DIGITAL SYSTEM DESIGN APPLICATIONS

(CRN: 11275)

THE REPORT OF EXPERIMENT – 4

Faculty of Electrical and Electronics Engineering

Electronics and Communication Engineering

Yusuf Tekin – 040200043

Yusuf Tekin
040200043

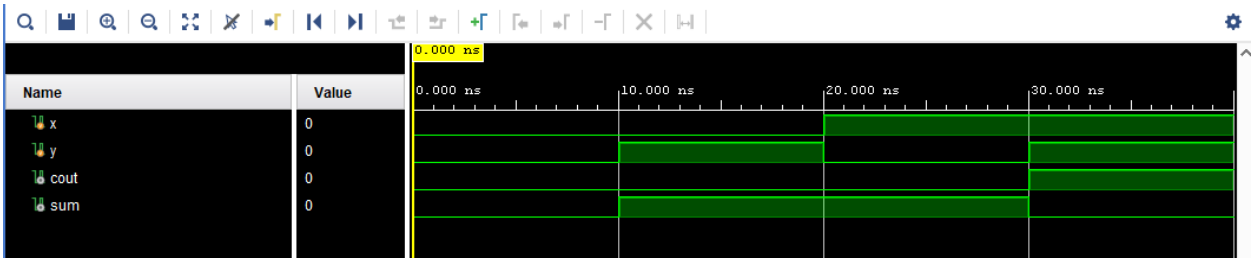
1. Half Adder

```
module HA(

    input x,
    input y,
    output cout,
    output sum
    );

assign cout = x & y;
assign sum = x ^ y;

endmodule
```

Half Adder Verilog Code

```
`timescale 1ns / 1ps
module HA_tb();

reg x, y;
wire cout, sum;

HA ha1(
    .x(x),
    .y(y),
    .cout(cout),
    .sum(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
```

Half Adder Testbench Code



Half Adder Behavioral Simulation

Yusuf Tekin
040200043

2. Full Adder

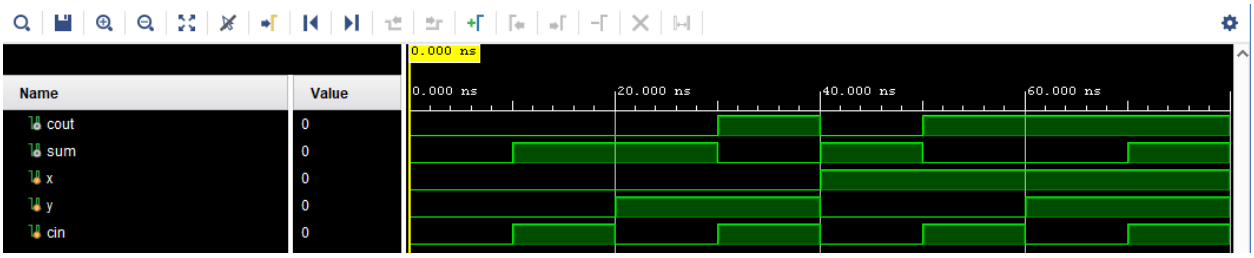
```
module FA(

    input x,
    input y,
    input cin,
    output cout,
    output sum
    );
wire [2:0] fa_temp_wire;

HA ha1(.x(x), .y(y), .sum(fa_temp_wire[0]), .cout(fa_temp_wire[1]));
HA ha2(.x(fa_temp_wire[0]), .y(cin), .sum(sum), .cout(fa_temp_wire[2]));

assign cout = fa_temp_wire[1] | fa_temp_wire[2];

endmodule
```

Full Adder Verilog Code

```
`timescale 1ns / 1ps

module FA_tb();

wire cout, sum;
reg x, y, cin;
FA uut(
    .x(x),
    .y(y),
    .cin(cin),
    .cout(cout),
    .sum(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 Testbench Code

Yusuf Tekin
040200043



Full Adder Behavioral Simulation

3. Ripple Carry Adder

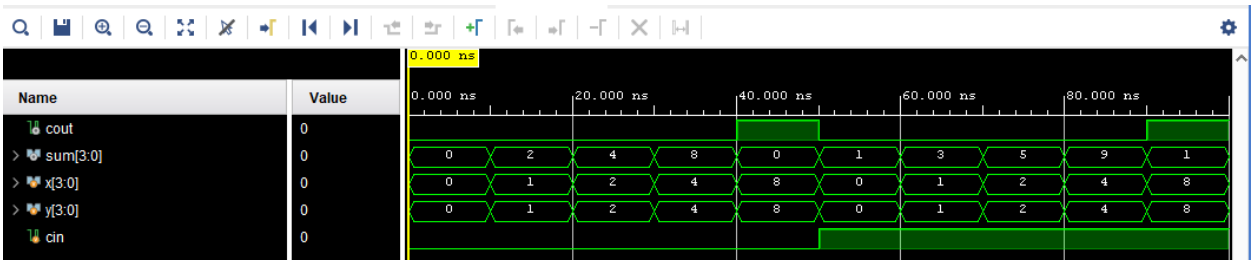
```
module RCA(

    input [3:0] x,
    input [3:0] y,
    input cin,
    output cout,
    output [3:0] sum
    );

wire [2:0] rca_wire;

FA fa0(.x(x[0]), .y(y[0]), .cin(cin), .cout(rca_wire[0]), .sum(sum[0]));
FA fa1(.x(x[1]), .y(y[1]), .cin(rca_wire[0]), .cout(rca_wire[1]), .sum(sum[1]));
FA fa2(.x(x[2]), .y(y[2]), .cin(rca_wire[1]), .cout(rca_wire[2]), .sum(sum[2]));
FA fa3(.x(x[3]), .y(y[3]), .cin(rca_wire[2]), .cout(cout), .sum(sum[3]));

endmodule
```

RCA Verilog Code

Yusuf Tekin
040200043

```
`timescale 1ns / 1ps


module RCA_tb();

wire cout;
wire [3:0] sum;
reg [3:0] x, y;
reg cin;

RCA uut(
    .x(x),
    .y(y),
    .cin(cin),
    .cout(cout),
    .sum(sum));

initial begin
    x = 4'b0000; y = 4'b0000; cin = 1'b0; #10;
    x = 4'b0001; y = 4'b0001; cin = 1'b0; #10;
    x = 4'b0010; y = 4'b0010; cin = 1'b0; #10;
    x = 4'b0100; y = 4'b0100; cin = 1'b0; #10;
    x = 4'b1000; y = 4'b1000; cin = 1'b0; #10;
    x = 4'b0000; y = 4'b0000; cin = 1'b1; #10;
    x = 4'b0001; y = 4'b0001; cin = 1'b1; #10;
    x = 4'b0010; y = 4'b0010; cin = 1'b1; #10;
    x = 4'b0100; y = 4'b0100; cin = 1'b1; #10;
    x = 4'b1000; y = 4'b1000; cin = 1'b1; #10;
    $finish();
end
endmodule
```

RCA Testbench Code



RCA Behavioral Simulation

Yusuf Tekin
040200043

4. Ripple Carry Adder “parametric”

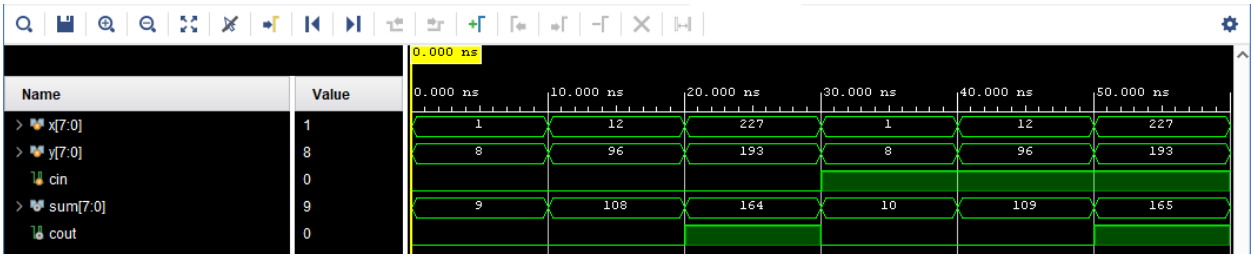
```
module parametric_RCA #(parameter SIZE = 8)(

    input [SIZE - 1:0] x,
    input [SIZE - 1:0] y,
    input cin,
    output cout,
    output [SIZE - 1:0] sum
    );

wire [SIZE - 2:0] carry;
genvar i;

generate
    for(i = 0; i < SIZE; i = i + 1) begin  : RCA_generate
        if (i == 0) begin
            FA fa_first(.x(x[i]),
                        .y(y[i]),
                        .cin(cin),
                        .cout(carry[i]),
                        .sum(sum[i]));
        end
        else if (i == SIZE - 1) begin
            FA fa_last (.x(x[i]),
                        .y(y[i]),
                        .cin(carry[i-1]),
                        .cout(cout),
                        .sum(sum[i]));
        end
        else begin
            FA fa_middle(.x(x[i]),
                         .y(y[i]),
                         .cin(carry[i-1]),
                         .cout(carry[i]),
                         .sum(sum[i]));
        end
    end
endgenerate
endmodule
```

Parametric RCA Verilog Code

Yusuf Tekin
040200043

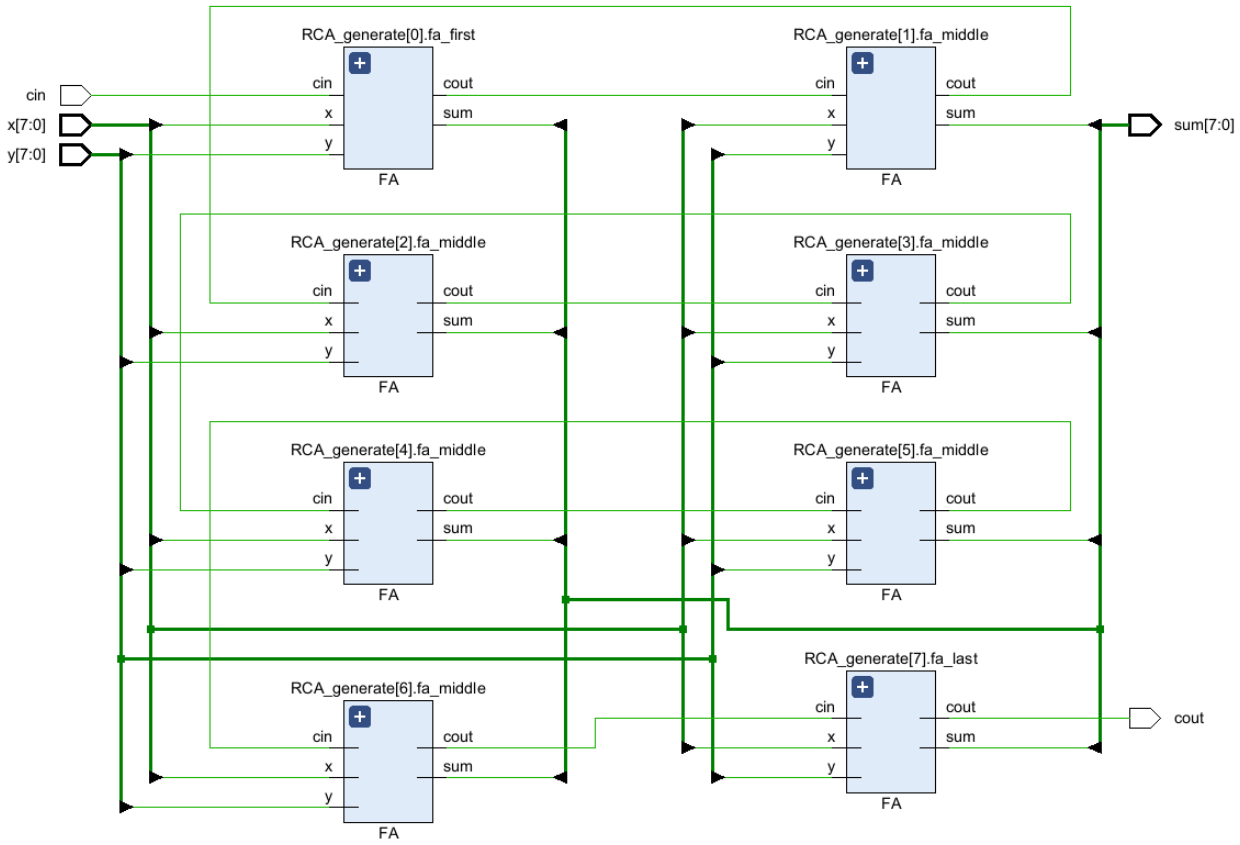
```
`timescale 1ns / 1ps

module parametric_RCA_tb();

reg [7:0] x, y;
reg cin;
wire [7:0] sum;
wire cout;



parametric_RCA #(.SIZE(8)) uut(
    .x(x),
    .y(y),
    .cin(cin),
    .cout(cout),
    .sum(sum)
    );

initial begin
    x = 8'b0000_0001; y = 8'b0000_1000; cin = 1'b0; #10;
    x = 8'b0000_1100; y = 8'b0110_0000; cin = 1'b0; #10;
    x = 8'b1110_0011; y = 8'b1100_0001; cin = 1'b0; #10;
    x = 8'b0000_0001; y = 8'b0000_1000; cin = 1'b1; #10;
    x = 8'b0000_1100; y = 8'b0110_0000; cin = 1'b1; #10;
    x = 8'b1110_0011; y = 8'b1100_0001; cin = 1'b1; #10;
    $finish();
end
endmodule
```

Parametric RCA Testbench Code



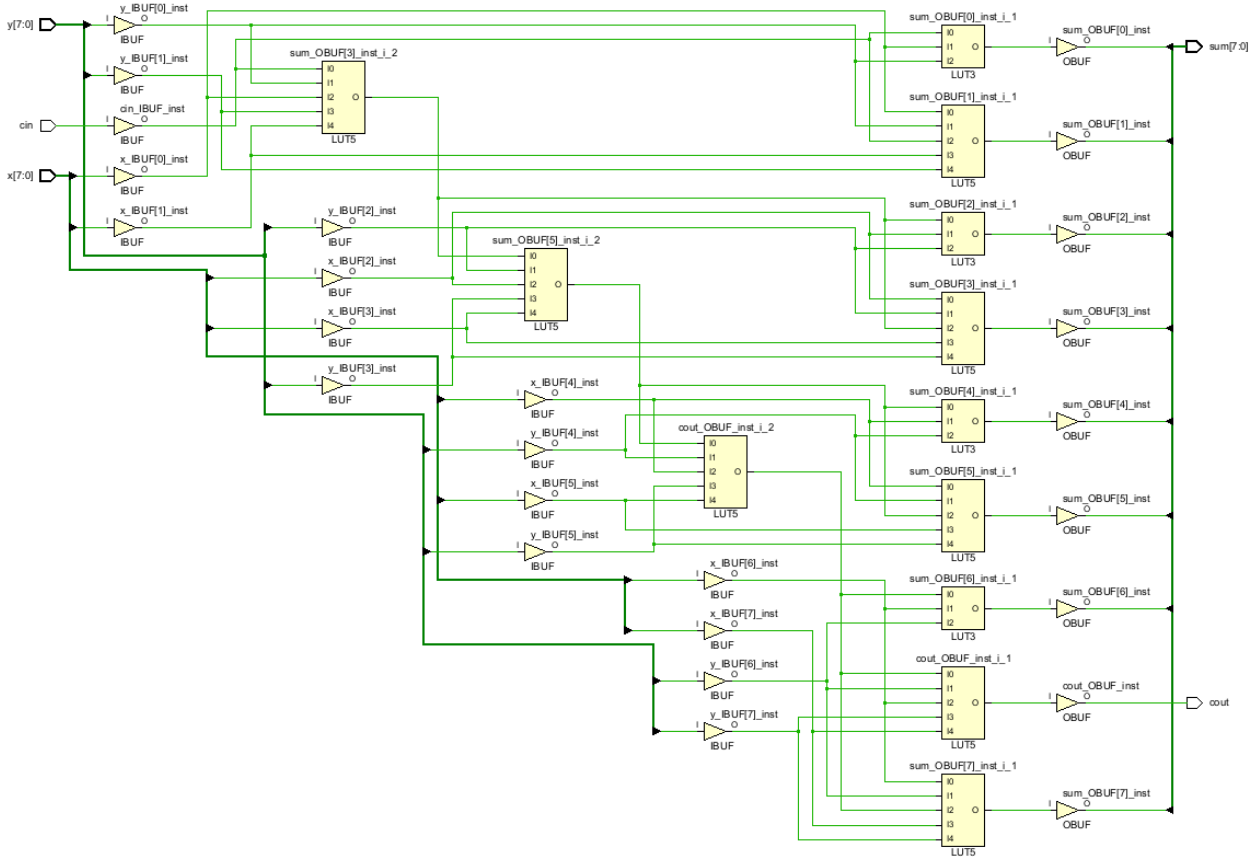
Parametric RCA Behavioral Simulation

Yusuf Tekin
040200043



RTL Schematic - Parametric RCA

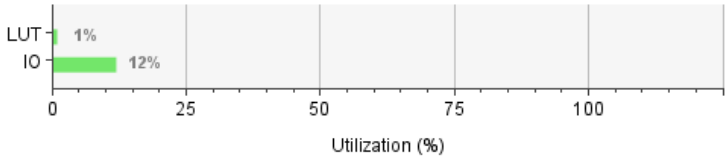
Yusuf Tekin
040200043



Technology Schematic - Parametric RCA

Summary

Resource	Utilization	Available	Utilization %
LUT	8	32600	0.02
IO	26	210	12.38



Utilization Summary - Parametric RCA

Yusuf Tekin
040200043

Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total ...	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	∞	6	5	3	y[0]	cout	14.793	5.688	9.104	∞	input port clock			0.000
Path 2	∞	6	5	3	y[0]	sum[7]	14.410	5.449	8.961	∞	input port clock			0.000
Path 3	∞	6	5	3	y[0]	sum[6]	14.259	5.452	8.807	∞	input port clock			0.000
Path 4	∞	5	4	3	y[0]	sum[5]	13.665	5.553	8.112	∞	input port clock			0.000
Path 5	∞	5	4	3	y[0]	sum[4]	12.821	5.332	7.489	∞	input port clock			0.000
Path 6	∞	4	3	3	y[0]	sum[2]	12.320	4.961	7.359	∞	input port clock			0.000
Path 7	∞	4	3	3	y[0]	sum[3]	12.097	4.954	7.142	∞	input port clock			0.000
Path 8	∞	3	2	3	y[0]	sum[1]	11.090	4.619	6.471	∞	input port clock			0.000
Path 9	∞	3	2	3	y[0]	sum[0]	10.859	4.600	6.259	∞	input port clock			0.000

Setup Timing - Parametric RCA

Unconstrained Paths - NONE - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total ...	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 18	∞	3	2	3	x[2]	sum[2]	2.839	1.517	1.322	-∞	input port clock			0.000
Path 17	∞	3	2	3	x[4]	sum[5]	2.616	1.582	1.034	-∞	input port clock			0.000
Path 16	∞	3	2	3	cin	sum[0]	2.592	1.508	1.083	-∞	input port clock			0.000
Path 15	∞	3	2	3	cin	sum[1]	2.565	1.527	1.038	-∞	input port clock			0.000
Path 14	∞	3	2	3	x[4]	sum[4]	2.503	1.532	0.971	-∞	input port clock			0.000
Path 13	∞	3	2	3	y[2]	sum[3]	2.450	1.516	0.933	-∞	input port clock			0.000
Path 12	∞	3	2	3	x[6]	cout	2.348	1.606	0.742	-∞	input port clock			0.000
Path 11	∞	3	2	3	x[6]	sum[7]	2.216	1.533	0.683	-∞	input port clock			0.000
Path 10	∞	3	2	3	x[6]	sum[6]	2.208	1.535	0.673	-∞	input port clock			0.000

Hold Timing - Parametric RCA

Yusuf Tekin
040200043

5. Carry Lookahead Adder “parametric”

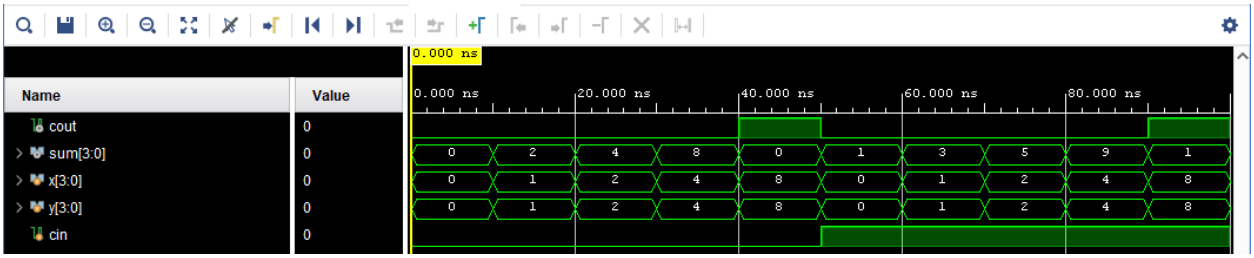
```
module CLA #(parameter SIZE = 8)(

    input [SIZE-1:0] x,
    input [SIZE-1:0] y,
    input cin,
    output cout,
    output [SIZE-1:0] s
    );

wire [SIZE-1:0] wire_g, wire_p, wire_c;
genvar i;

    generate
        for (i = 0; i<SIZE; i = i + 1) begin
            assign wire_g[i] = x[i] & y[i];
            assign wire_p[i] = x[i] ^ y[i];
        end
    endgenerate

    assign wire_c[0] = cin;
    generate
    for (i = 1; i<SIZE; i = i + 1) begin
        assign wire_c[i] = wire_g[i-1] | (wire_p[i-1] & wire_c[i-1]);
    end
    endgenerate

    generate
    for (i = 0; i<SIZE; i = i + 1) begin
        assign s[i] = wire_p[i] ^ wire_c[i];
    end
    endgenerate

    assign cout = wire_g[SIZE-1] | (wire_p[SIZE-1] & wire_c[SIZE-1]);
endmodule
```

CLA Verilog Code

Yusuf Tekin
040200043

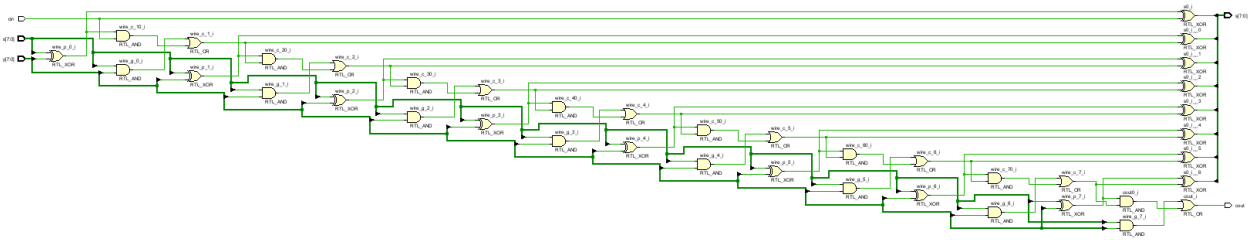
```
`timescale 1ns / 1ps

module CLA_tb();

wire [7:0] s;
wire cout;
reg cin;
reg [7:0] x, y;

CLA #(.SIZE(8)) uut(
    .x(x),
    .y(y),
    .cin(cin),
    .cout(cout),
    .s(s));

initial begin
    x = 4'b0000; y = 4'b0000; cin = 1'b0; #10;
    x = 4'b0001; y = 4'b0001; cin = 1'b0; #10;
    x = 4'b0010; y = 4'b0010; cin = 1'b0; #10;
    x = 4'b0100; y = 4'b0100; cin = 1'b0; #10;
    x = 4'b1000; y = 4'b1000; cin = 1'b0; #10;
    x = 4'b0000; y = 4'b0000; cin = 1'b1; #10;
    x = 4'b0001; y = 4'b0001; cin = 1'b1; #10;
    x = 4'b0010; y = 4'b0010; cin = 1'b1; #10;
    x = 4'b0100; y = 4'b0100; cin = 1'b1; #10;
    x = 4'b1000; y = 4'b1000; cin = 1'b1; #10;
    $finish();
end
endmodule
```

CLA Testbench Code

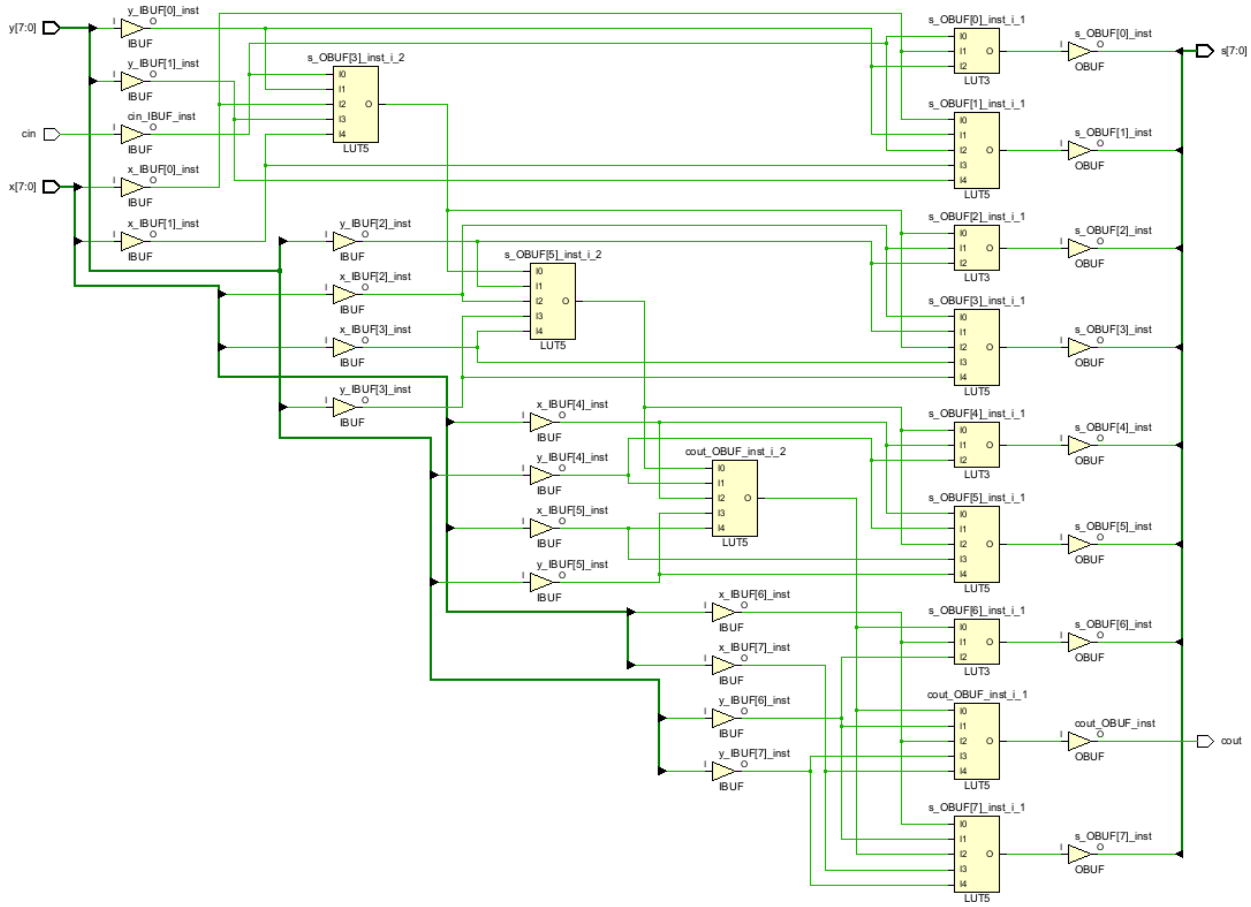


CLA Behavioral Simulation

Yusuf Tekin
040200043



CLA RTL Schematic



CLA Technology Schematic

Yusuf Tekin
040200043

Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total ...	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	∞	6	5	3	y[0]	cout	14.793	5.688	9.104	∞	input port clock			0.000
Path 2	∞	6	5	3	y[0]	s[7]	14.410	5.449	8.961	∞	input port clock			0.000
Path 3	∞	6	5	3	y[0]	s[6]	14.259	5.452	8.807	∞	input port clock			0.000
Path 4	∞	5	4	3	y[0]	s[5]	13.665	5.553	8.112	∞	input port clock			0.000
Path 5	∞	5	4	3	y[0]	s[4]	12.821	5.332	7.489	∞	input port clock			0.000
Path 6	∞	4	3	3	y[0]	s[2]	12.320	4.961	7.359	∞	input port clock			0.000
Path 7	∞	4	3	3	y[0]	s[3]	12.097	4.954	7.142	∞	input port clock			0.000
Path 8	∞	3	2	3	y[0]	s[1]	11.090	4.619	6.471	∞	input port clock			0.000
Path 9	∞	3	2	3	y[0]	s[0]	10.859	4.600	6.259	∞	input port clock			0.000

Setup Delays - CLA

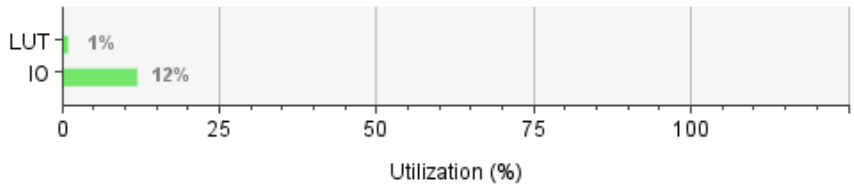
Unconstrained Paths - NONE - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total ...	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 18	∞	3	2	3	x[2]	s[2]	2.839	1.517	1.322	-∞	input port clock			0.000
Path 17	∞	3	2	3	x[4]	s[5]	2.616	1.582	1.034	-∞	input port clock			0.000
Path 16	∞	3	2	3	cin	s[0]	2.592	1.508	1.083	-∞	input port clock			0.000
Path 15	∞	3	2	3	cin	s[1]	2.565	1.527	1.038	-∞	input port clock			0.000
Path 14	∞	3	2	3	x[4]	s[4]	2.503	1.532	0.971	-∞	input port clock			0.000
Path 13	∞	3	2	3	y[2]	s[3]	2.450	1.516	0.933	-∞	input port clock			0.000
Path 12	∞	3	2	3	x[6]	cout	2.348	1.606	0.742	-∞	input port clock			0.000
Path 11	∞	3	2	3	x[6]	s[7]	2.216	1.533	0.683	-∞	input port clock			0.000
Path 10	∞	3	2	3	x[6]	s[6]	2.208	1.535	0.673	-∞	input port clock			0.000

Hold Delays - CLA

Summary

Resource	Utilization	Available	Utilization %
LUT	8	32600	0.02
IO	26	210	12.38



Utilization Summary - CLA

6. Behavioral Adder

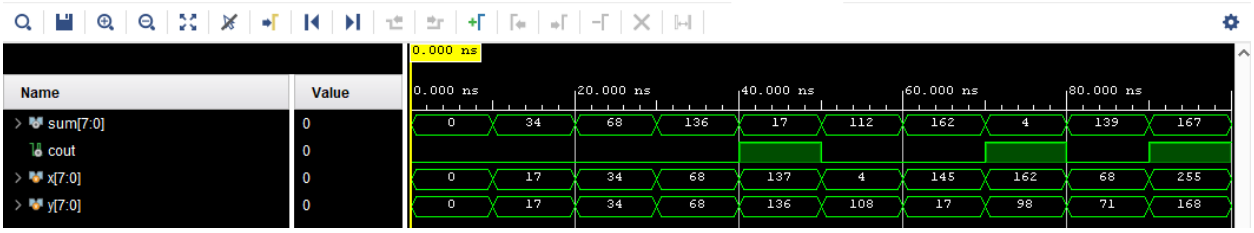
```
module BA #(parameter SIZE = 8)(

    input [SIZE-1:0] x,
    input [SIZE-1:0] y,
    output cout,
    output [SIZE-1:0] sum
    );

wire [SIZE:0] temp;

assign temp = x + y;
assign sum = temp[SIZE-1:0];
assign cout = temp[SIZE];

endmodule
```

Behavioral Adder Verilog Code

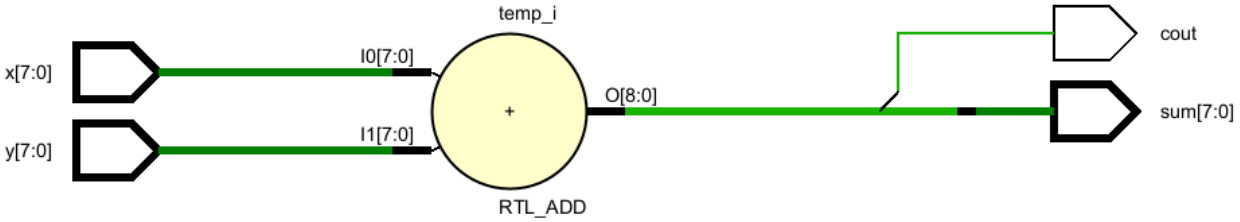
```
`timescale 1ns / 1ps

module BA_tb();

wire [7:0] sum;
wire cout;
reg [7:0] x, y;

BA #(.SIZE(8)) uut(
    .x(x),
    .y(y),
    .cout(cout),
    .sum(sum));

initial begin
    x = 8'b0000_0000; y = 8'b0000_0000; #10;
    x = 8'b0001_0001; y = 8'b0001_0001; #10;
    x = 8'b0010_0010; y = 8'b0010_0010; #10;
    x = 8'b0100_0100; y = 8'b0100_0100; #10;
    x = 8'b1000_1001; y = 8'b1000_1000; #10;
    x = 8'b0000_0100; y = 8'b0110_1100; #10;
    x = 8'b1001_0001; y = 8'b0001_0001; #10;
    x = 8'b1010_0010; y = 8'b0110_0010; #10;
    x = 8'b0100_0100; y = 8'b0100_0111; #10;
    x = 8'b1111_1111; y = 8'b1010_1000; #10;
    $finish();
end
endmodule
```

Behavioral Adder Testbench Code

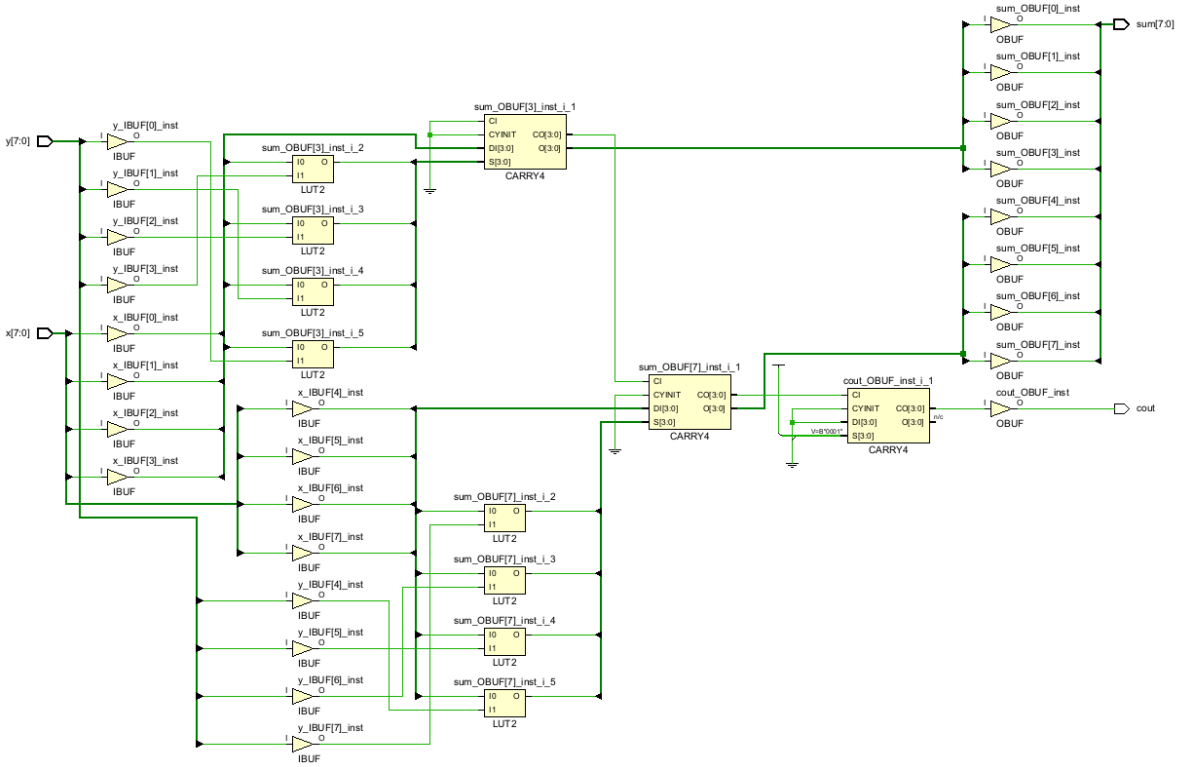
Yusuf Tekin
040200043



Behavioral Adder Behavioral Simulation



Behavioral Adder RTL Schematic

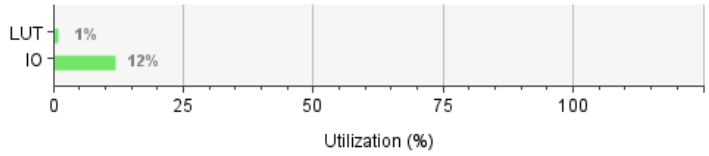


Behavioral Adder Technology Schematic

Yusuf Tekin
040200043

Summary

Resource	Utilization	Available	Utilization %
LUT	8	32600	0.02
IO	25	210	11.90



Utilization Summary

Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total ...	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	∞	4	2	1	y[0]	sum[2]	11.993	5.334	6.658	∞	input port clock			0.000
Path 2	∞	4	2	1	y[0]	sum[0]	11.948	5.022	6.926	∞	input port clock			0.000
Path 3	∞	5	2	1	y[0]	sum[5]	11.908	5.656	6.252	∞	input port clock			0.000
Path 4	∞	4	2	1	y[0]	sum[1]	11.896	5.222	6.674	∞	input port clock			0.000
Path 5	∞	6	2	1	y[0]	cout	11.734	5.788	5.945	∞	input port clock			0.000
Path 6	∞	5	2	1	y[0]	sum[6]	11.662	5.565	6.097	∞	input port clock			0.000
Path 7	∞	5	2	1	y[0]	sum[4]	11.611	5.549	6.062	∞	input port clock			0.000
Path 8	∞	5	2	1	y[0]	sum[7]	11.591	5.640	5.950	∞	input port clock			0.000
Path 9	∞	4	2	1	y[0]	sum[3]	11.307	5.390	5.917	∞	input port clock			0.000

Setup Delays - Behavioral Adder

Unconstrained Paths - NONE - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total ...	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 18	∞	4	2	2	x[0]	sum[0]	3.351	1.661	1.690	-∞	input port clock			0.000
Path 17	∞	4	2	2	x[1]	sum[1]	3.102	1.668	1.434	-∞	input port clock			0.000
Path 16	∞	4	2	2	x[2]	sum[2]	2.928	1.646	1.282	-∞	input port clock			0.000
Path 15	∞	4	2	2	x[2]	sum[3]	2.652	1.674	0.977	-∞	input port clock			0.000
Path 14	∞	4	2	2	x[4]	cout	2.649	1.807	0.842	-∞	input port clock			0.000
Path 13	∞	4	2	2	x[4]	sum[4]	2.637	1.664	0.973	-∞	input port clock			0.000
Path 12	∞	3	2	2	x[4]	sum[5]	2.636	1.664	0.972	-∞	input port clock			0.000
Path 11	∞	3	2	2	x[4]	sum[6]	2.596	1.702	0.893	-∞	input port clock			0.000
Path 10	∞	3	2	2	x[4]	sum[7]	2.577	1.723	0.853	-∞	input port clock			0.000

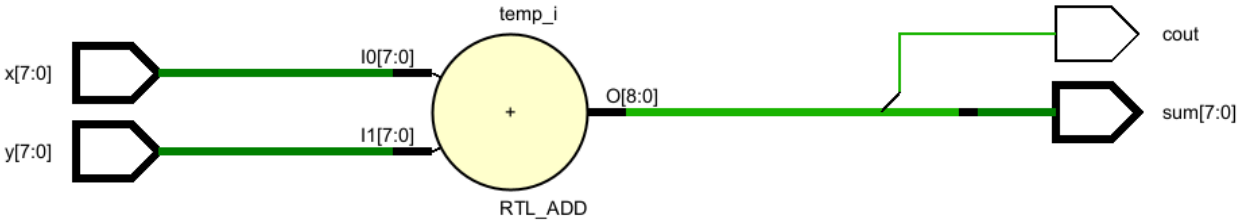
Hold Delays - Behavioral Adder

To compare the behavioral adder with previous adders, it can be seen that: I/O ports are dropped by one because the Cin did not implement to the behavioral adder. Even though all the designs use 8 LUTs, the delays are different. For the RCA and CLA designs, Cout has the longest delay which is around 14ns for both designs. However, that's not the case for the behavioral design. The values of the delays of the behavioral design are close numbers around 11 ns and all of them are nearly the same.

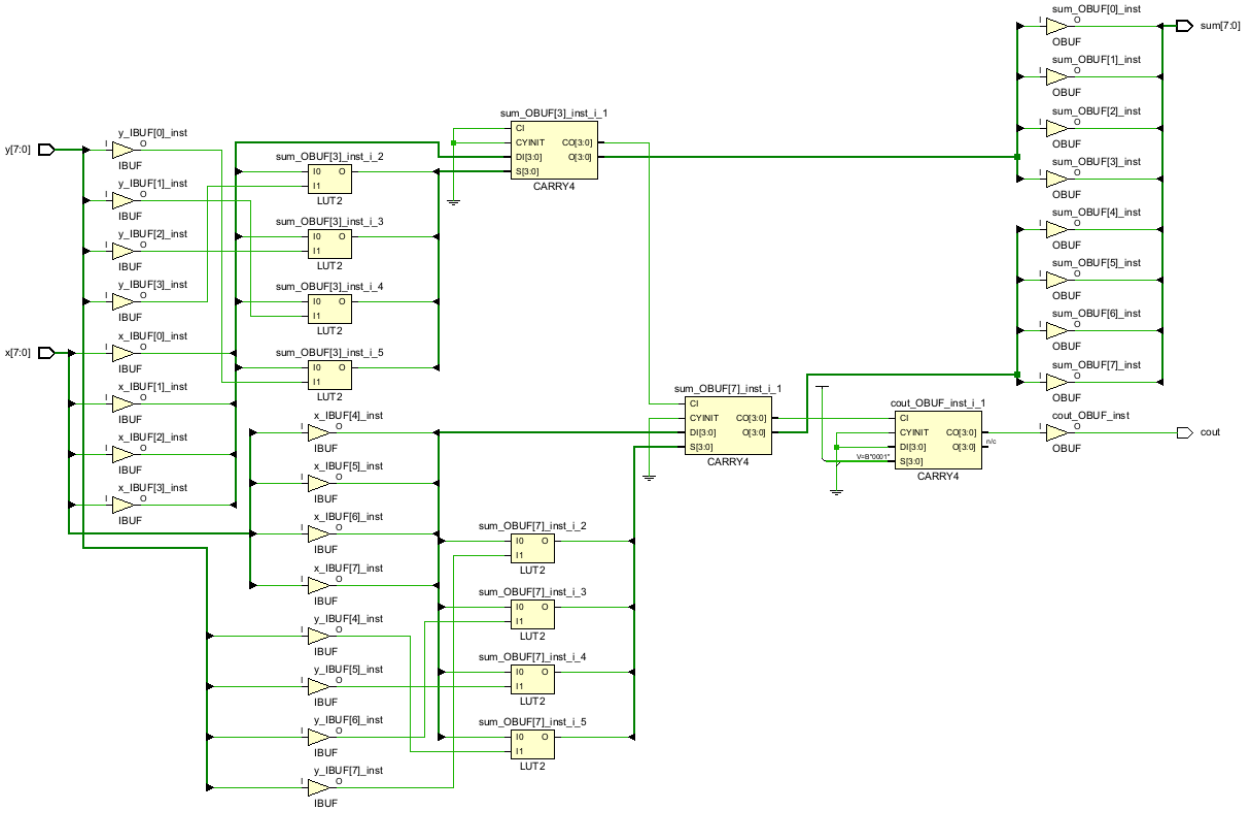
Both RCA and CLA design uses the LUT5s and LUT3s in the technology implementation however, in the behavioral design LUT2s and CARRY4s are observed. Also, the behavioral design has the simplest RTL schematic of all. RCA has the full-adders and CLA has the basic logic gates in the RTL schematic.

Yusuf Tekin
040200043

After “DON’T_TOUCH” Attribute:



RTL Schematic of BA (Behavioral Adder)



Technology Schematic of BA

Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total ...	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	∞	4	2	1	y[0]	sum[2]	11.993	5.334	6.658	∞	input port clock			0.000
Path 2	∞	4	2	1	y[0]	sum[0]	11.948	5.022	6.926	∞	input port clock			0.000
Path 3	∞	5	2	1	y[0]	sum[5]	11.908	5.656	6.252	∞	input port clock			0.000
Path 4	∞	4	2	1	y[0]	sum[1]	11.896	5.222	6.674	∞	input port clock			0.000
Path 5	∞	6	2	1	y[0]	cout	11.734	5.788	5.945	∞	input port clock			0.000
Path 6	∞	5	2	1	y[0]	sum[6]	11.662	5.565	6.097	∞	input port clock			0.000
Path 7	∞	5	2	1	y[0]	sum[4]	11.611	5.549	6.062	∞	input port clock			0.000
Path 8	∞	5	2	1	y[0]	sum[7]	11.591	5.640	5.950	∞	input port clock			0.000
Path 9	∞	4	2	1	y[0]	sum[3]	11.307	5.390	5.917	∞	input port clock			0.000

Setup Delays - BA

Yusuf Tekin
040200043

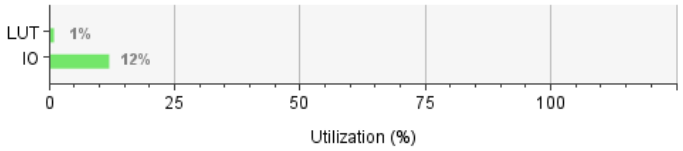
Unconstrained Paths - NONE - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 10	∞	3	2	2	x[4]	sum[7]	2.577	1.723	0.853	-∞	input port clock			0.000
Path 11	∞	3	2	2	x[4]	sum[6]	2.596	1.702	0.893	-∞	input port clock			0.000
Path 12	∞	3	2	2	x[4]	sum[5]	2.636	1.664	0.972	-∞	input port clock			0.000
Path 13	∞	4	2	2	x[4]	sum[4]	2.637	1.664	0.973	-∞	input port clock			0.000
Path 14	∞	4	2	2	x[4]	cout	2.649	1.807	0.842	-∞	input port clock			0.000
Path 15	∞	4	2	2	x[2]	sum[3]	2.652	1.674	0.977	-∞	input port clock			0.000
Path 16	∞	4	2	2	x[2]	sum[2]	2.928	1.646	1.282	-∞	input port clock			0.000
Path 17	∞	4	2	2	x[1]	sum[1]	3.102	1.668	1.434	-∞	input port clock			0.000
Path 18	∞	4	2	2	x[0]	sum[0]	3.351	1.661	1.690	-∞	input port clock			0.000

Hold Delays - BA

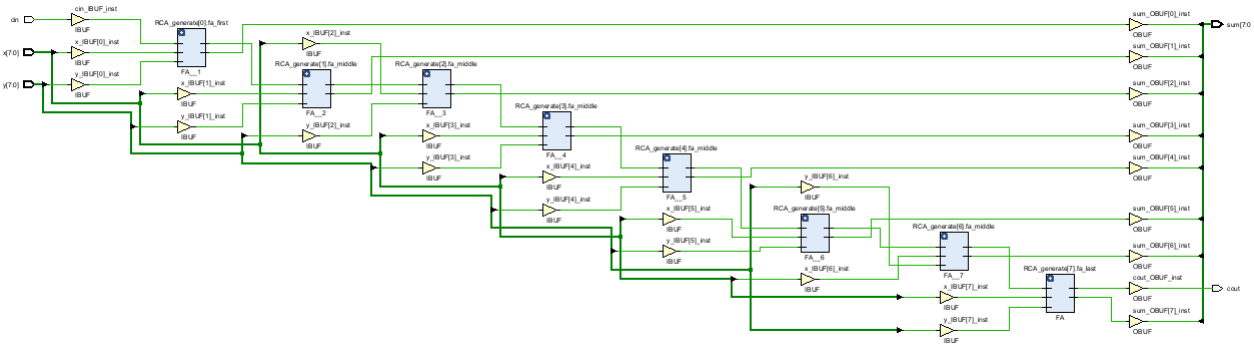
Summary

Resource	Utilization	Available	Utilization %
LUT	8	32600	0.02
IO	25	210	11.90



Utilization Summary - BA

It is observed that there is no change for the behavioral design. This means, Vivado do not apply any optimization to the behavioral design. The optimizations occur after synthesis therefore, any difference cannot be seen in any RTL schematics.

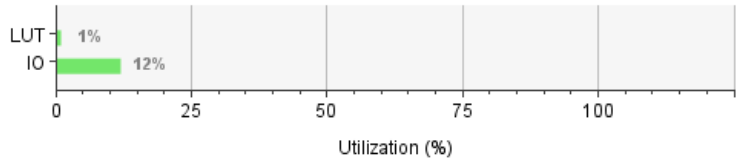


Technology Schematic - parametric RCA

Yusuf Tekin
040200043

Summary

Resource	Utilization	Available	Utilization %
LUT	8	32600	0.02
IO	26	210	12.38



Utilization Summary - parametric RCA

Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total ...	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	∞	10	9	2	y[0]	sum[7]	16.743	5.910	10.833	∞	input port clock			0.000
Path 2	∞	10	9	2	y[0]	cout	16.655	5.686	10.969	∞	input port clock			0.000
Path 3	∞	9	8	2	y[0]	sum[6]	15.736	5.759	9.978	∞	input port clock			0.000
Path 4	∞	8	7	2	y[0]	sum[5]	15.495	5.655	9.840	∞	input port clock			0.000
Path 5	∞	7	6	2	y[0]	sum[4]	14.509	5.515	8.993	∞	input port clock			0.000
Path 6	∞	6	5	2	y[0]	sum[3]	13.912	5.398	8.513	∞	input port clock			0.000
Path 7	∞	5	4	2	y[0]	sum[2]	13.597	5.283	8.314	∞	input port clock			0.000
Path 8	∞	4	3	2	y[0]	sum[1]	12.246	5.175	7.071	∞	input port clock			0.000
Path 9	∞	3	2	2	y[0]	sum[0]	11.175	4.600	6.575	∞	input port clock			0.000

Setup Delays - parametric RCA

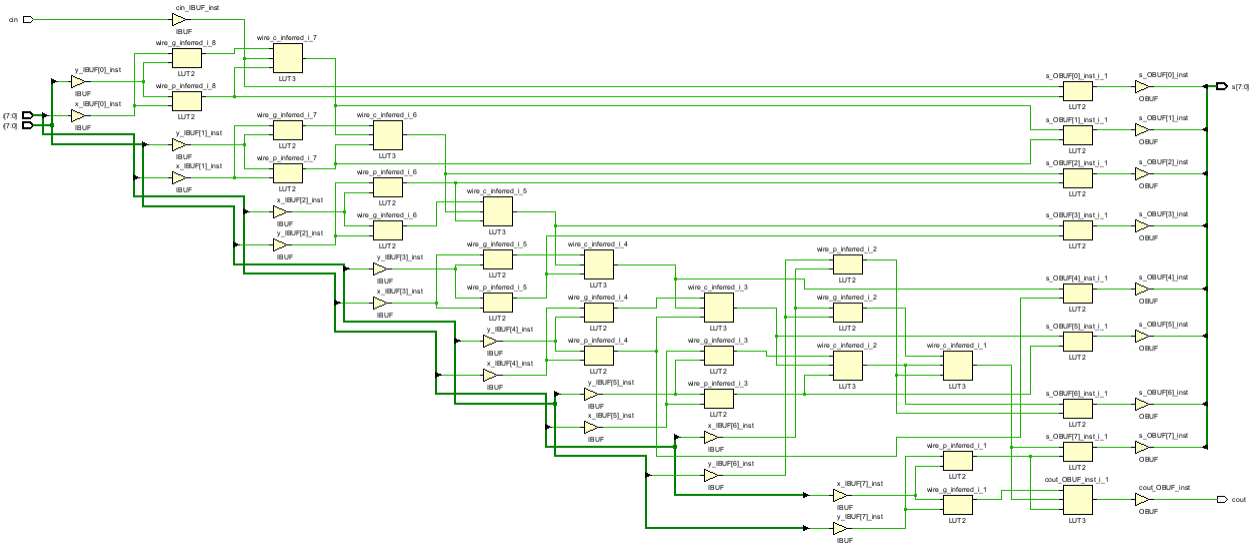
Unconstrained Paths - NONE - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total ...	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 18	∞	3	2	2	cin	sum[0]	2.719	1.508	1.211	-∞	input port clock			0.000
Path 17	∞	3	2	2	x[2]	sum[2]	2.671	1.582	1.089	-∞	input port clock			0.000
Path 16	∞	3	2	2	x[3]	sum[3]	2.635	1.570	1.065	-∞	input port clock			0.000
Path 15	∞	3	2	2	x[1]	sum[1]	2.581	1.612	0.969	-∞	input port clock			0.000
Path 14	∞	3	2	2	x[4]	sum[4]	2.564	1.597	0.967	-∞	input port clock			0.000
Path 13	∞	3	2	2	y[7]	sum[7]	2.550	1.609	0.941	-∞	input port clock			0.000
Path 12	∞	3	2	2	y[7]	cout	2.546	1.554	0.992	-∞	input port clock			0.000
Path 11	∞	3	2	2	y[5]	sum[5]	2.496	1.617	0.879	-∞	input port clock			0.000
Path 10	∞	3	2	2	x[6]	sum[6]	2.333	1.600	0.733	-∞	input port clock			0.000

Hold Delays - parametric RCA

For RCA, the utilization did not change however, technology schematic enlarged by non-optimized full-adders and the setup delays are increased significantly. No difference between RTLs.

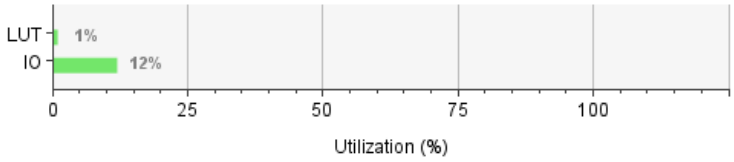
Yusuf Tekin
040200043



Technology Schematic of CLA

Summary

Resource	Utilization	Available	Utilization %
LUT	32	32600	0.10
IO	26	210	12.38



Utilization Summary of CLA

Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total ...	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	∞	11	10	2	y[0]	s[7]	17.861	5.605	12.256	∞	input port clock			0.000
Path 2	∞	11	10	2	y[0]	cout	17.857	5.614	12.243	∞	input port clock			0.000
Path 3	∞	10	9	2	y[0]	s[6]	17.071	5.484	11.587	∞	input port clock			0.000
Path 4	∞	9	8	2	y[0]	s[5]	16.270	5.355	10.915	∞	input port clock			0.000
Path 5	∞	8	7	2	y[0]	s[4]	15.321	5.240	10.081	∞	input port clock			0.000
Path 6	∞	7	6	2	y[0]	s[3]	14.052	5.098	8.954	∞	input port clock			0.000
Path 7	∞	6	5	2	y[0]	s[2]	13.904	4.981	8.922	∞	input port clock			0.000
Path 8	∞	5	4	2	y[0]	s[1]	12.684	4.867	7.817	∞	input port clock			0.000
Path 9	∞	4	3	2	y[0]	s[0]	12.259	4.724	7.535	∞	input port clock			0.000

Setup Delays - CLA

Both resource usage and delays have increased significantly for the CLA design. This is because of the non-optimization of LUT5s into a LUT2 and a LUT3. The number of LUTs increases therefore the path delays increases.

Yusuf Tekin
040200043

7. Adder-Subtractor Circuit

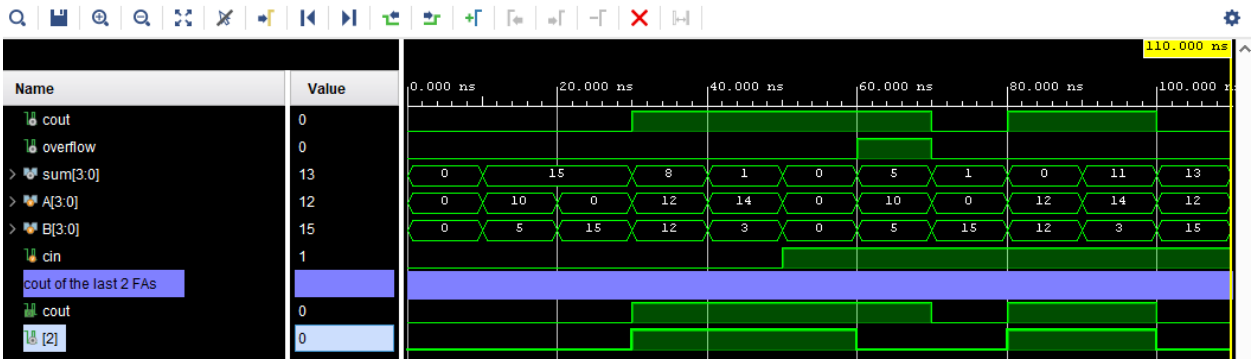
```
module Add_Sub(

    input [3:0] A,
    input [3:0] B,
    input cin,
    output [3:0] sum,
    output cout,
    output overflow
    );

wire [2:0] wire_c;
wire [3:0] wire_fa_b;
genvar i;

generate
    for(i=0; i<4; i = i +1) begin
        assign wire_fa_b[i] = cin ^ B[i];
    end
endgenerate

FA fa0(.x(A[0]), .y(wire_fa_b[0]), .cin(cin), .cout(wire_c[0]), .sum(sum[0]));
FA fa1(.x(A[1]), .y(wire_fa_b[1]), .cin(wire_c[0]), .cout(wire_c[1]), .sum(sum[1]));
FA fa2(.x(A[2]), .y(wire_fa_b[2]), .cin(wire_c[1]), .cout(wire_c[2]), .sum(sum[2]));
FA fa3(.x(A[3]), .y(wire_fa_b[3]), .cin(wire_c[2]), .cout(cout), .sum(sum[3]));

assign overflow = wire_c[2] ^ cout;

endmodule
```

Add-Sub Verilog Code

Yusuf Tekin
040200043

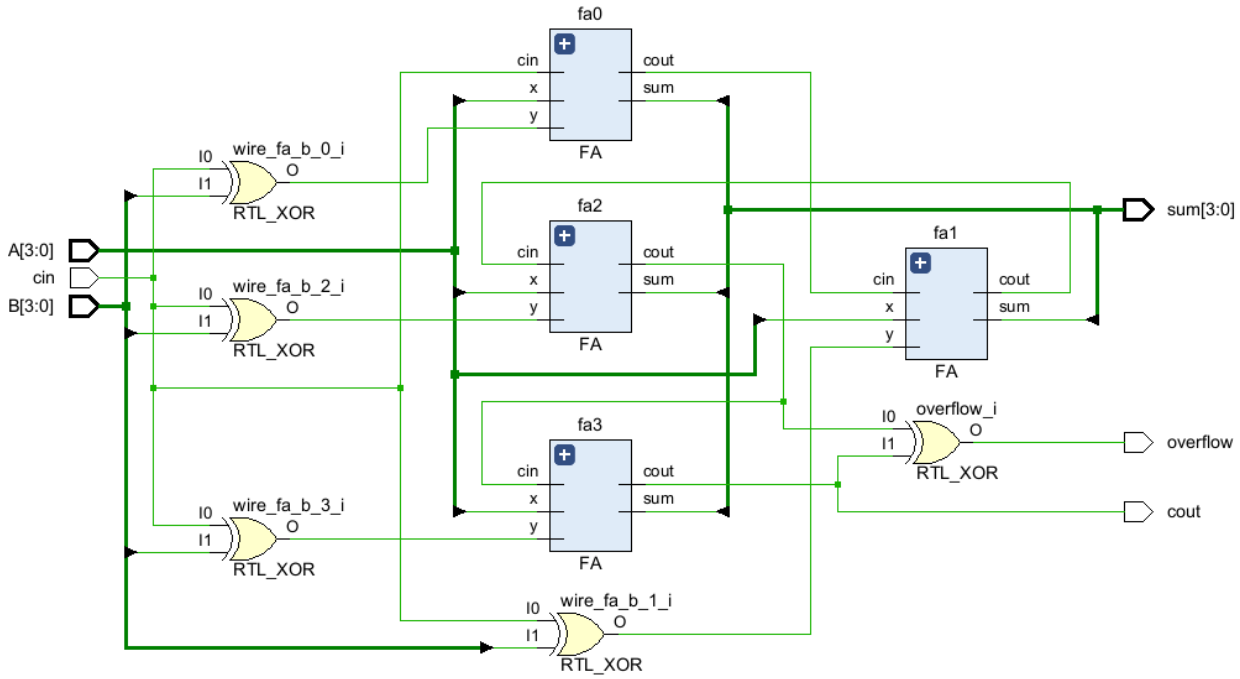
```
`timescale 1ns / 1ps
module Add_Sub_tb();

wire cout, overflow;
wire [3:0] sum;
reg [3:0] A, B;
reg cin;

Add_Sub uut(
    .A(A),
    .B(B),
    .cin(cin),
    .cout(cout),
    .sum(sum),
    .overflow(overflow));

initial begin
    A = 4'b0000; B = 4'b0000; cin = 1'b0; #10;
    A = 4'b1010; B = 4'b0101; cin = 1'b0; #10;
    A = 4'b0000; B = 4'b1111; cin = 1'b0; #10;
    A = 4'b1100; B = 4'b1100; cin = 1'b0; #10;
    A = 4'b1110; B = 4'b0011; cin = 1'b0; #10;
    A = 4'b0000; B = 4'b0000; cin = 1'b1; #10;
    A = 4'b1010; B = 4'b0101; cin = 1'b1; #10;
    A = 4'b0000; B = 4'b1111; cin = 1'b1; #10;
    A = 4'b1100; B = 4'b1100; cin = 1'b1; #10;
    A = 4'b1110; B = 4'b0011; cin = 1'b1; #10;
    A = 4'b1100; B = 4'b1111; cin = 1'b1; #10;
    $finish();
end
endmodule
```

Add-Sub Testbench Code

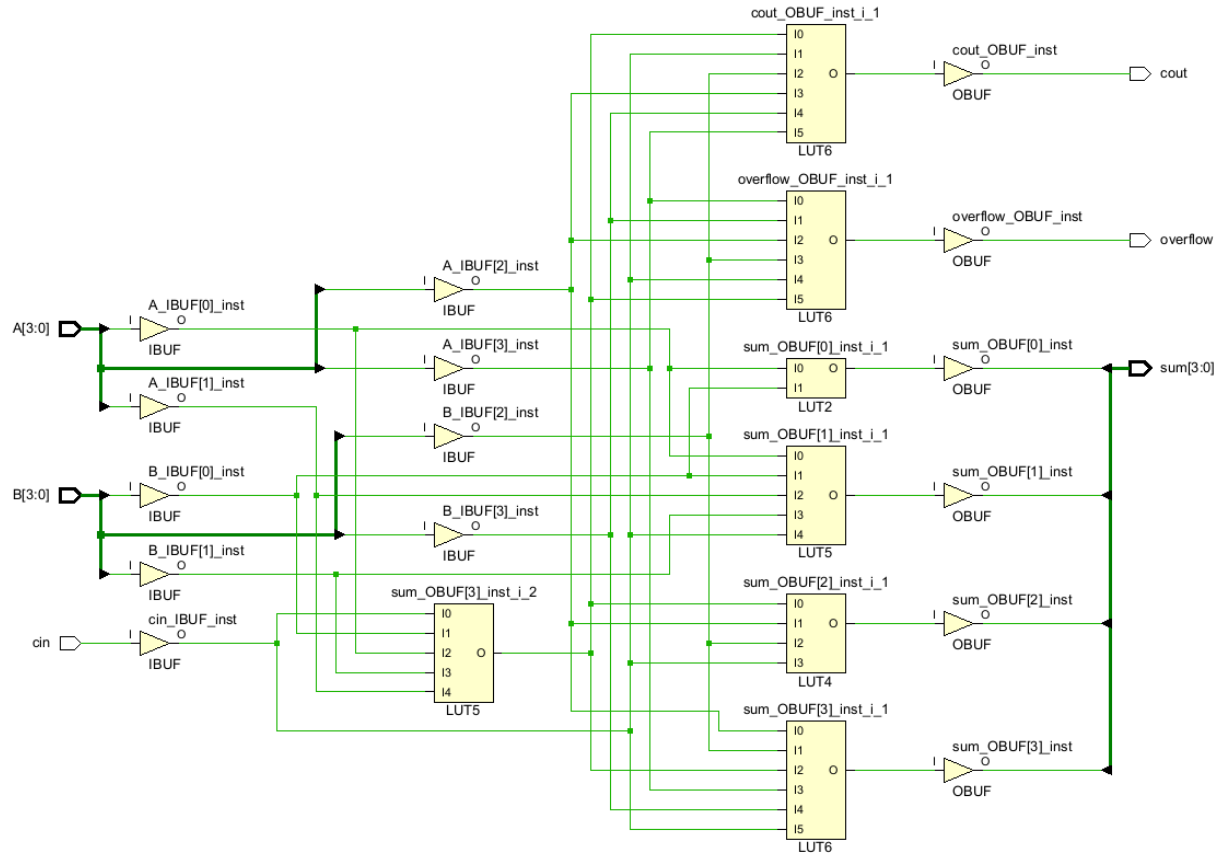


Add-Sub Behavioral Simulation

Yusuf Tekin
040200043



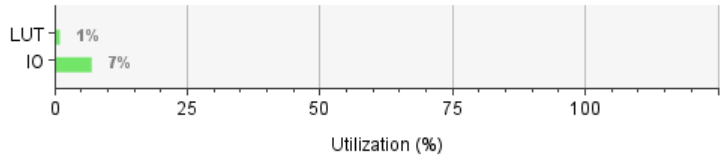
Add-Sub RTL Schematic



Add-Sub Technology Schematic

Summary

Resource	Utilization	Available	Utilization %
LUT	6	32600	0.02
IO	15	210	7.14



Add-Sub Utilization Summary

Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total ...	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 1	∞	4	3	4	A[0]	overflow	10.603	5.287	5.316	∞	input port clock			0.000
Path 2	∞	4	3	4	A[0]	sum[2]	10.563	5.272	5.291	∞	input port clock			0.000
Path 3	∞	4	3	4	A[0]	cout	9.553	5.283	4.270	∞	input port clock			0.000
Path 4	∞	4	3	4	A[0]	sum[3]	9.551	5.265	4.286	∞	input port clock			0.000
Path 5	∞	3	2	3	A[0]	sum[1]	9.096	5.388	3.709	∞	input port clock			0.000
Path 6	∞	3	2	3	B[0]	sum[0]	8.962	5.122	3.840	∞	input port clock			0.000

Setup Delays - Add-Sub

Unconstrained Paths - NONE - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total ...	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exception	Clock Uncertainty
Path 12	∞	3	2	4	A[2]	sum[2]	2.749	1.517	1.232	-∞	input port clock			0.000
Path 11	∞	3	2	3	B[3]	overflow	2.739	1.546	1.192	-∞	input port clock			0.000
Path 10	∞	3	2	3	A[0]	sum[0]	2.607	1.529	1.079	-∞	input port clock			0.000
Path 9	∞	3	2	2	A[1]	sum[1]	2.590	1.604	0.986	-∞	input port clock			0.000
Path 8	∞	3	2	6	cin	cout	2.355	1.528	0.827	-∞	input port clock			0.000
Path 7	∞	3	2	4	A[2]	sum[3]	2.249	1.510	0.739	-∞	input port clock			0.000

Hold Delays - Add-Sub

8. Research

DSP Blocks: DSP (Digital Signal Processing) blocks are build-in hardware units in FPGAs designed for compute intensive operations like Multiplication, MAC (Multiply-Accumulate), Filtering, FFT (Fast Fourier Transform). However, operands must fit 25x18-bit multipliers in Xilinx 7-Series FPGAs. To be able to use DPS blocks the multiplication **(*)** operation must be performed. After the synthesis, open “Utilization Report” to look DSP block usage. It can also be seen in the technology schematic.[1] Here is a RTL code example:

```
module mult_unsigned (clk, A, B, RES);

parameter WIDTHA = 16;
parameter WIDTHB = 24;
input clk;
input [WIDTHA-1:0] A;
input [WIDTHB-1:0] B;
output [WIDTHA+WIDTHB-1:0] RES;

reg [WIDTHA-1:0] rA;
reg [WIDTHB-1:0] rB;
reg [WIDTHA+WIDTHB-1:0] M [3:0];

integer i;
always @(posedge clk)
begin
rA <= A;
rB <= B;
M[0] <= rA * rB;
for (i = 0; i < 3; i = i+1)
M[i+1] <= M[i];
end

assign RES = M[3];

endmodule
```

Multiply-Accumulate (MAC) Operation

Fixed-Point Representation in FPGA Design: Fixed-point representation is a way of encoding real numbers as integers, where the position of the decimal point is fixed. This design is commonly used in FPGA designs due to the efficiency in arithmetic operations. In binary 0110.1010 represents 6.625 in decimal. Compared with the floating-point design, fixed point design has a lower power consumption, faster operation speed in simpler operations, minimalization in LUT and DSP usage and easier design/implement. However, it has limited precision which makes it unsuitable for some complex computation and scientific calculations.[2]

References

1- Vivado Design Suite User Guide: Synthesis (UG901), Multipliers, DSP Block Implementations
2- Vivado Design Suite User Guide: Logic Simulation (UG900), Fixed and Floating Point Packages