DIGITAL SYSTEM DESIGN APPLICATION – EHB 436E

Experiment IV

Yiğit Bektaş GÜRSOY

040180063

Class Lecturer: Sıddıka Berna Örs Yalçın

Class Assistant:
Serdar Duran
Yasin Fırat Kula
Mehmet Onur Demirtürk

1. HALF ADDER

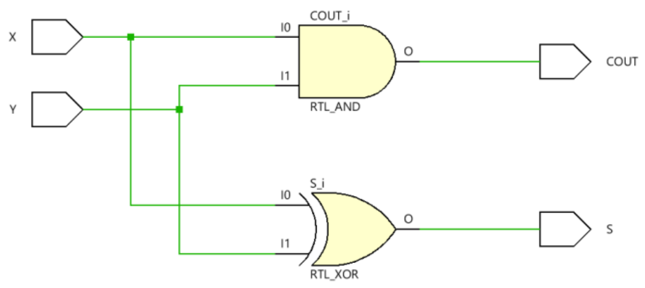
- Verilog Code

```
module HA(
    input X,
    input Y,

    output COUT,
    output S
    );

    assign COUT = X & Y; //CARRY
    assign S = X ^ Y ; //SUM

endmodule
```

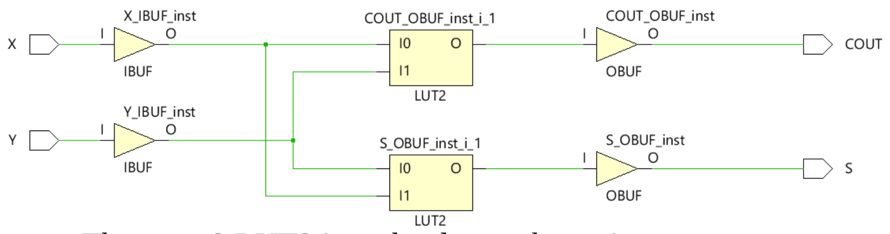
- Testbench Code

```
  //HALF ADDER TEST BENCH
module HA_tb();
    reg X;
    reg Y;
    wire COUT;
    wire SUM;
    HA DUT(.X(X),
           .Y(Y),
           .COUT(COUT),
           .S(SUM)
           );
    initial
    begin
        X = 1'b0 ; Y = 1'b0;
        #10 X = 1'b0 ; Y = 1'b1;
        #10 X = 1'b1 ; Y = 1'b0;
        #10 X = 1'b1 ; Y = 1'b1;
        #10 $finish;
    end
    endmodule
```

- RTL Schematic

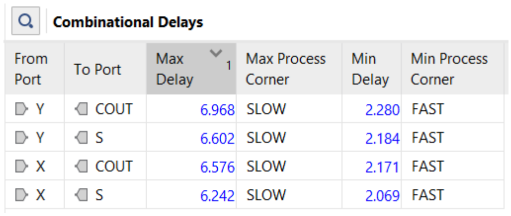


- Technology Schematic



  - There are 2 LUT2 in technology schematic.

- Combinational Delay



Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
Y	COUT	6.968	SLOW	2.280	FAST
Y	S	6.602	SLOW	2.184	FAST
X	COUT	6.576	SLOW	2.171	FAST
X	S	6.242	SLOW	2.069	FAST

  - The maximum delay in my circuit is 6.968ns.

2. FULL ADDER

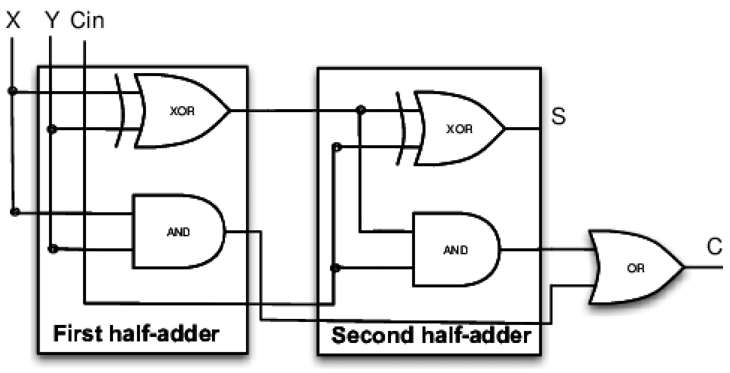
- Verilog Code

```
module FA(
    input X,
    input Y,
    input CIN,

    output COUT,
    output S
    );
    wire signal_1;
    wire signal_2;
    wire signal_3;

    HA halfadder1 ( .X(X),
                    .Y(Y),

                    .COUT(signal_1),
                    .S(signal_2)
                    );
    HA halfadder2 ( .X(signal_2),
                    .Y(CIN),

                    .COUT(signal_3),
                    .S(S)
                    );
    assign COUT = signal_3 | signal_1;
endmodule
```

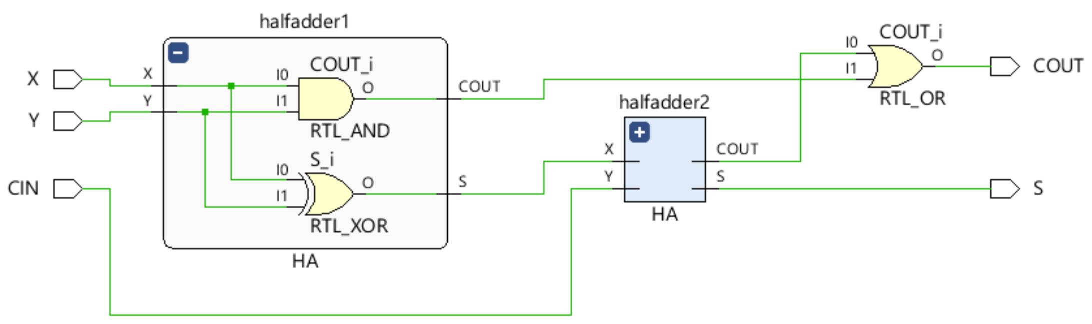
- Testbench Code

```
    //FULL ADDER TEST BENCH
module FA_tb();
    reg X;
    reg Y;
    reg CIN;

    wire COUT;
    wire SUM;
    FA DUT(.X(X),
           .Y(Y),
           .CIN(CIN),
           .COUT(COUT),
           .S(SUM)
           );
    initial
        begin
        X = 1'b0 ; Y = 1'b0; CIN = 1'b0;
        #10 X = 1'b0 ; Y = 1'b0; CIN = 1'b1;
        #10 X = 1'b0 ; Y = 1'b1; CIN = 1'b0;
        #10 X = 1'b0 ; Y = 1'b1; CIN = 1'b1;
        #10 X = 1'b1 ; Y = 1'b0; CIN = 1'b0;
        #10 X = 1'b1 ; Y = 1'b0; CIN = 1'b1;
        #10 X = 1'b1 ; Y = 1'b1; CIN = 1'b0;
        #10 X = 1'b1 ; Y = 1'b1; CIN = 1'b1;
        #10 $finish;
        end
    endmodule
```

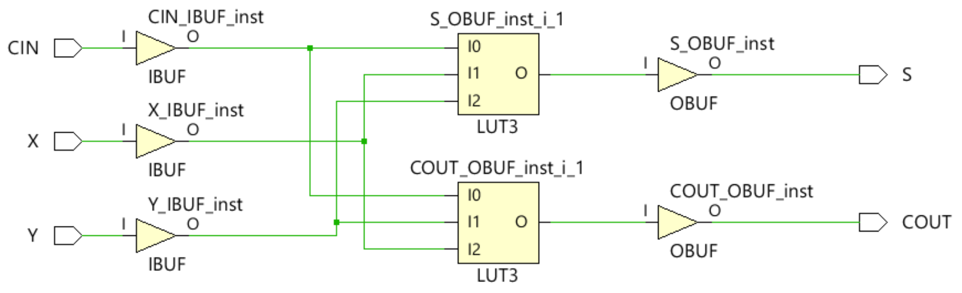
- Full-adder Circuit



- RTL Schematic

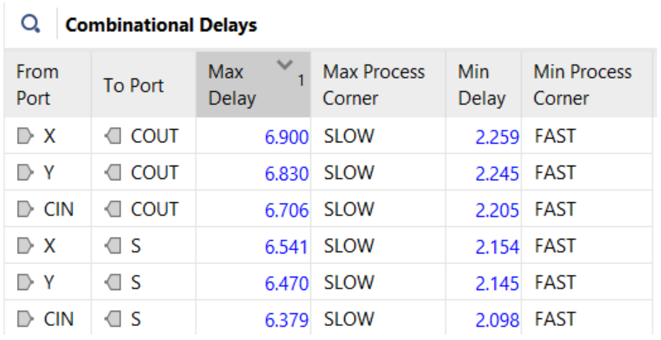


- Technology Schematic



  - There are 2 LUT3 in technology schematic.

- Combinational Delay



Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
X	COUT	6.900	SLOW	2.259	FAST
Y	COUT	6.830	SLOW	2.245	FAST
CIN	COUT	6.706	SLOW	2.205	FAST
X	S	6.541	SLOW	2.154	FAST
Y	S	6.470	SLOW	2.145	FAST
CIN	S	6.379	SLOW	2.098	FAST

  - The maximum delay in my circuit is 6.9ns.

3. RIPPLE CARRY ADDER

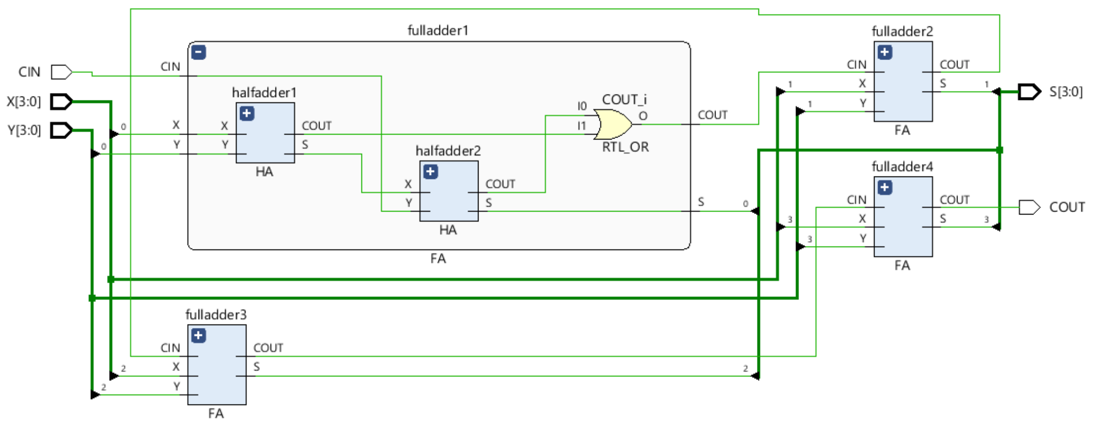
- Verilog Code

```
module RCA(
    input [3:0]X,
    input [3:0]Y,
    input CIN,

    output COUT,
    output [3:0]S
    );
    wire c1,c2,c3,c4;

    FA fulladder1(
        .X(X[0]),
        .Y(Y[0]),
        .CIN(CIN),

        .COUT(c1),
        .S(S[0])
        );

    FA fulladder2(
        .X(X[1]),
        .Y(Y[1]),
        .CIN(c1),

        .COUT(c2),
        .S(S[1])
        );

    FA fulladder3(
        .X(X[2]),
        .Y(Y[2]),
        .CIN(c2),

        .COUT(c3),
        .S(S[2])
        );

    FA fulladder4(
        .X(X[3]),
        .Y(Y[3]),
        .CIN(c3),

        .COUT(COUT),
        .S(S[3])
        );

endmodule
```

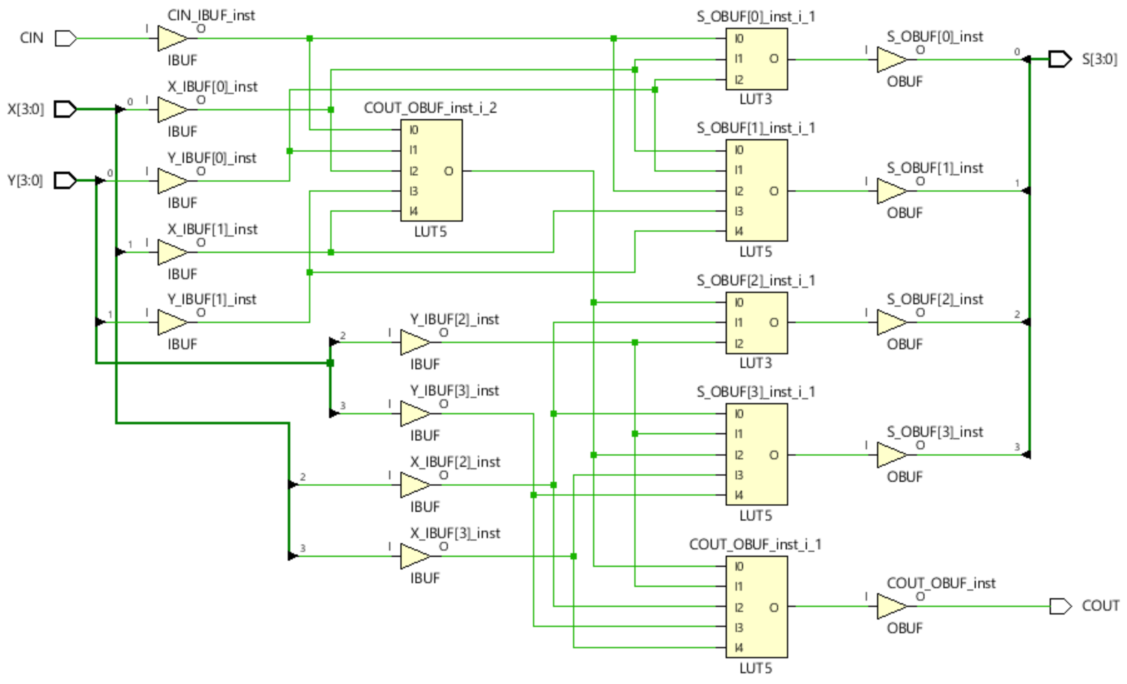
- Testbench Code

```
    //RIPPLE CARRY ADDER TEST BENCH
module RCA_tb();
    reg [3:0]X;
    reg [3:0]Y;
    reg CIN;

    wire COUT;
    wire [3:0]SUM;

    RCA DUT(.X(X),
            .Y(Y),
            .CIN(CIN),
            .COUT(COUT),
            .S(SUM)
            );
    initial
        begin
        X = 4'b0000 ; Y = 4'b0000; CIN = 1'b0;
        #10 X = 4'b1110 ; Y = 4'b0011; CIN = 1'b1;
        #10 X = 4'b0110 ; Y = 4'b1100; CIN = 1'b0;
        #10 X = 4'b0011 ; Y = 4'b1010; CIN = 1'b1;
        #10 X = 4'b1000 ; Y = 4'b1111; CIN = 1'b0;
        #10 X = 4'b1111 ; Y = 4'b0101; CIN = 1'b1;
        #10 X = 4'b1001 ; Y = 4'b1100; CIN = 1'b0;
        #10 X = 4'b1101 ; Y = 4'b1111; CIN = 1'b1;
        #10 $finish;
        end
    endmodule
```

- RTL Schematic

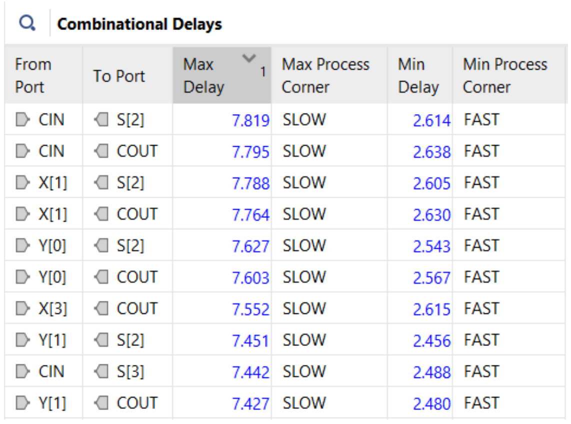


- Technology Schematic



  - There are 4 LUT5 and 2 LUT3 in technology schematic.

- Combinational Delay


Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
CIN	S[2]	7.819	SLOW	2.614	FAST
CIN	COUT	7.795	SLOW	2.638	FAST
X[1]	S[2]	7.788	SLOW	2.605	FAST
X[1]	COUT	7.764	SLOW	2.630	FAST
Y[0]	S[2]	7.627	SLOW	2.543	FAST
Y[0]	COUT	7.603	SLOW	2.567	FAST
X[3]	COUT	7.552	SLOW	2.615	FAST
Y[1]	S[2]	7.451	SLOW	2.456	FAST
CIN	S[3]	7.442	SLOW	2.488	FAST
Y[1]	COUT	7.427	SLOW	2.480	FAST

  - The maximum delay in my circuit is 7.819ns.

4. RIPPLE CARRY ADDER WITH GENERATE FOR

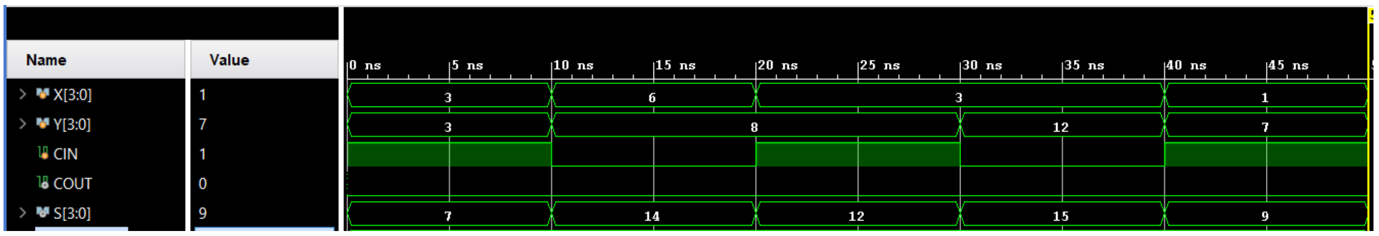
- Verilog Code

```
module parametric_RCA #( parameter
SIZE=4 )
    (
    input [SIZE-1:0]X,
    input [SIZE-1:0]Y,
    input CIN,

    output COUT,
    output [SIZE-1:0]S
    );

    wire [SIZE:0]signal;

    assign signal[0] = CIN;

    genvar i;
    generate
    for(i = 0; i < SIZE; i = i + 1)
        begin: generated_FA

        FA fulladder(
            X[i],
            Y[i],
            signal[i],

            signal[i+1],
            S[i]
            );
        end
    endgenerate

    assign COUT = signal[SIZE];
endmodule
```

- 4-bit wide Testbench Code

```
module parametric_RCA_tb();
    parameter SIZE = 4 ;

    reg [SIZE-1:0]X;
    reg [SIZE-1:0]Y;
    reg CIN;

    wire COUT;
    wire [SIZE-1:0]S;

    parametric_RCA #(
                    .SIZE(SIZE)
                    )
    RCA1(.X(X),
        .Y(Y),
        .CIN(CIN),

        .COUT(COUT),
        .S(S));

    initial
        begin
        X = 4'b0011 ; Y = 4'b0011; CIN = 1'b1;
        #10 X = 4'b0110 ; Y = 4'b1000; CIN = 1'b0;
        #10 X = 4'b0011 ; Y = 4'b1000; CIN = 1'b1;
        #10 X = 4'b0011 ; Y = 4'b1100; CIN = 1'b0;
        #10 X = 4'b0001 ; Y = 4'b0111; CIN = 1'b1;
        #10 $finish;
        end
endmodule
```

- 4-bit wide simulation result

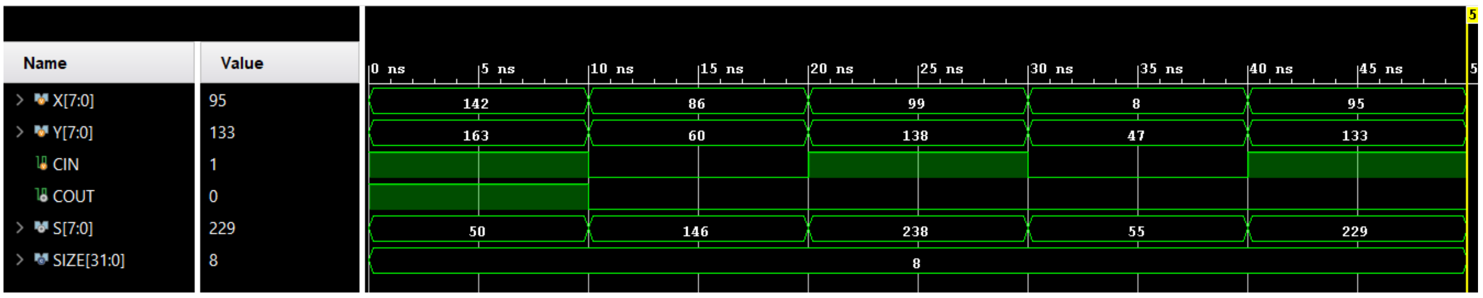


```
X[3:0]   1   3 | 6 | 3 | 1
Y[3:0]   7   3 | 8 | 12 | 7
CIN      1
COUT     0
S[3:0]   9   7 | 14 | 12 | 15 | 9
```

- 8-bit wide testbench code

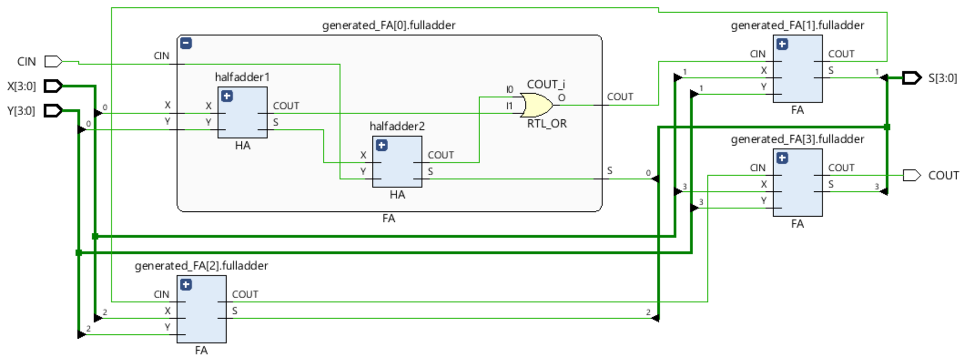
```
module parametric_RCA_tb();
    parameter SIZE = 4 ;

    reg [SIZE-1:0]X;
    reg [SIZE-1:0]Y;
    reg CIN;

    wire COUT;
    wire [SIZE-1:0]S;

    parametric_RCA #(
                    .SIZE(SIZE)
                    )
    RCA1(.X(X),
        .Y(Y),
        .CIN(CIN),

        .COUT(COUT),
        .S(S));

    initial
        begin
        X = 8'b10001110  ; Y = 8'b10100011; CIN = 1'b1;
        #10 X = 8'b01010110 ; Y = 8'b00111100; CIN = 1'b0;
        #10 X = 8'b01100011 ; Y = 8'b10001010; CIN = 1'b1;
        #10 X = 8'b00001000 ; Y = 8'b00101111; CIN = 1'b0;
        #10 X = 8'b01011111 ; Y = 8'b10000101; CIN = 1'b1;
        #10 $finish;
        end
endmodule
```

- 8-bit wide simulation result

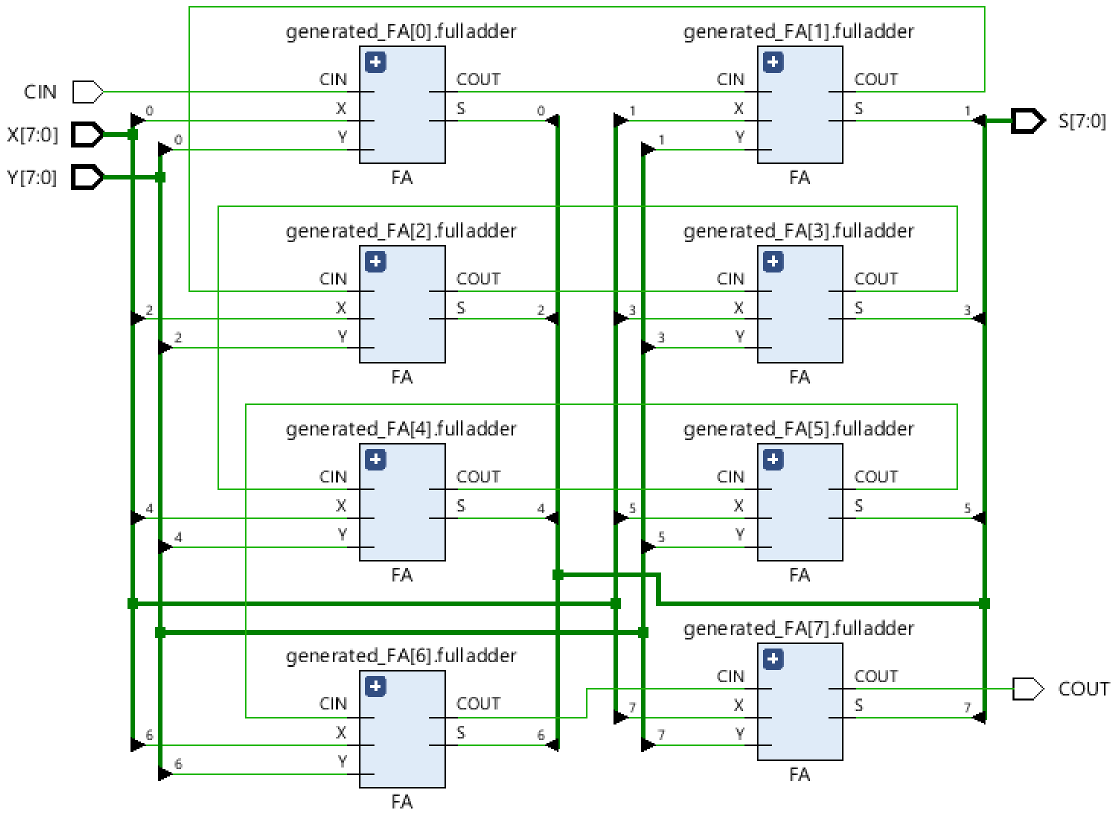


```
X[7:0]      95    142 | 86 | 99 | 8 | 95
Y[7:0]      133   163 | 60 | 138 | 47 | 133
CIN         1
COUT        0
S[7:0]      229   50 | 146 | 238 | 55 | 229
SIZE[31:0]  8     8
```

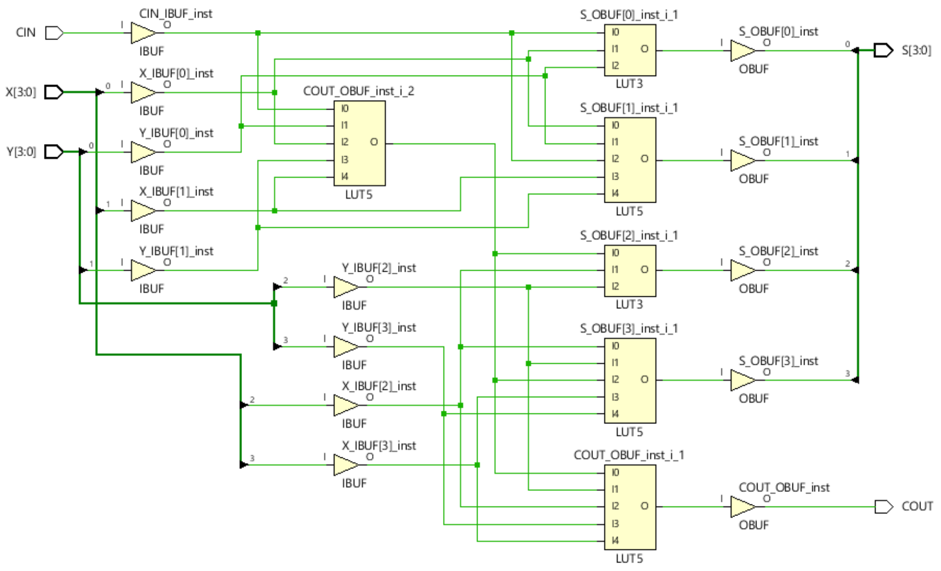
- RTL Schematic
  - 4-bit wide RCA with generate for


generated_FA[0].fulladder
halfadder1
halfadder2
HA
FA
RTL_OR
COUT_i
generated_FA[1].fulladder
generated_FA[2].fulladder
generated_FA[3].fulladder
CIN
X[3:0]
Y[3:0]
S[3:0]
COUT

  - 8-bit wide RCA with generate for

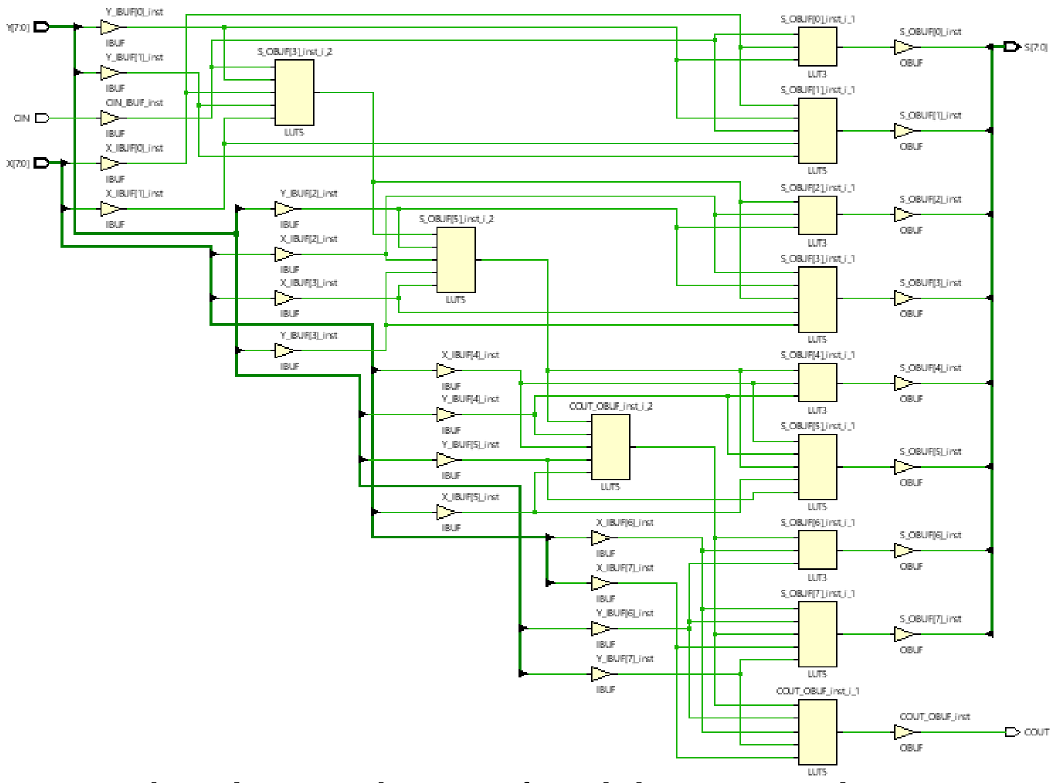

generated_FA[0].fulladder
generated_FA[1].fulladder
generated_FA[2].fulladder
generated_FA[3].fulladder
generated_FA[4].fulladder
generated_FA[5].fulladder
generated_FA[6].fulladder
generated_FA[7].fulladder
CIN
X[7:0]
Y[7:0]
S[7:0]
COUT
FA

- Technology Schematic
  - 4-bit wide RCA with generate for



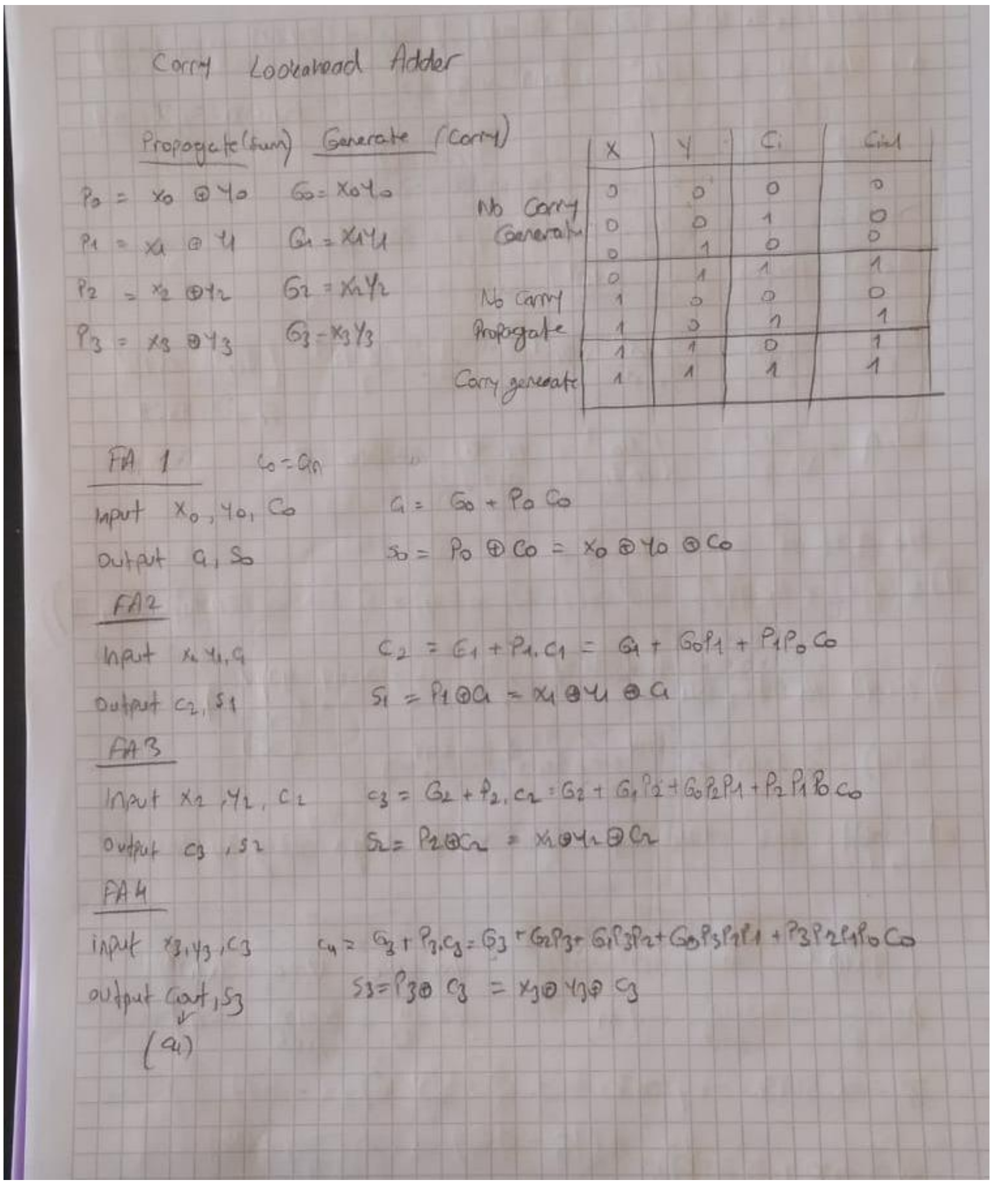
4-bit wide RCA with generate for includes 4 LUT5s and 2 LUT3.

  - 8-bit wide RCA with generate for



8-bit wide RCA with generate for includes 8 LUT5s and 3 LUT3.
Our 4-bit RCA circuits are very similar to each other. In our 8-bit RCA circuit, the use of LUT5 and LUT3 is more.

5. CARRY LOOKAHEAD ADDER



Carry Lookahead Adder

Propogate(Sum) Generate (Carry)

Propogate(Sum)	Generate (Carry)
$P_0 = x_0 \oplus y_0$	$G_0 = x_0 y_0$
$P_1 = x_1 \oplus y_1$	$G_1 = x_1 y_1$
$P_2 = x_2 \oplus y_2$	$G_2 = x_2 y_2$
$P_3 = x_3 \oplus y_3$	$G_3 - x_3 y_3$

	X	Y	C_i	C_{i+1}
No Carry Generate	0	0	0	0
	0	0	1	0
	0	1	0	0
No Carry Propogate	0	1	1	1
	1	0	0	0
	1	0	1	1
Carry generate	1	1	0	1
	1	1	1	1

FA 1 $C_0 = C_{in}$

input x_0, y_0, C_0 $C_1 = G_0 + P_0 C_0$

output C_1, S_0 $S_0 = P_0 \oplus C_0 = x_0 \oplus y_0 \oplus C_0$

FA2

input $x_1 y_1, C_1$ $C_2 = G_1 + P_1 . C_1 = G_1 + G_0 P_1 + P_1 P_0 C_0$

output C_2, S_1 $S_1 = P_1 \oplus C_1 = x_1 \oplus y_1 \oplus C_1$

FA3

input x_2, y_2, C_2 $C_3 = G_2 + P_2 . C_2 = G_2 + G_1 P_2 + G_0 P_2 P_1 + P_2 P_1 P_0 C_0$

output C_3, S_2 $S_2 = P_2 \oplus C_2 = x_2 \oplus y_2 \oplus C_2$

FA4

input x_3, y_3, C_3 $C_4 = G_3 + P_3 . C_3 = G_3 + G_2 P_3 + G_1 P_3 P_2 + G_0 P_3 P_2 P_1 + P_3 P_2 P_1 P_0 C_0$

output C_{out}, S_3 $S_3 = P_3 \oplus C_3 = x_3 \oplus y_3 \oplus C_3$

(C_4)

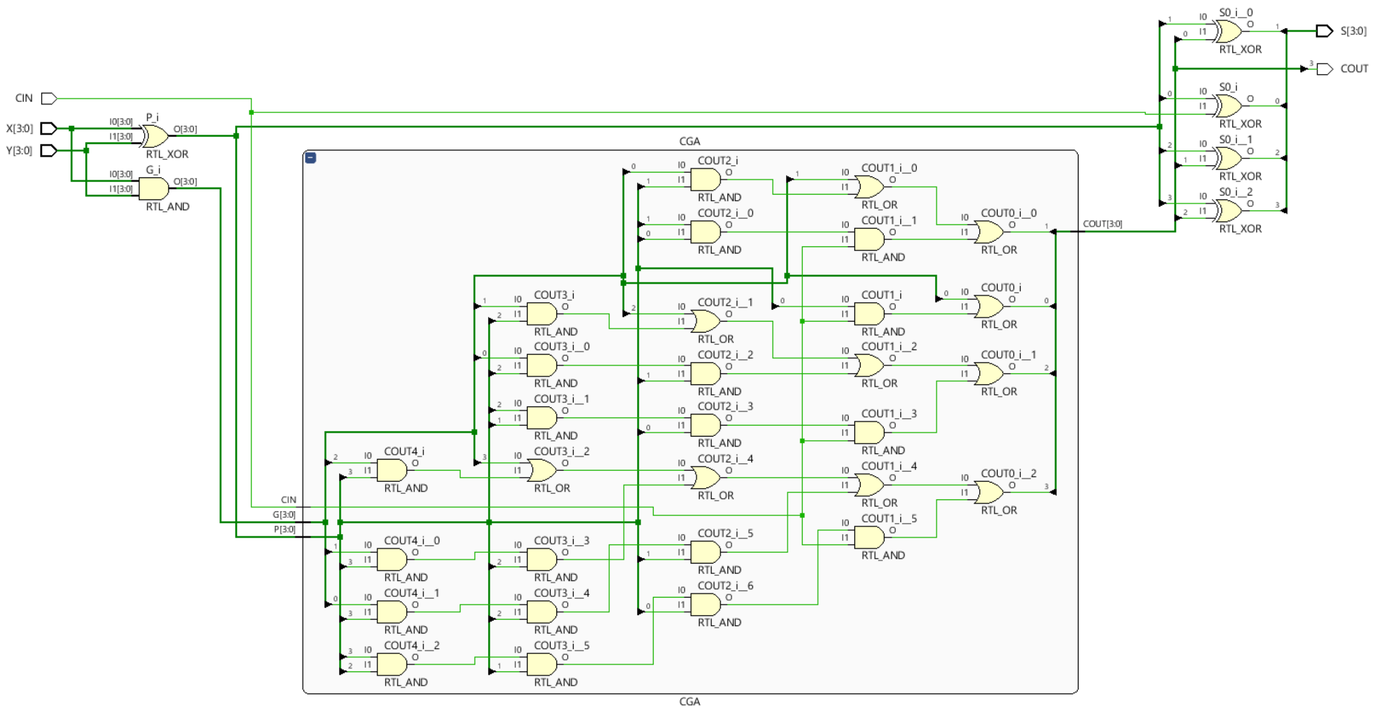
- Verilog Code

```
module CGA ( //carry , generate , propagate
            input [3:0]P,
            input [3:0]G,
            input CIN,

            output [3:0]COUT
            );

    assign COUT[0] = G[0] | (P[0] & CIN);
    assign COUT[1] = G[1] | (G[0] & P[1]) | (P[1] & P[0] & CIN);
    assign COUT[2] = G[2] | (G[1] & P[2]) | (G[0] & P[2] & P[1]) | ( P[2] & P[1] & P[0] & CIN);
    assign COUT[3] = G[3] | (G[2] & P[3]) | (G[1] & P[3] & P[2]) | (G[0] & P[3] & P[2] & P[1]) | (P[3] &
P[2] & P[1] & P[0] & CIN);
endmodule

module CLA (
            input [3:0]X,
            input [3:0]Y,
            input CIN,

            output COUT,
            output [3:0]S
            );

    wire [3:0]P;
    wire [3:0]G;
    wire [3:0]C;

    assign P = X ^ Y;
    assign G = X & Y;

    CGA CGA(
            .P(P),
            .G(G),

            .CIN(CIN),
            .COUT(C)
            );

    assign COUT = C[3];

    assign S[0] = P[0] ^ CIN;
    assign S[1] = P[1] ^ C[0];
    assign S[2] = P[2] ^ C[1];
    assign S[3] = P[3] ^ C[2];

endmodule
```

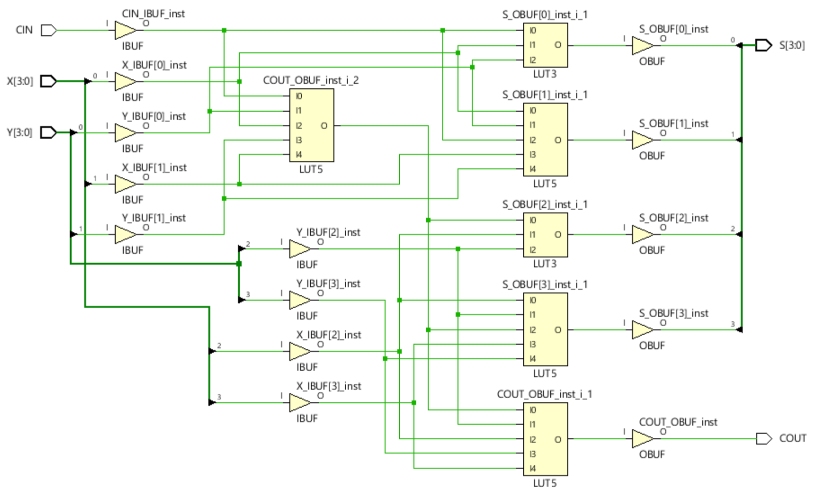
- Testbench Code

```
    //CLA TEST BENCH
module CLA_tb();
    reg [3:0]X;
    reg [3:0]Y;
    reg CIN;

    wire COUT;
    wire [3:0]S;

    CLA DUT(.X(X),
            .Y(Y),
            .CIN(CIN),

            .COUT(COUT),
            .S(S)
            );

    initial
        begin
        X = 4'b0000 ; Y = 4'b0000; CIN = 1'b0;
        #10 X = 4'b1110 ; Y = 4'b0011; CIN = 1'b1;
        #10 X = 4'b0110 ; Y = 4'b1100; CIN = 1'b0;
        #10 X = 4'b0011 ; Y = 4'b1010; CIN = 1'b1;
        #10 X = 4'b1000 ; Y = 4'b1111; CIN = 1'b0;
        #10 X = 4'b1111 ; Y = 4'b0101; CIN = 1'b1;
        #10 X = 4'b1001 ; Y = 4'b1100; CIN = 1'b0;
        #10 X = 4'b1101 ; Y = 4'b1111; CIN = 1'b1;
        #10 $finish;
        end
    endmodule
```

- RTL Schematic

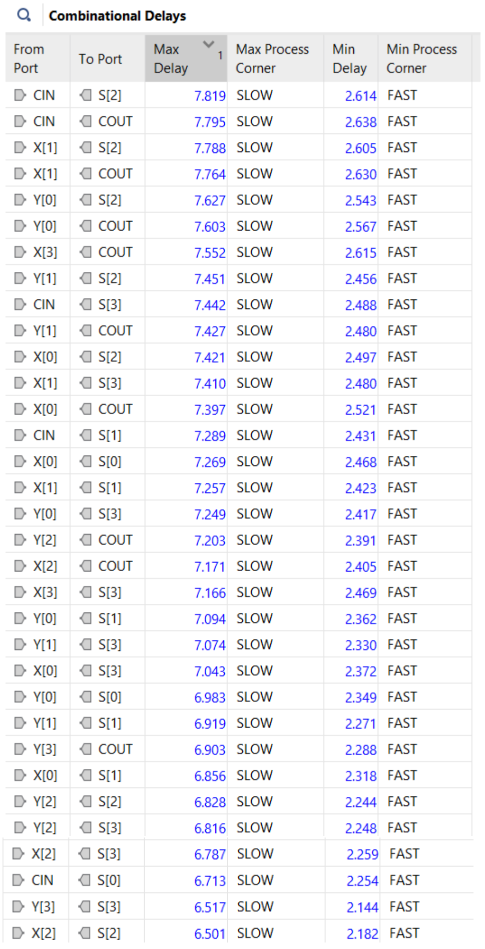


- Technology Schematic



  - There are 2 LUT3 and 4 LUT5 in technology schematic.

- Combinational Delay


Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
CIN	S[2]	7.819	SLOW	2.614	FAST
CIN	COUT	7.795	SLOW	2.638	FAST
X[1]	S[2]	7.788	SLOW	2.605	FAST
X[1]	COUT	7.764	SLOW	2.630	FAST
Y[0]	S[2]	7.627	SLOW	2.543	FAST
Y[0]	COUT	7.603	SLOW	2.567	FAST
X[3]	COUT	7.552	SLOW	2.615	FAST
Y[1]	S[2]	7.451	SLOW	2.456	FAST
CIN	S[3]	7.442	SLOW	2.488	FAST
Y[1]	COUT	7.427	SLOW	2.480	FAST
X[0]	S[2]	7.421	SLOW	2.497	FAST
X[1]	S[3]	7.410	SLOW	2.480	FAST
X[0]	COUT	7.397	SLOW	2.521	FAST
CIN	S[1]	7.289	SLOW	2.431	FAST
X[0]	S[0]	7.269	SLOW	2.468	FAST
X[1]	S[1]	7.257	SLOW	2.423	FAST
Y[0]	S[3]	7.249	SLOW	2.417	FAST
Y[2]	COUT	7.203	SLOW	2.391	FAST
X[2]	COUT	7.171	SLOW	2.405	FAST
X[3]	S[3]	7.166	SLOW	2.469	FAST
Y[0]	S[1]	7.094	SLOW	2.362	FAST
Y[1]	S[3]	7.074	SLOW	2.330	FAST
X[0]	S[3]	7.043	SLOW	2.372	FAST
Y[0]	S[0]	6.983	SLOW	2.349	FAST
Y[1]	S[1]	6.919	SLOW	2.271	FAST
Y[3]	COUT	6.903	SLOW	2.288	FAST
X[0]	S[1]	6.856	SLOW	2.318	FAST
Y[2]	S[2]	6.828	SLOW	2.244	FAST
Y[2]	S[3]	6.816	SLOW	2.248	FAST
X[2]	S[3]	6.787	SLOW	2.259	FAST
CIN	S[0]	6.713	SLOW	2.254	FAST
Y[3]	S[3]	6.517	SLOW	2.144	FAST
X[2]	S[2]	6.501	SLOW	2.182	FAST

  - The maximum delay in my circuit is 7.819ns. We can say that most of the combinational delays are either close to 7ns or higher than 7ns.

6. ADDER-SUBSTRACTOR CIRCUIT WITH OVERFLOW DETECTION

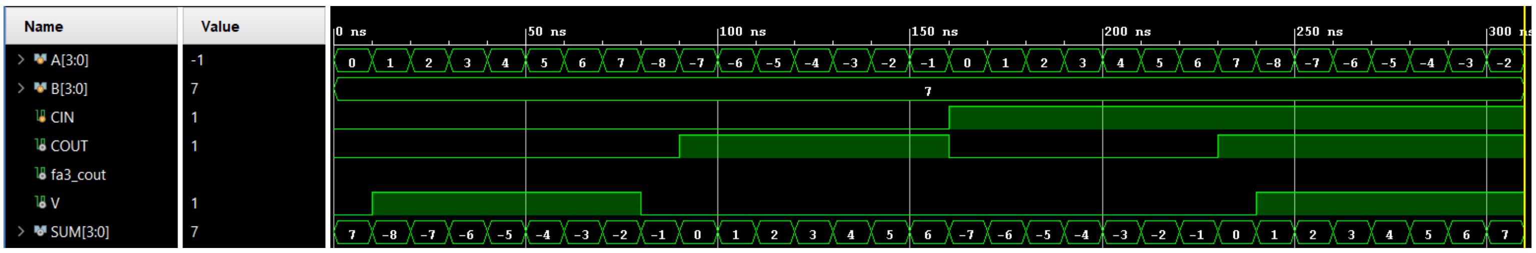
- Verilog Code

```
module ADD_SUB
    (
    input [3:0]A,
    input [3:0]B,
    input CIN,

    output [3:0]SUM,
    output COUT,
    output V
    );
    wire fa1_in, fa2_in, fa3_in, fa4_in;
    wire fa1_cout, fa2_cout, fa3_cout;

    assign fa1_in = CIN ^ B[0];
    assign fa2_in = CIN ^ B[1];
    assign fa3_in = CIN ^ B[2];
    assign fa4_in = CIN ^ B[3];
    FA FA1
    (
    .X(A[0]),
    .Y(fa1_in),
    .CIN(CIN),

    .COUT(fa1_cout),
    .S(SUM[0])
    );

    FA FA2
    (
    .X(A[1]),
    .Y(fa2_in),
    .CIN(fa1_cout),

    .COUT(fa2_cout),
    .S(SUM[1])
    );

    FA FA3
    (
    .X(A[2]),
    .Y(fa3_in),
    .CIN(fa2_cout),

    .COUT(fa3_cout),
    .S(SUM[2])
    );

    FA FA4
    (
    .X(A[3]),
    .Y(fa4_in),
    .CIN(fa3_cout),

    .COUT(COUT),
    .S(SUM[3])
    );

    assign V = fa3_cout ^ COUT;

endmodule
```

- Testbench Code

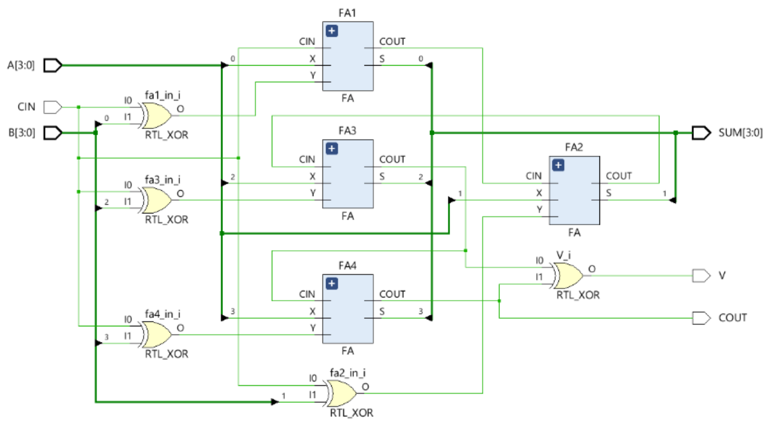
```
module ADD_SUB_tb();

     reg [3:0]A;
     reg [3:0]B;
     reg CIN;

     wire COUT;
     wire V;
     wire [3:0] SUM;
     ADD_SUB AS
     (
     .A(A),
     .B(B),
     .CIN(CIN),

     .COUT(COUT),
     .V(V),
     .SUM(SUM)
     );
     initial
        begin
        A = 4'b0000; B = 4'b0111; CIN = 1'b0;
        #10 A = 4'b0001; B = 4'b0111; CIN = 1'b0;
        #10 A = 4'b0010; B = 4'b0111; CIN = 1'b0;
        #10 A = 4'b0011; B = 4'b0111; CIN = 1'b0;
        #10 A = 4'b0100; B = 4'b0111; CIN = 1'b0;
        #10 A = 4'b0101; B = 4'b0111; CIN = 1'b0;
        #10 A = 4'b0110; B = 4'b0111; CIN = 1'b0;
        #10 A = 4'b0111; B = 4'b0111; CIN = 1'b0;
        #10 A = 4'b1000; B = 4'b0111; CIN = 1'b0;
        #10 A = 4'b1001; B = 4'b0111; CIN = 1'b0;
        #10 A = 4'b1010; B = 4'b0111; CIN = 1'b0;
        #10 A = 4'b1011; B = 4'b0111; CIN = 1'b0;
        #10 A = 4'b1100; B = 4'b0111; CIN = 1'b0;
        #10 A = 4'b1101; B = 4'b0111; CIN = 1'b0;
        #10 A = 4'b1110; B = 4'b0111; CIN = 1'b0;
        #10 A = 4'b1111; B = 4'b0111; CIN = 1'b0;
        #10 A = 4'b0000; B = 4'b0111; CIN = 1'b1;
        #10 A = 4'b0001; B = 4'b0111; CIN = 1'b1;
        #10 A = 4'b0010; B = 4'b0111; CIN = 1'b1;
        #10 A = 4'b0011; B = 4'b0111; CIN = 1'b1;
        #10 A = 4'b0100; B = 4'b0111; CIN = 1'b1;
        #10 A = 4'b0101; B = 4'b0111; CIN = 1'b1;
        #10 A = 4'b0110; B = 4'b0111; CIN = 1'b1;
        #10 A = 4'b0111; B = 4'b0111; CIN = 1'b1;
        #10 A = 4'b1000; B = 4'b0111; CIN = 1'b1;
        #10 A = 4'b1001; B = 4'b0111; CIN = 1'b1;
        #10 A = 4'b1010; B = 4'b0111; CIN = 1'b1;
        #10 A = 4'b1011; B = 4'b0111; CIN = 1'b1;
        #10 A = 4'b1100; B = 4'b0111; CIN = 1'b1;
        #10 A = 4'b1101; B = 4'b0111; CIN = 1'b1;
        #10 A = 4'b1110; B = 4'b0111; CIN = 1'b1;
        #10 A = 4'b1111; B = 4'b0111; CIN = 1'b1;
        $finish;
        end
endmodule
```

- Behavioral Simulation

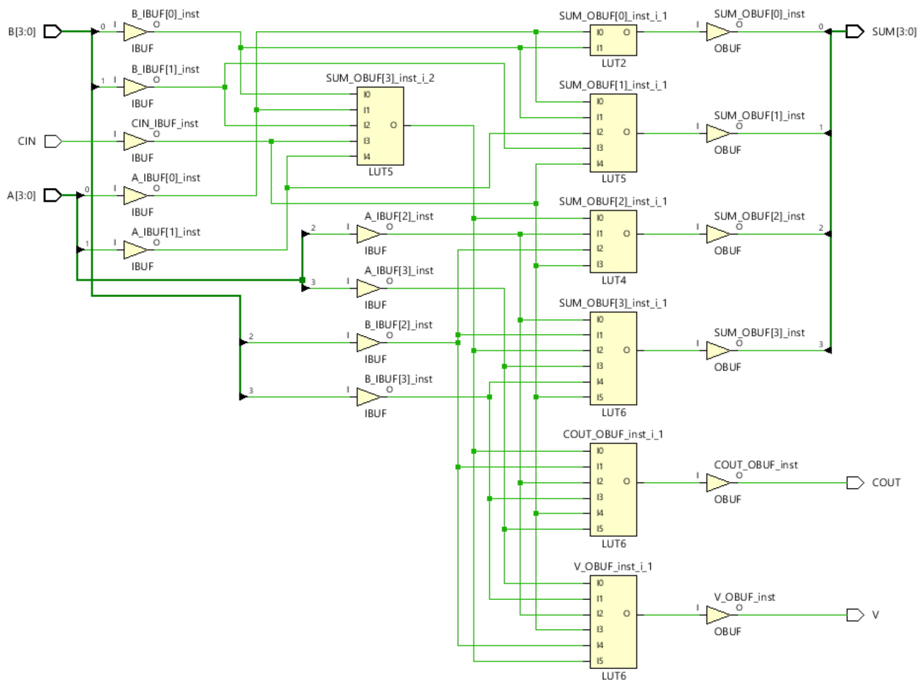


- The values representing the yields at the outputs of the 3rd and 4th Full adders are fa3_cout and COUT, respectively.

- RTL Schematic

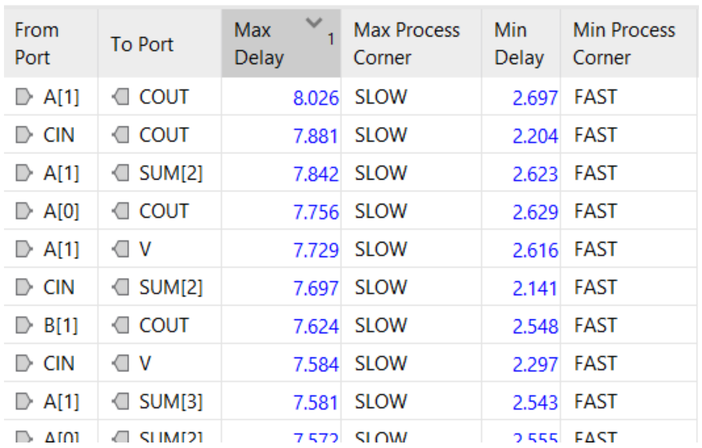


- Technology Schematic



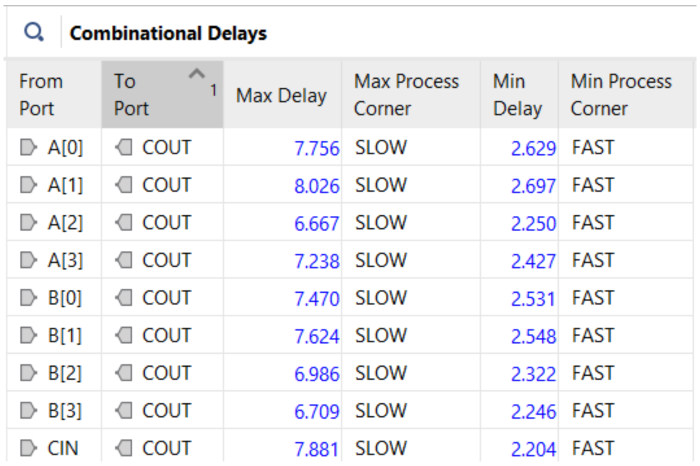
➢ There are 1 LUT2, 1 LUT4, 2 LUT5 and 3 LUT6 in technology schematic.

- Combinational Delay



From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
A[1]	COUT	8.026	SLOW	2.697	FAST
CIN	COUT	7.881	SLOW	2.204	FAST
A[1]	SUM[2]	7.842	SLOW	2.623	FAST
A[0]	COUT	7.756	SLOW	2.629	FAST
A[1]	V	7.729	SLOW	2.616	FAST
CIN	SUM[2]	7.697	SLOW	2.141	FAST
B[1]	COUT	7.624	SLOW	2.548	FAST
CIN	V	7.584	SLOW	2.297	FAST
A[1]	SUM[3]	7.581	SLOW	2.543	FAST
A[0]	SUM[2]	7.572	SLOW	2.555	FAST

➢ The maximum delay in my circuit is 8.026ns



Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
A[0]	COUT	7.756	SLOW	2.629	FAST
A[1]	COUT	8.026	SLOW	2.697	FAST
A[2]	COUT	6.667	SLOW	2.250	FAST
A[3]	COUT	7.238	SLOW	2.427	FAST
B[0]	COUT	7.470	SLOW	2.531	FAST
B[1]	COUT	7.624	SLOW	2.548	FAST
B[2]	COUT	6.986	SLOW	2.322	FAST
B[3]	COUT	6.709	SLOW	2.246	FAST
CIN	COUT	7.881	SLOW	2.204	FAST

➢ In the photo above, the delays are arranged according to the output COUT. Delays are usually between 7ns and 8ns.

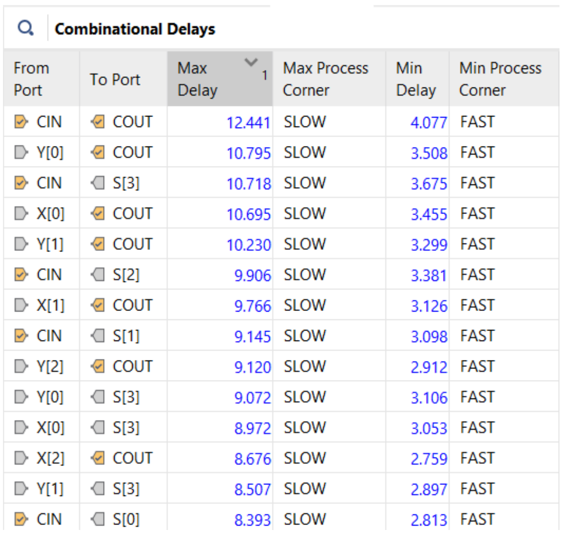
7. RCA AND CLA COMPARISON WITH DONT TOUCH STRUCTURE (OPTIONAL)

- Our results with our max delays were supposed to have less delay in terms of coding and structure in CLA, but since our program did its own optimization, the delays were equal in both cases. If we give the structure (*dont _touch = "true"*) to the wire parts, we can find the differences between them.
- Verilog Code (RCA and CLA)

```
module RCA(
    input [3:0]X,
    input [3:0]Y,
    input CIN,

    output COUT,
    output [3:0]S
    );
    (* dont_touch = "true" *)wire c1,c2,c3,c4;

    FA fulladder1(
        .X(X[0]),
        .Y(Y[0]),
        .CIN(CIN),

        .COUT(c1),
        .S(S[0])
        );

    FA fulladder2(
        .X(X[1]),
        .Y(Y[1]),
        .CIN(c1),

        .COUT(c2),
        .S(S[1])
        );

    FA fulladder3(
        .X(X[2]),
        .Y(Y[2]),
        .CIN(c2),

        .COUT(c3),
        .S(S[2])
        );

    FA fulladder4(
        .X(X[3]),
        .Y(Y[3]),
        .CIN(c3),

        .COUT(COUT),
        .S(S[3])
        );

endmodule
```

```
module CGA ( //carry , generate , propagate
            input [3:0]P,
            input [3:0]G,
            input CIN,

            output [3:0]COUT
            );

    assign COUT[0] = G[0] | (P[0] & CIN);
    assign COUT[1] = G[1] | (G[0] & P[1]) | (P[1] &
P[0] & CIN);
    assign COUT[2] = G[2] | (G[1] & P[2]) | (G[0] &
P[2] & P[1]) | ( P[2] & P[1] & P[0] & CIN);
    assign COUT[3] = G[3] | (G[2] & P[3]) | (G[1] &
P[3] & P[2]) | (G[0] & P[3] & P[2] & P[1]) | (P[3] &
P[2] & P[1] & P[0] & CIN);
endmodule

module CLA (
            input [3:0]X,
            input [3:0]Y,
            input CIN,

            output COUT,
            output [3:0]S
            );

(* dont_touch = "true" *) wire [3:0]P;
(* dont_touch = "true" *) wire [3:0]G;
(* dont_touch = "true" *) wire [3:0]C;

    assign P = X ^ Y;
    assign G = X & Y;

    CGA CGA(
            .P(P),
            .G(G),

            .CIN(CIN),
            .COUT(C)
            );

    assign COUT = C[3];

    assign S[0] = P[0] ^ CIN;
    assign S[1] = P[1] ^ C[0];
    assign S[2] = P[2] ^ C[1];
    assign S[3] = P[3] ^ C[2];

endmodule
```

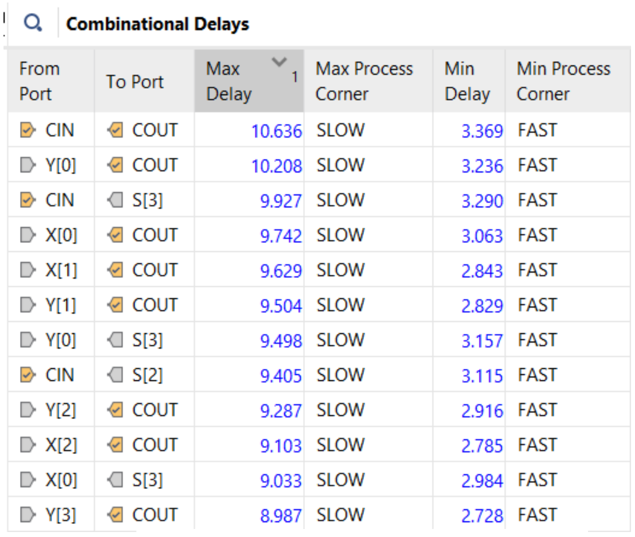
- Combinational Delay



Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
CIN	COUT	12.441	SLOW	4.077	FAST
Y[0]	COUT	10.795	SLOW	3.508	FAST
CIN	S[3]	10.718	SLOW	3.675	FAST
X[0]	COUT	10.695	SLOW	3.455	FAST
Y[1]	COUT	10.230	SLOW	3.299	FAST
CIN	S[2]	9.906	SLOW	3.381	FAST
X[1]	COUT	9.766	SLOW	3.126	FAST
CIN	S[1]	9.145	SLOW	3.098	FAST
Y[2]	COUT	9.120	SLOW	2.912	FAST
Y[0]	S[3]	9.072	SLOW	3.106	FAST
X[0]	S[3]	8.972	SLOW	3.053	FAST
X[2]	COUT	8.676	SLOW	2.759	FAST
Y[1]	S[3]	8.507	SLOW	2.897	FAST
CIN	S[0]	8.393	SLOW	2.813	FAST

RCA DELAY WITH DONT_TOUCH



Combinational Delays

From Port	To Port	Max Delay	Max Process Corner	Min Delay	Min Process Corner
CIN	COUT	10.636	SLOW	3.369	FAST
Y[0]	COUT	10.208	SLOW	3.236	FAST
CIN	S[3]	9.927	SLOW	3.290	FAST
X[0]	COUT	9.742	SLOW	3.063	FAST
X[1]	COUT	9.629	SLOW	2.843	FAST
Y[1]	COUT	9.504	SLOW	2.829	FAST
Y[0]	S[3]	9.498	SLOW	3.157	FAST
CIN	S[2]	9.405	SLOW	3.115	FAST
Y[2]	COUT	9.287	SLOW	2.916	FAST
X[2]	COUT	9.103	SLOW	2.785	FAST
X[0]	S[3]	9.033	SLOW	2.984	FAST
Y[3]	COUT	8.987	SLOW	2.728	FAST

CLA DELAY WITH DONT_TOUCH

- As you can see, the max delay in the RCA module that we made with the dont_touch structure was 12,441 ns, and the max delay in the CLA module was 10.636ns. There is a difference of about 2ns between them. Although both are summation circuits, the CLA module works faster than the RCA module. This is due to its design.