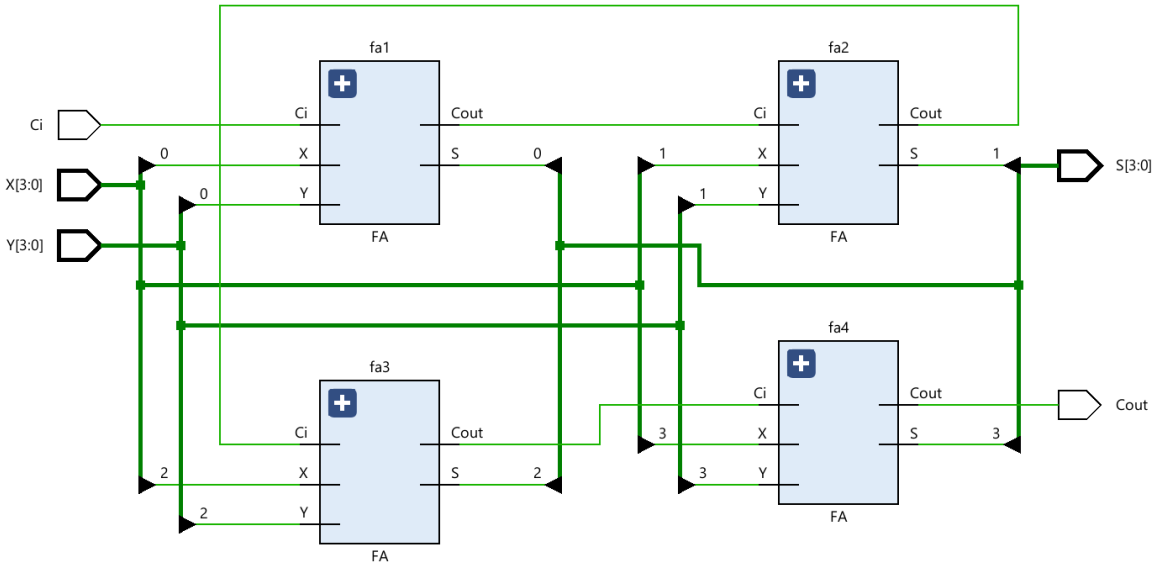
```
`timescale 1ns / 1ps
module RCA(
    input [3:0]X,[3:0]Y,
    input Ci,
    output [3:0]S,
    output Cout
);
    wire Ci;
    wire [2:0]OX;
    FA fa1(.X(X[0]),.Y(Y[0]),.Ci(Ci),.Cout(OX[0]),.S(S[0]));
    FA fa2(.X(X[1]),.Y(Y[1]),.Ci(OX[0]),.Cout(OX[1]),.S(S[1]));
    FA fa3(.X(X[2]),.Y(Y[2]),.Ci(OX[1]),.Cout(OX[2]),.S(S[2]));
    FA fa4(.X(X[3]),.Y(Y[3]),.Ci(OX[2]),.Cout(Cout),.S(S[3]));
endmodule
```

Testbench Code

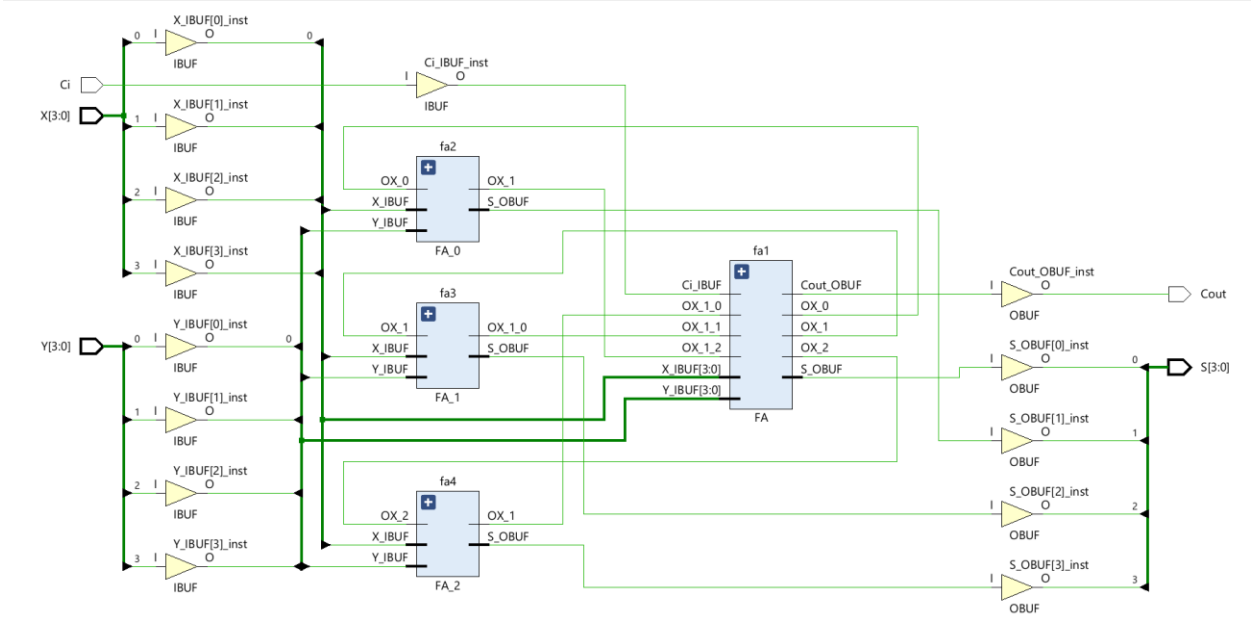
```
`timescale 1ns / 1ps

module Top_module_tb;
    reg [3:0]X;
    reg [3:0]Y;
    wire [3:0]S;
    wire Cout;
    wire Ci;
    assign Ci=1'b0;


    RCA uut(.X(X),.Y(Y),.Ci(Ci),.Cout(Cout),.S(S));
    initial
    begin
        //inp = 1 sel = 00
        X=4'b0000;
        Y=4'b0000;
        #10
        X=4'b0011;
        Y=4'b1011;
        #10
        X=4'b0101;
```

```
        Y=4'b1101;
        #10
        X=4'b1101;
        Y=4'b1101;
        #10
        X=4'b1111;
        Y=4'b1111;
        #10
        $finish;
    end
endmodule
```

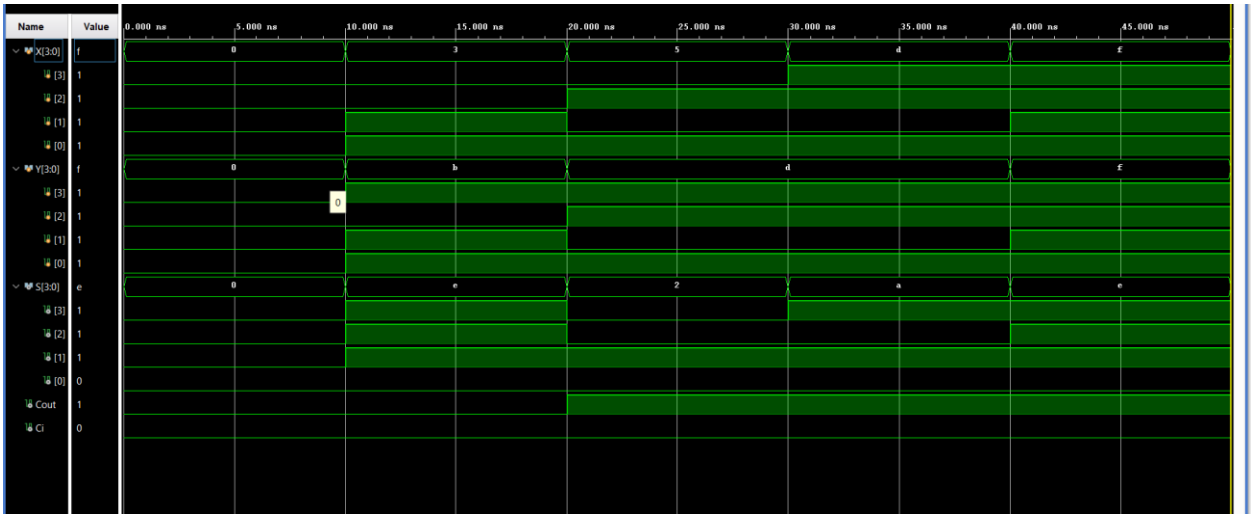
RTL Schematic



Technology Schematic



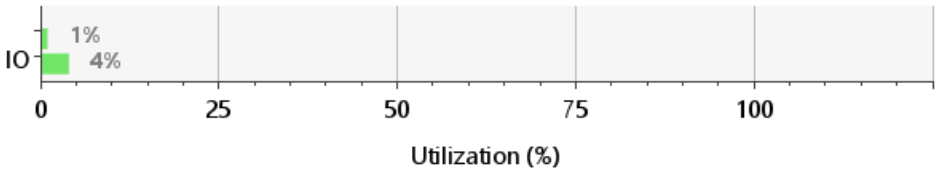
Behavioral Simulation Waveform



Design Summary

Summary

Resource	Utilization	Available	Utilization %
LUT	12	134600	0.01
IO	14	400	3.50



Combinational Delay

Combinational Delays

From Port	To Port	Ma 1	Max Process Corner	Min Delay	Min Process Corner
X[1]	S[2]	7.989	SLOW	2.603	FAST
Y[1]	S[2]	7.968	SLOW	2.583	FAST
X[1]	S[3]	7.875	SLOW	2.607	FAST
Y[1]	S[3]	7.855	SLOW	2.651	FAST
X[0]	S[2]	7.842	SLOW	2.680	FAST
X[0]	S[3]	7.728	SLOW	2.648	FAST
X[1]	Cout	7.691	SLOW	2.511	FAST
Y[1]	Cout	7.670	SLOW	2.555	FAST
Y[0]	S[2]	7.647	SLOW	2.770	FAST
X[0]	Cout	7.544	SLOW	2.552	FAST
Y[0]	S[3]	7.533	SLOW	2.738	FAST
Ci	S[2]	7.397	SLOW	2.882	FAST
Y[0]	Cout	7.349	SLOW	2.642	FAST
Ci	S[3]	7.284	SLOW	2.851	FAST
X[0]	S[1]	7.273	SLOW	2.427	FAST
X[2]	S[3]	7.157	SLOW	2.789	FAST
Ci	Cout	7.099	SLOW	2.754	FAST
Y[0]	S[1]	7.078	SLOW	2.511	FAST
Y[2]	S[3]	7.016	SLOW	2.630	FAST
X[2]	Cout	6.973	SLOW	2.693	FAST
Y[2]	Cout	6.831	SLOW	2.534	FAST
Ci	S[1]	6.817	SLOW	2.625	FAST
X[1]	S[1]	6.772	SLOW	2.623	FAST
Y[1]	S[1]	6.752	SLOW	2.603	FAST
X[3]	Cout	6.659	SLOW	2.305	FAST
X[0]	S[0]	6.647	SLOW	2.561	FAST
X[3]	S[3]	6.547	SLOW	2.529	FAST
Y[3]	Cout	6.526	SLOW	2.106	FAST
Y[0]	S[0]	6.452	SLOW	2.507	FAST

Y[3]	S[3]	6.413	SLOW	2.470	FAST
X[2]	S[2]	6.300	SLOW	2.448	FAST
Y[2]	S[2]	6.235	SLOW	2.409	FAST
Ci	S[0]	5.490	SLOW	2.090	FAST

Worst case for last FA's carry output is 7.691ns.

4-) Ripple Carry Adder with "generate for"

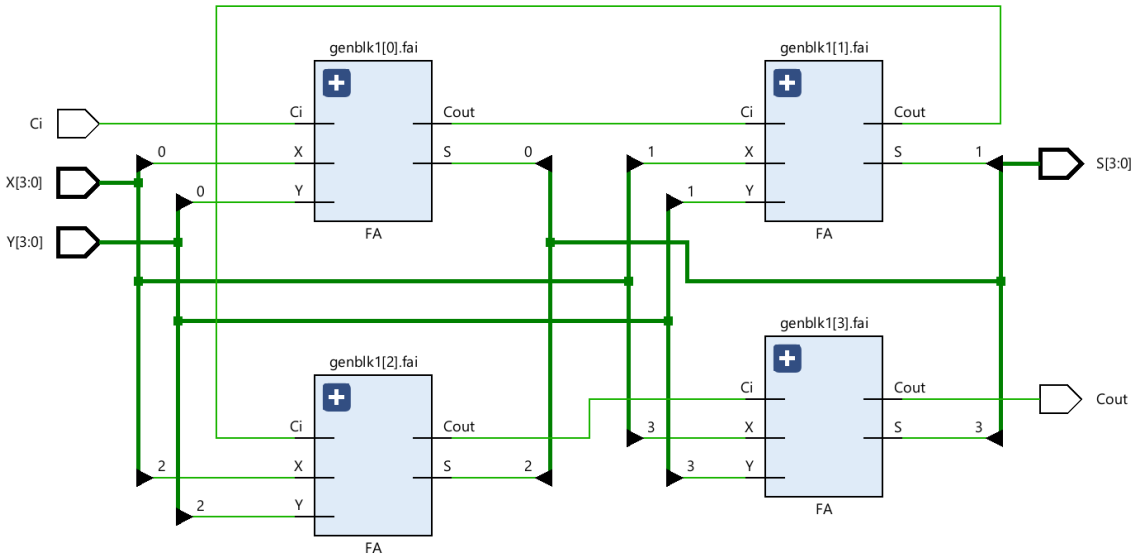
Verilog Code for Module

```
`timescale 1ns / 1ps
module parametric_RCA(X,Y,Ci,S,Cout);
parameter SIZE=4;

input [SIZE-1:0]X;
input [SIZE-1:0]Y;
input Ci;
output [SIZE-1:0]S;
output Cout;
wire OX[SIZE:0];
assign OX[0]=Ci;

genvar i;
generate
    for(i=0;i<SIZE;i=i+1)
    begin
        FA fai(.X(X[i]),.Y(Y[i]),.Ci(OX[i]),.Cout(OX[i+1]),.S(S[i]));
    end
    assign Cout=OX[SIZE];
endgenerate

endmodule
```

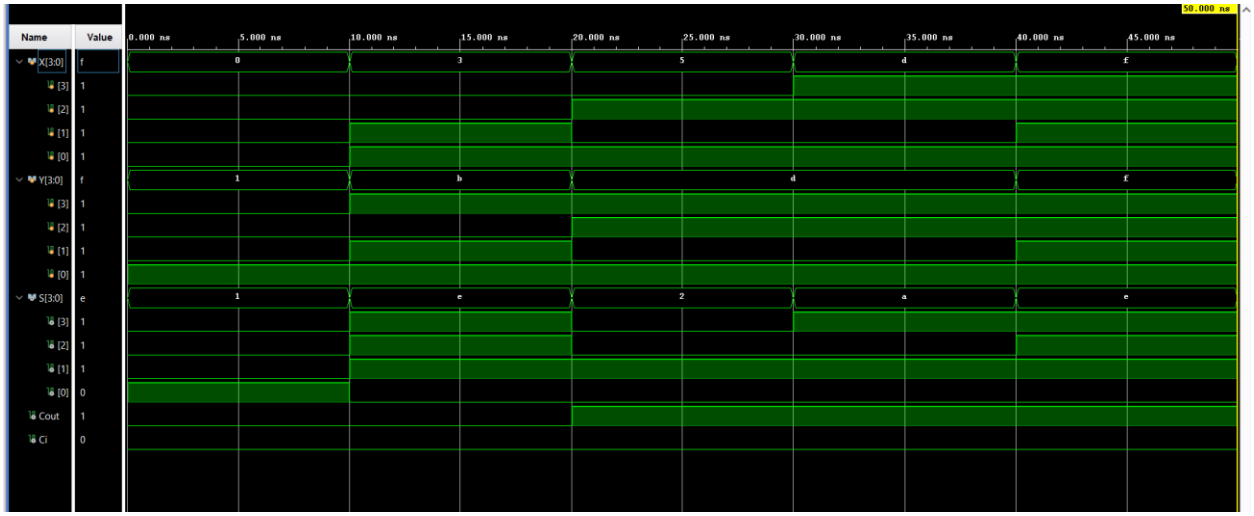
Verilog Code for 4-bit Testbench

```
`timescale 1ns / 1ps

module Top_module_tb;
    reg [3:0]X;
    reg [3:0]Y;
    wire [3:0]S;
    wire Cout;
    wire Ci;
    assign Ci=1'b0;


    parametric_RCA uut(.X(X),.Y(Y),.Ci(Ci),.Cout(Cout),.S(S));
    initial
    begin
        //inp = 1 sel = 00
        X=4'b0000;
        Y=4'b0001;
        #10
        X=4'b0011;
        Y=4'b1011;
        #10
        X=4'b0101;
        Y=4'b1101;
        #10
        X=4'b1101;
        Y=4'b1101;
        #10
        X=4'b1111;
        Y=4'b1111;
        #10
        $finish;
    end
endmodule
```

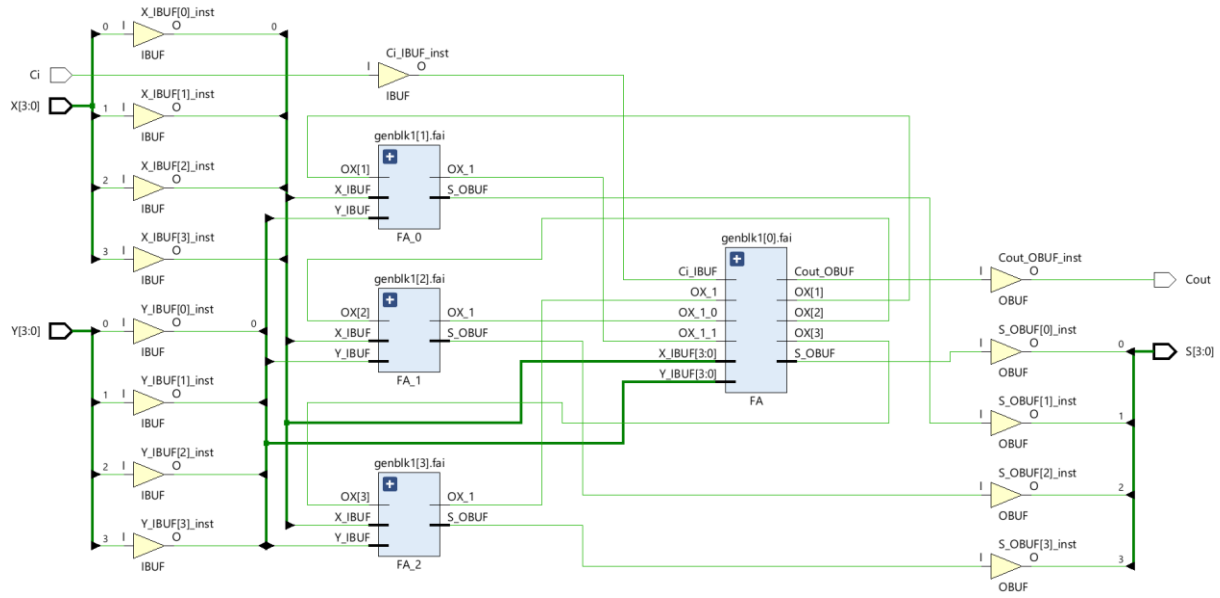
RTL Schematic for 4-bit Adder



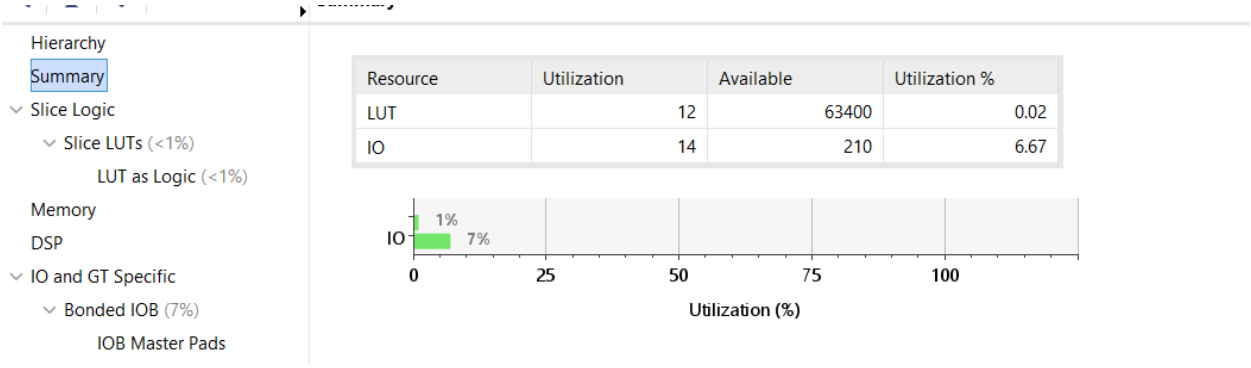
Simulation Wave Form for 4-bit Adder



Technology Schematic for 4-bit adder

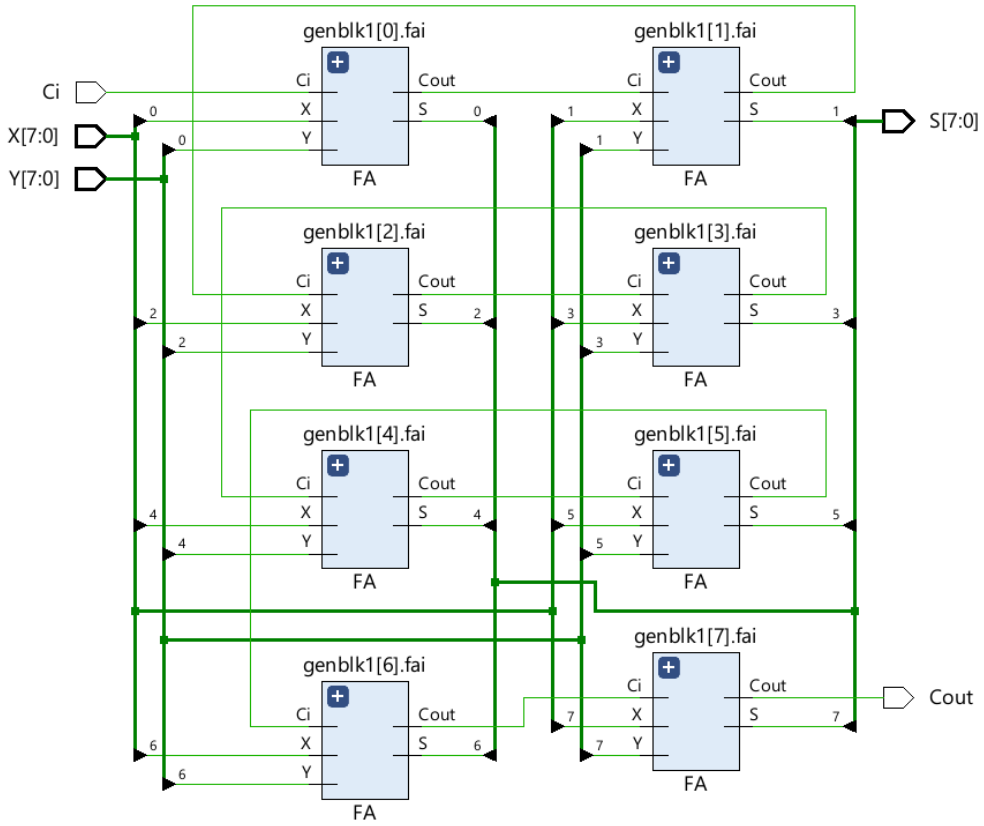


Design Summary for 4-bit Adder

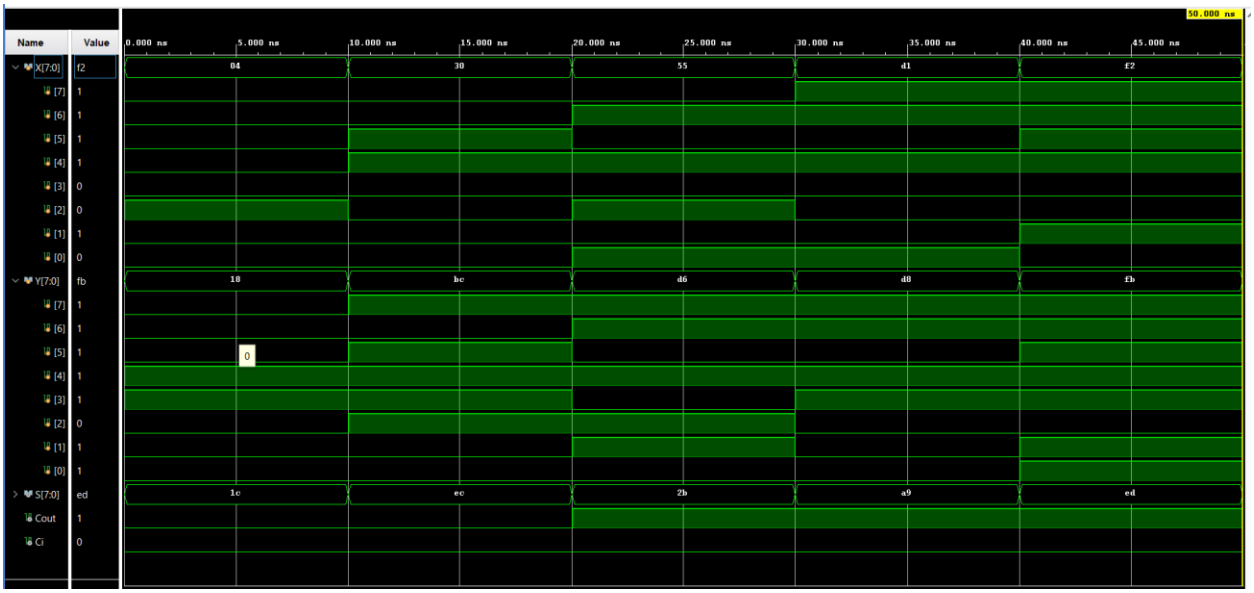


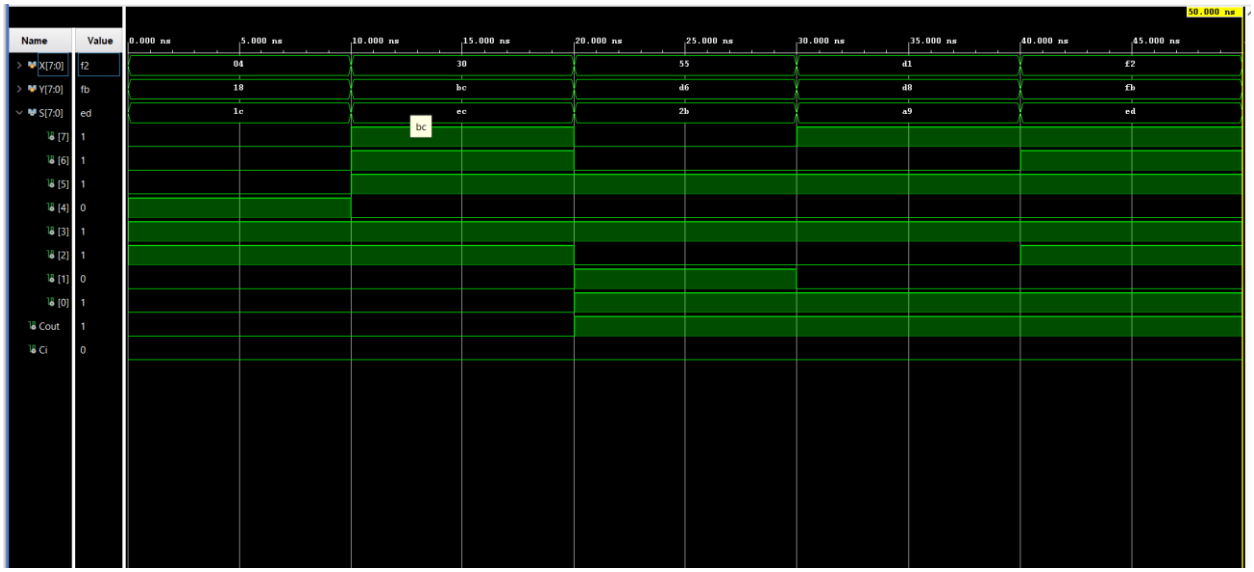
There is no difference between regular 4-bit adder and generate for loop 4-bit adder except naming of modules. Same for the technology schematic.

RTL Schematic for 8-bit Adder



Simulation Wave Form for 8-bit Adder



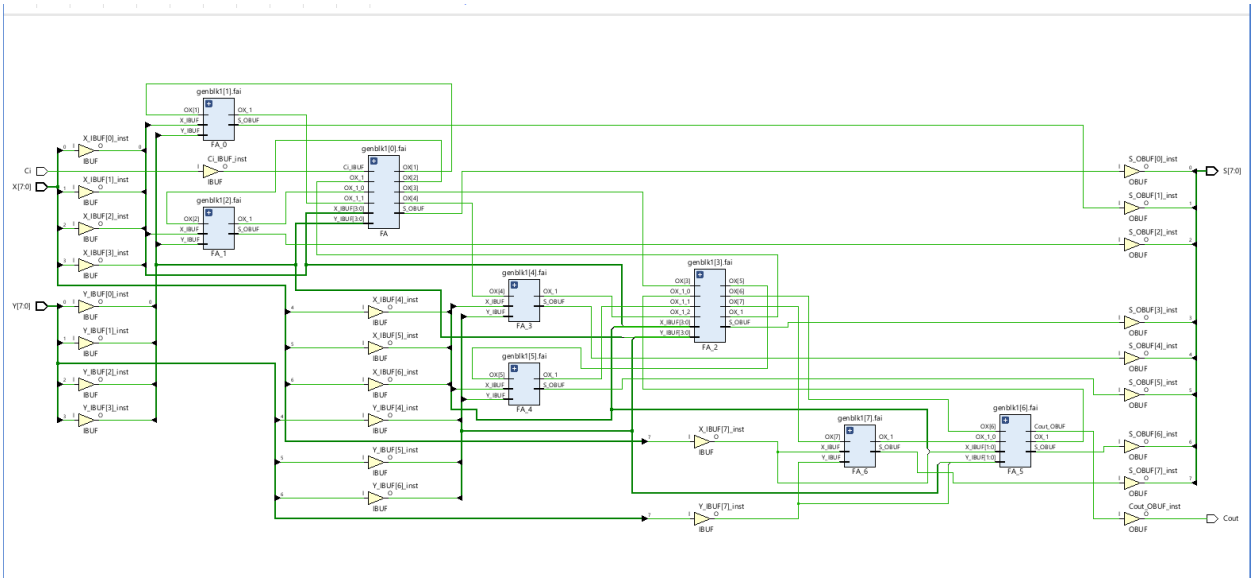


Verilog Code for Testbench 8-bit Adder

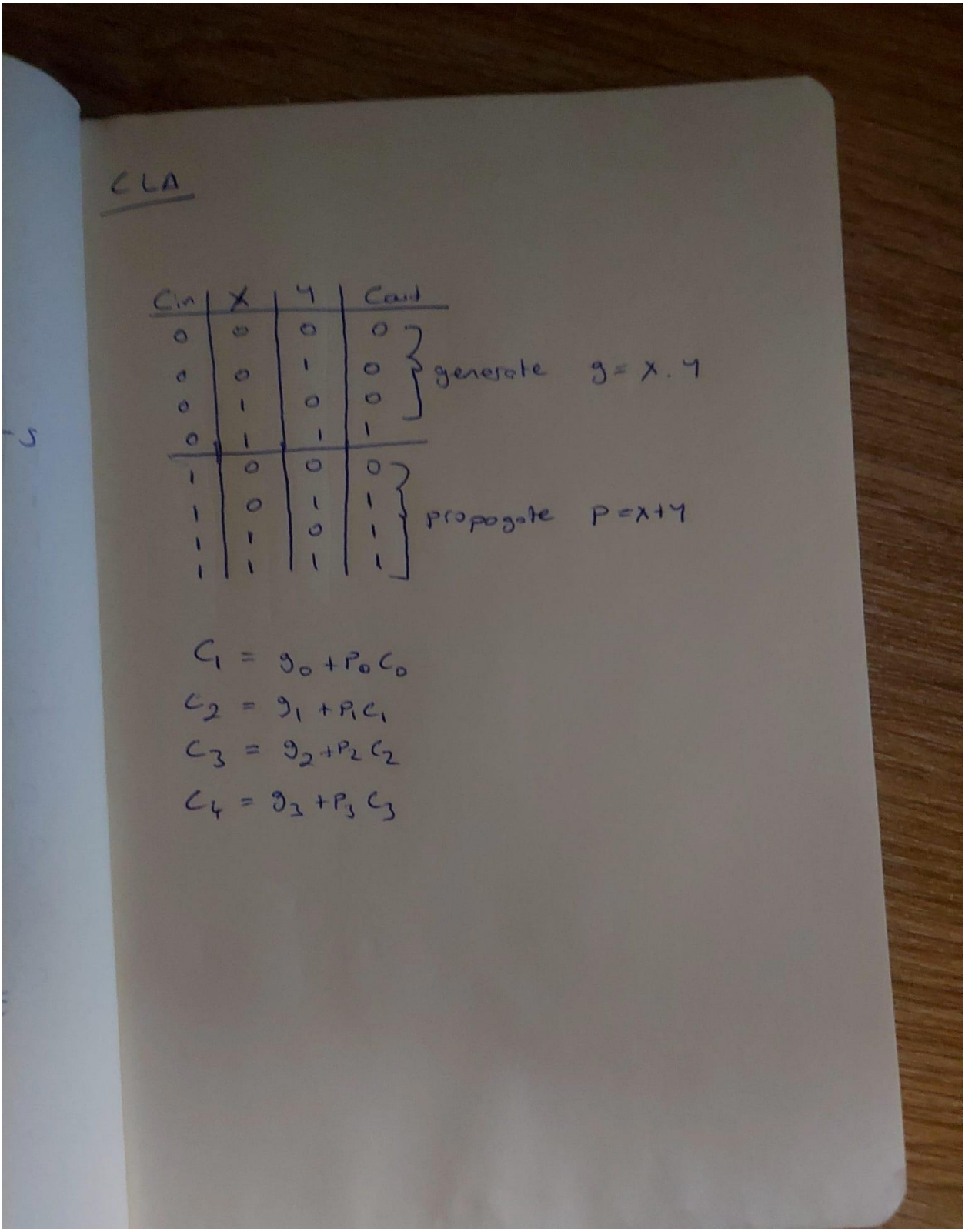
```
`timescale 1ns / 1ps

module Top_module_tb;
    reg [7:0]X;
    reg [7:0]Y;
    wire [7:0]S;
    wire Cout;
    wire Ci;
    assign Ci=1'b0;


    parametric_RCA uut(.X(X),.Y(Y),.Ci(Ci),.Cout(Cout),.S(S));
    initial
    begin
        //inp = 1 sel = 00
        X=8'b00000100;
        Y=8'b00011000;
        #10
        X=8'b00110000;
        Y=8'b10111100;
        #10
        X=8'b01010101;
        Y=8'b11010110;
        #10
        X=8'b11010001;
```

```
        Y=8'b11011000;
        #10
        X=8'b11110010;
        Y=8'b11111011;
        #10
        $finish;
    end
endmodule
```

Technology Schematic for 8-bit Adder

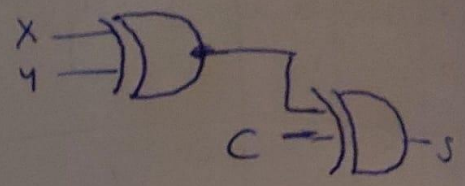


5-) CLA


CLA
Cin | X | Y | Cout
generate g = X . Y
propogate P = X + Y
C1 = g0 + P0 C0
C2 = g1 + P1 C1
C3 = g2 + P2 C2
C4 = g3 + P3 C3

CLA

C	X	Y	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



C \ XY	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$= \bar{C}\bar{X}Y + \bar{C}X\bar{Y} + C\bar{X}\bar{Y} + CXY$

$= \bar{C}(\bar{X}Y + X\bar{Y}) + C(\bar{X}\bar{Y} + XY)$

$A \oplus B \oplus C = ABC + \overbrace{\bar{A}BC + A\bar{B}C + AB\bar{C}}^{0} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C}$

$C \oplus X \oplus Y = CXY + \bar{C}\bar{X}Y + \bar{C}X\bar{Y} + C\bar{X}\bar{Y}$

$\Rightarrow$ it satisfies

Verilog Code

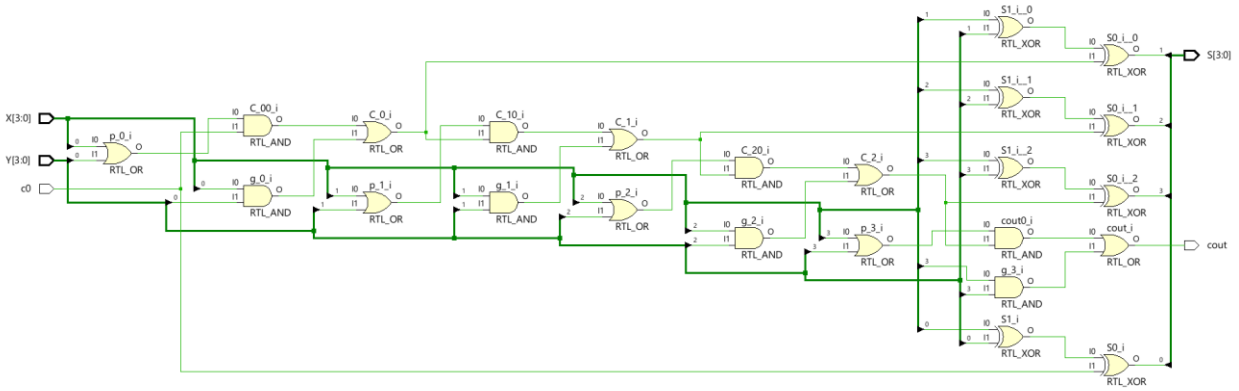
Module Code

```
`timescale 1ns / 1ps
module CLA(
    input [3:0]X,
    input [3:0]Y,
    input c0,
    output cout,
    output [3:0]S
);
wire [3:1]C;
wire [3:0]g;
wire [3:0]p;

assign S[0]=X[0]^Y[0]^c0;
assign g[0]=X[0]&Y[0];
assign p[0]=X[0]|Y[0];
assign C[1]=(p[0]&c0)|g[0];

assign S[1]=X[1]^Y[1]^C[1];
assign g[1]=X[1]&Y[1];
assign p[1]=X[1]|Y[1];
assign C[2]=(p[1]&C[1])|g[1];

assign S[2]=X[2]^Y[2]^C[2];
assign g[2]=X[2]&Y[2];
assign p[2]=X[2]|Y[2];
assign C[3]=(p[2]&C[2])|g[2];

assign S[3]=X[3]^Y[3]^C[3];
assign g[3]=X[3]&Y[3];
assign p[3]=X[3]|Y[3];
assign cout=(p[3]&C[3])|g[3];



endmodule
```

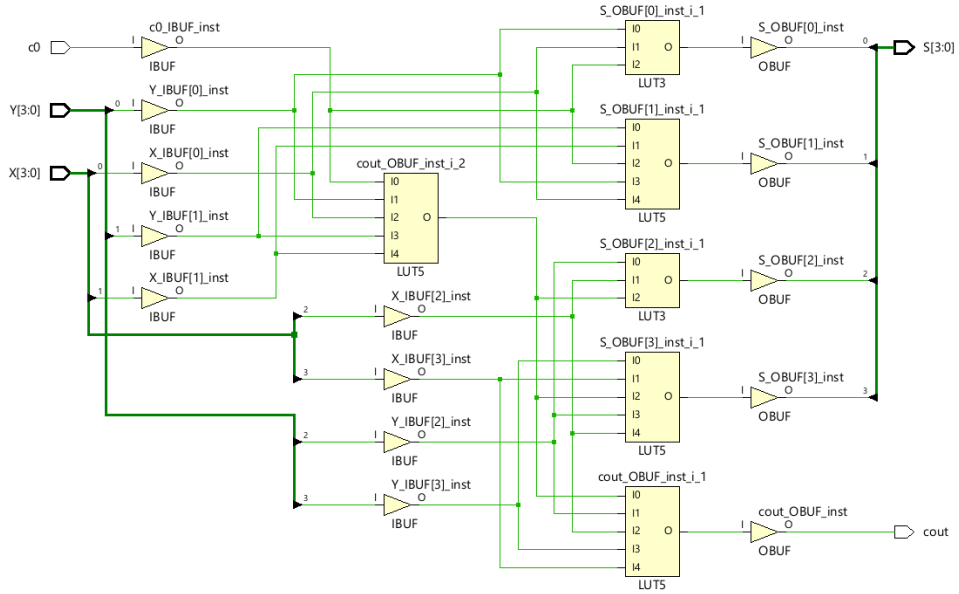
Testbench Code

```
`timescale 1ns / 1ps

module Top_module_tb;
    reg [3:0]X;
    reg [3:0]Y;
    wire [3:0]S;
    wire Cout;
    wire Ci;
    assign Ci=1'b0;


    CLA uut(.X(X),.Y(Y),.c0(Ci),.cout(Cout),.S(S));
    initial
    begin
        //inp = 1 sel = 00
        X=4'b0000;
        Y=4'b0001;
        #10
        X=4'b0011;
        Y=4'b1011;
        #10
        X=4'b0101;
        Y=4'b1101;
        #10
        X=4'b1101;
        Y=4'b1101;
        #10
        X=4'b1111;
        Y=4'b1111;
        #10
        $finish;
    end
endmodule
```

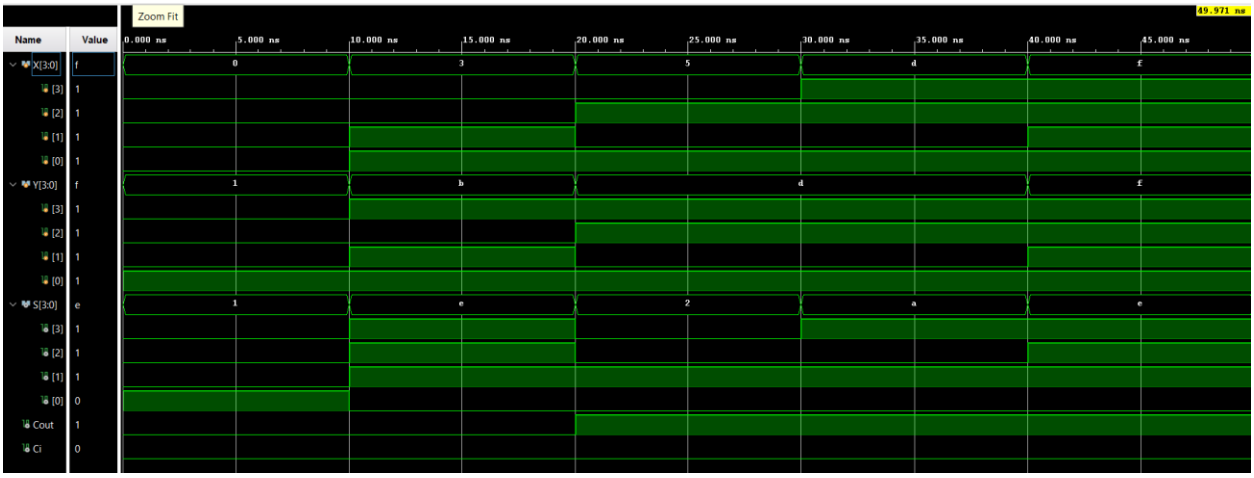
RTL Schematic



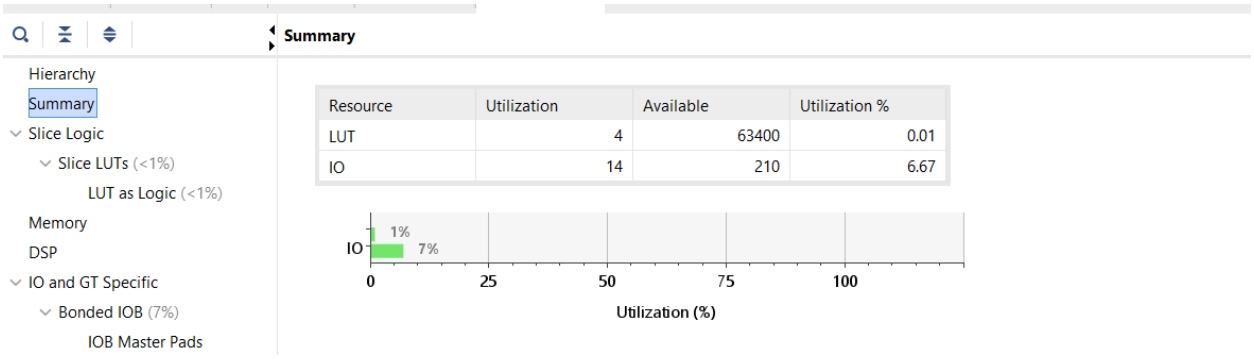
Technology Schematic



Simulation Wave Form



Design Summary



Combinational Delays

From Port	To Port	Ma 1	Max Process Corner	Min Delay	Min Process Corner
X[2]	cout	5.357	SLOW	2.072	FAST
Y[3]	cout	5.359	SLOW	2.071	FAST
X[3]	cout	5.361	SLOW	2.074	FAST
Y[2]	cout	5.361	SLOW	2.072	FAST
X[1]	cout	5.910	SLOW	2.321	FAST
Y[0]	cout	5.940	SLOW	2.317	FAST
X[0]	cout	5.942	SLOW	2.316	FAST
Y[1]	cout	5.944	SLOW	2.319	FAST
c0	cout	5.944	SLOW	2.317	FAST
X[0]	S[0]	6.221	SLOW	2.143	FAST
Y[0]	S[0]	6.221	SLOW	2.143	FAST
c0	S[0]	6.221	SLOW	2.143	FAST
X[0]	S[1]	6.236	SLOW	2.158	FAST
X[1]	S[1]	6.236	SLOW	2.158	FAST
Y[0]	S[1]	6.236	SLOW	2.158	FAST
Y[1]	S[1]	6.236	SLOW	2.158	FAST
c0	S[1]	6.236	SLOW	2.158	FAST
X[2]	S[3]	6.252	SLOW	2.173	FAST
X[3]	S[3]	6.252	SLOW	2.173	FAST
Y[2]	S[3]	6.252	SLOW	2.173	FAST
Y[3]	S[3]	6.252	SLOW	2.173	FAST
X[2]	S[2]	6.254	SLOW	2.175	FAST
Y[2]	S[2]	6.254	SLOW	2.175	FAST
X[1]	S[3]	6.835	SLOW	2.418	FAST
X[1]	S[2]	6.837	SLOW	2.420	FAST
Y[0]	S[3]	6.865	SLOW	2.414	FAST
X[0]	S[3]	6.867	SLOW	2.413	FAST
Y[0]	S[2]	6.867	SLOW	2.416	FAST

Y[1]	S[3]	6.869	SLOW	2.416	FAST
c0	S[3]	6.869	SLOW	2.414	FAST
X[0]	S[2]	6.869	SLOW	2.415	FAST
Y[1]	S[2]	6.871	SLOW	2.418	FAST
c0	S[2]	6.871	SLOW	2.416	FAST

Fastest parts of circuit are carry-out parts

6-) Adder Subtractor

Verilog Code

Module Verilog Code

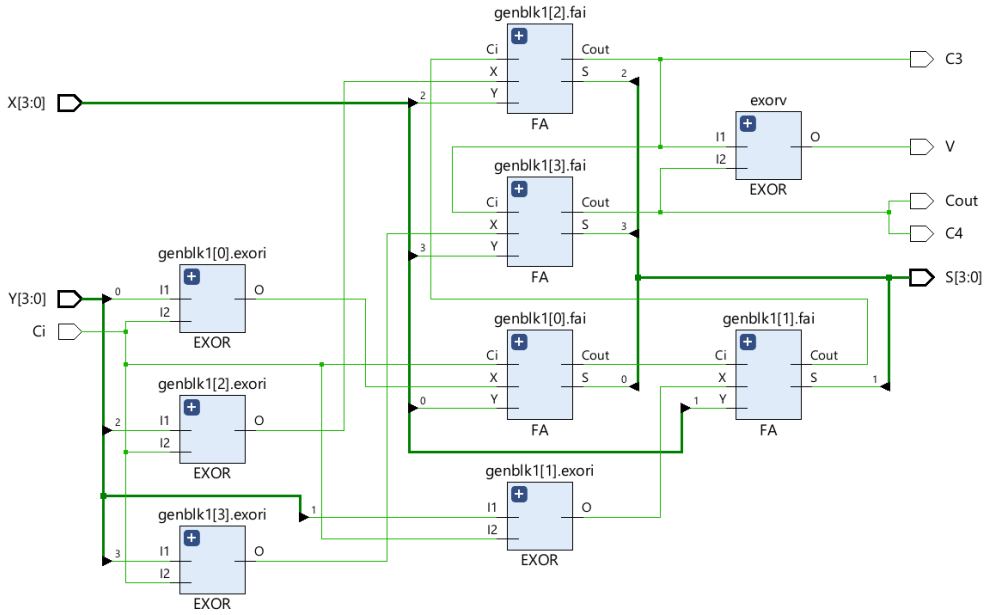
```
`timescale 1ns / 1ps
module ASV(
    input [3:0]X,
    input [3:0]Y,
    input Ci,
    output V,
    output [3:0]S,
    output Cout
);
wire [4:0]C;
wire [3:0]B;

assign C[0]=Ci;
genvar i;
generate
    for(i=0;i<4;i=i+1)
    begin
        EXOR exori(.I1(Y[i]),.I2(Ci),.O(B[i]));
        FA fai(.X(B[i]),.Y(X[i]),.Ci(C[i]),.S(S[i]),.Cout(C[i+1]));
    end
endgenerate
EXOR exorv(.I1(C[3]),.I2(C[4]),.O(V));
assign Cout=C[4];
endmodule
```

Testbench Verilog Code

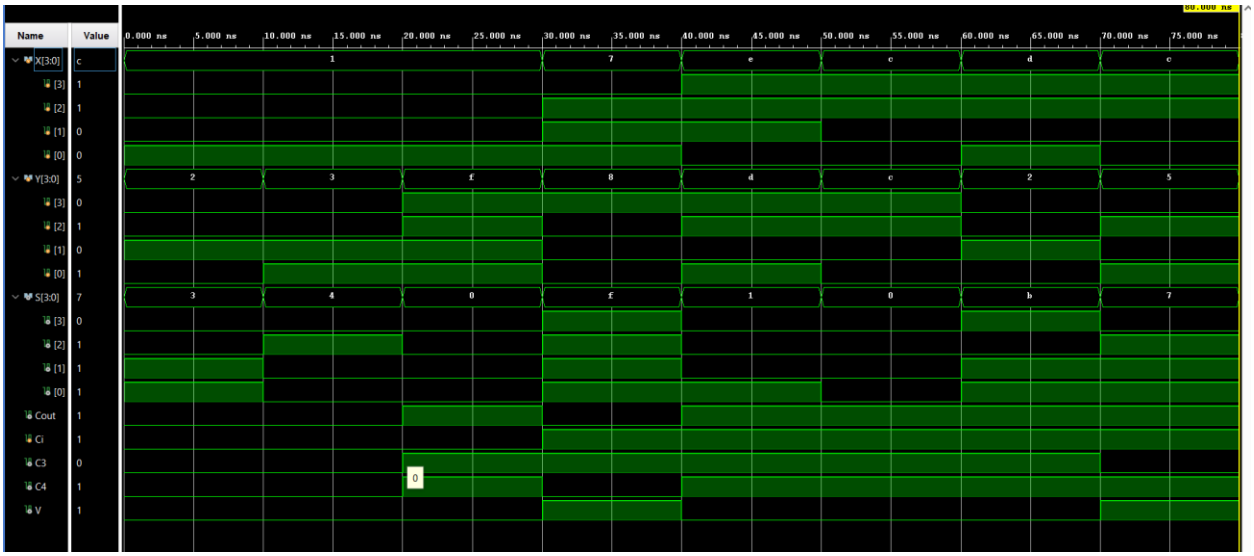
```
`timescale 1ns / 1ps

module Top_module_tb;
    reg [3:0]X;
    reg [3:0]Y;
    wire [3:0]S;
    wire Cout;
    reg Ci;
    wire C3;
    wire C4;
    wire V;


    ASV uut(.X(X),.Y(Y),.Ci(Ci),.V(V),.Cout(Cout),.S(S),.C3(C3),.C4(C4));

    initial
    begin
        Ci=1'b0;
        X=4'b0001;
        Y=4'b0010;
        #10
        X=4'b0001;
        Y=4'b0011;
        #10
        X=4'b0001;
        Y=4'b1111;
        #10
        Ci=1'b1;
        X=4'b0111;
        Y=4'b1000;
        #10
        X=4'b1110;
        Y=4'b1101;
        #10
        X=4'b1100;
        Y=4'b1100;
        #10
        X=4'b1101;
        Y=4'b0010;
        #10
        X=4'b1100;
        Y=4'b0101;
        #10
```

```
        $finish;
    end
endmodule
```

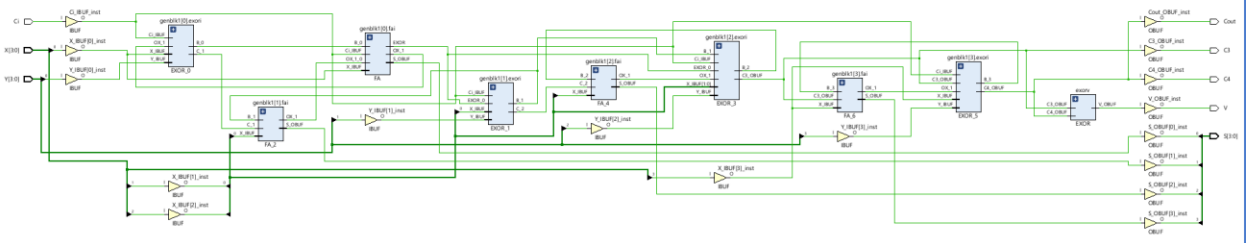
RTL Schematic



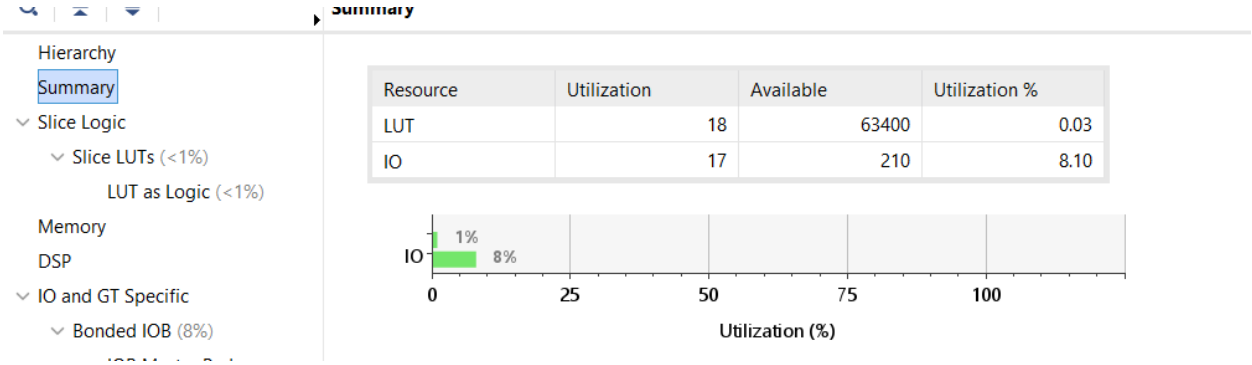
Behavioral Simulation Wave Form



Technology Schematic



Design Summary



Combinational Delays

From Port	To Port 1	Max Delay	Max Process Corner	Min Delay	Min Process Corner
Ci	C3	7.105	SLOW	2.322	FAST
X[0]	C3	6.986	SLOW	2.357	FAST
X[1]	C3	7.011	SLOW	2.120	FAST
X[2]	C3	6.403	SLOW	2.126	FAST
Y[0]	C3	7.592	SLOW	2.733	FAST
Y[1]	C3	7.620	SLOW	2.499	FAST
Y[2]	C3	6.996	SLOW	2.465	FAST
Ci	C4	7.702	SLOW	2.442	FAST
X[0]	C4	7.583	SLOW	2.601	FAST
X[1]	C4	7.608	SLOW	2.365	FAST
X[2]	C4	7.000	SLOW	2.370	FAST
X[3]	C4	6.395	SLOW	2.118	FAST
Y[0]	C4	8.189	SLOW	2.977	FAST
Y[1]	C4	8.217	SLOW	2.743	FAST
Y[2]	C4	7.593	SLOW	2.709	FAST
Y[3]	C4	7.010	SLOW	2.506	FAST
Ci	Cout	8.619	SLOW	2.543	FAST
X[0]	Cout	8.501	SLOW	2.703	FAST
X[1]	Cout	8.526	SLOW	2.466	FAST
X[2]	Cout	7.918	SLOW	2.472	FAST
X[3]	Cout	7.313	SLOW	2.219	FAST
Y[0]	Cout	9.107	SLOW	3.079	FAST
Y[1]	Cout	9.135	SLOW	2.845	FAST
Y[2]	Cout	8.511	SLOW	2.810	FAST
Y[3]	Cout	7.927	SLOW	2.608	FAST
Ci	S[0]	7.414	SLOW	2.154	FAST
X[0]	S[0]	7.319	SLOW	2.430	FAST
Y[0]	S[0]	7.925	SLOW	2.687	FAST
Ci	S[1]	8.002	SLOW	2.403	FAST

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
X[0]	S[1]	7.907	SLOW	2.438	FAST
X[1]	S[1]	7.299	SLOW	2.432	FAST
Y[0]	S[1]	8.513	SLOW	2.682	FAST
Y[1]	S[1]	7.907	SLOW	2.691	FAST
Ci	S[2]	8.596	SLOW	2.649	FAST
X[0]	S[2]	8.478	SLOW	2.694	FAST
X[1]	S[2]	8.503	SLOW	2.457	FAST
X[2]	S[2]	7.321	SLOW	2.454	FAST
Y[0]	S[2]	9.084	SLOW	3.070	FAST
Y[1]	S[2]	9.112	SLOW	2.704	FAST
Y[2]	S[2]	7.914	SLOW	2.702	FAST
Ci	S[3]	8.618	SLOW	2.649	FAST
X[0]	S[3]	8.500	SLOW	2.702	FAST
X[1]	S[3]	8.525	SLOW	2.465	FAST
X[2]	S[3]	7.916	SLOW	2.470	FAST
X[3]	S[3]	7.311	SLOW	2.444	FAST
Y[0]	S[3]	9.106	SLOW	3.078	FAST
Y[1]	S[3]	9.134	SLOW	2.844	FAST
Y[2]	S[3]	8.509	SLOW	2.809	FAST
Y[3]	S[3]	7.926	SLOW	2.714	FAST
Ci	V	9.211	SLOW	2.656	FAST
X[0]	V	9.092	SLOW	2.690	FAST
X[1]	V	9.117	SLOW	2.453	FAST
X[2]	V	8.509	SLOW	2.459	FAST
X[3]	V	7.904	SLOW	2.461	FAST
Y[0]	V	9.698	SLOW	3.066	FAST
Y[1]	V	9.726	SLOW	2.832	FAST
Y[2]	V	9.102	SLOW	2.798	FAST
Y[3]	V	8.518	SLOW	2.850	FAST

Worst delay is from Y[1] to V and it is 9.726ns