

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

EEE 468 (January 2024)
VLSI Laboratory

Final Project Report

Section: G1 Group: 03

16 Bit Skip Carry Adder: High-Performance Arithmetic Circuit

Course Instructors:

Nafis Sadik, Lecturer
Rafid Hassan Palash, Part-Time Lecturer

Signature of Instructor: _____

Signature of Instructor: _____

Academic Honesty Statement:

"In signing this statement, We hereby certify that the work on this project is our own and that we have not copied the work of any other students (past or present), and cited all relevant sources while completing this project. We understand that if we fail to honor this agreement, We will each receive a score of ZERO for this project and be subject to failure of this course."

Signature: _____ Full Name: Md Mostaqul Islam Student ID: 1906070	Signature: _____ Full Name: Md Abu Sayed Chowdhury Student ID: 1906076
Signature: _____ Full Name: Md Sharif Uddin Student ID: 1906077	Signature: _____ Full Name: Md Liton Ali Student ID: 1906080
Signature: _____ Full Name: Shakil Ahammad Student ID: 1806052	Signature: _____

Table of Contents

1	Abstract.....	1
2	Introduction	1
3	Design	1
3.1	Problem Formulation	1
3.2	Design Method.....	2
3.3	Circuit Diagram	3
4	Implementation	4
4.1	RTL Code	4
4.2	Verification	6
4.2.1	Direct Testbench	6
Output from Cadence:.....		7
Waveform from Cadence:.....		8
4.2.2	Layered	9
4.2.3	-----Displaying Coverage Results-----	23
4.3	Synthesis	31
4.3.1	TCL Code	31
4.3.2	SDC Code	32
4.3.3	Module	33
4.3.4	Submodule	33
4.3.5	Reports	35
4.3.6	Comparison among Low , Medium , and High SYN_EFF ,MAP_EFF & OPT_EFF 42	
4.3.7	Timing Report.....	43
4.4	Physical Design.....	45
4.4.1	Design Import	45
4.4.2	Floor planning.....	45
4.4.3	Creating Power Mesh.....	47
4.4.4	Placement.....	48
4.4.5	Clock Tree Synthesis (CTS)	49
4.4.6	Running Nano route.....	49
4.4.7	Post-Route timing and SI Optimization.....	51
4.4.8	Adding filler cell and metal filling.....	52
4.4.9	Exporting Data.....	54

4.4.10	Run DRC.....	55
4.4.11	Utilization Report.....	56
4.4.12	Power Report	56
5	Design Analysis and Optimization	57
5.1	Result Analysis	57
5.2	Novelty.....	58
5.2.1	Signed Number Consideration and Flag Addition	58
5.2.2	Coverage	58
5.3	Design Optimization	59
5.3.1	Delay minimization, Higher Speed, and Lower Latency	59
5.3.2	Minimum Power Consumption.....	63
5.3.3	Area Minimization	65
5.3.4	Overall Optimization of Block Size(k)	66
5.3.5	Maximum Utilization of Chip Area.....	67
5.4	Design Considerations	68
6	Reflection on Individual and Team work	68
6.1	Individual Contribution of Each Member.....	69
6.2	Mode of Team Work.....	69
6.3	Log Book of Project Implementation	70
7	Summary.....	70
8	Conclusion.....	71
9	References	72

Table of Figures:

Figure 1 : Synthesis Module for Carry Skip Adder	33
Figure 2 : RCA0 Submodule.....	33
Figure 3 : RCA1 Submodule	34
Figure 4 : Skip0 Submodule	34
Figure 5 : Full Adder Submodule	34
Figure 6 : Mux21 Submodule	35
Figure 7: Percentage power consumption for different area.....	35
Figure 8 : Comparison of power consumption with and without clock.....	36
Figure 9 : Detail power consumption table of the components	37
Figure 10 : A pie-chart of detail power consumption table of the components	38
Figure 11 : Detail area consumption table of the components.....	39
Figure 12 : Detail power consumption table of the instances	40
Figure 13 : Bar chart of the different components and modules.....	40
Figure 14 : Area of different submodules of adder and mux	41
Figure 15 : Comparision of the Syn_EFF, MAP_EFF, OPT_EFF for Low, Medium & High respectively from up to down	42
Figure 16 : Design Import Window	45
Figure 17 : Floorplan Specification	46
Figure 18 : Creating Pin in "Pin Editor"	46
Figure 19 : Add stripes input	47
Figure 20 : Adding stripes for power planning	47
Figure 21 : M1 horizontal rail created	48
Figure 22 : Layout after optimizing placement.....	48
Figure 23 : Command Line Report after optimizing placement	49
Figure 24 : Timing reports	49
Figure 25 : Layout View after Nano route '	50
Figure 26 : SI timing summary	51
Figure 27 : Geometry Verification.....	51
Figure 28 :Adding filler Cell.....	52
Figure 29 : Layout view after adding filler cell	52
Figure 30 : List of filler cell.....	53
Figure 31 :Layout view after adding metal fill	53
Figure 32: Verify Geometry after metal fill.....	54
Figure 33: Verify Connectivity	54
Figure 34: Final Layout at virtuoso	55
Figure 35: PVS Debug window showing no error	55
Figure 36 : DRC Check with clean result	56
Figure 37 :Carry Skip Chain.....	59
Figure 38 : Delay Vs Block size (k).....	60
Figure 39 : Delay vs Block Size (k) Variying t_PG	60
Figure 40:; Delay vs Block Size (k) Variying t_carry	61
Figure 41 : : Delay vs Block Size (k) Variying t_MUX	61
Figure 42: Timing report for k=2.....	62
Figure 43: Timing report for k=4.....	62
Figure 44 : Timing report for k=8.....	63
Figure 45: Power Report for k=8	63
Figure 46 : Power Report for k=4	64
Figure 47 : Power Report for k=2	64
Figure 48 : Area Report for k=8	65
Figure 49 : Area Report for k=4	65
Figure 50 : Area Report for k=2	66
Figure 51 : Normalized Power, Area, & Output Arrival Time	66

1 Abstract

The 16-bit Carry Skip Adder (CSKA) is a high-performance arithmetic circuit designed to enhance the speed of addition operations in digital systems. Traditional ripple-carry adders suffer from high latency due to sequential carry propagation across all bits, making them inefficient for large bit-width operations. The CSKA addresses this limitation by employing a block-based structure with skip logic, allowing carry signals to bypass unnecessary propagation stages and significantly improving speed. This project focuses on designing, simulating, and implementing a 16-bit CSKA capable of performing both signed and unsigned additions, controlled by a dedicated signal.

Key aspects of the project include the use of propagate-generate logic for optimal performance, a comprehensive verification environment, and PPA (Power, Performance, Area) optimization. The final implementation meets stringent design constraints, achieving clean design rule checks (DRC) and layout-versus-schematic (LVS) compliance. The results demonstrate a robust and efficient adder architecture suitable for integration into modern high-speed digital systems.

2 Introduction

Efficient arithmetic circuits are essential for modern digital systems, where speed, power efficiency, and compactness are critical requirements. Traditional ripple-carry adders, though simple, exhibit high latency due to sequential carry propagation, making them inadequate for high-speed applications involving large bit-width operations.

The 16-bit Carry Skip Adder (CSKA) addresses these limitations by employing a block-based structure with skip logic, enabling faster carry propagation and reduced latency. This project focuses on designing and implementing a 16-bit CSKA with support for both signed and unsigned addition operations, controlled by a signal. The design leverages propagate-generate logic for optimized carry handling, ensuring correctness and high performance.

The development process includes simulation, synthesis, and physical implementation, with a focus on PPA optimization. Rigorous verification ensures functional accuracy, while adherence to industry-standard DRC and LVS requirements validates the design for manufacturability. This project demonstrates a high-performance, energy-efficient solution for modern digital arithmetic operations.

3 Design

3.1 Problem Formulation

The primary objective of this project is to design a high-performance 16-bit Carry Skip Adder (CSKA) to address the inherent latency issues of ripple carry adders. By leveraging block-based skip logic and propagate-generate logic, the design aims to achieve faster addition operations for both signed and unsigned binary numbers.

Key goals include:

- ✓ Reducing propagation delay using optimized block sizes and skip logic.
- ✓ Supporting both signed and unsigned additions via a control signal (S/U^-).
- ✓ Ensuring correctness across all operational scenarios.
- ✓ Achieving optimal Power, Performance, and Area (PPA) metrics through synthesis and physical design optimization.

3.2 Design Method

The design methodology for the 16-bit CSKA is structured into the following steps:

- **Block Division:**

Divide the 16-bit input into smaller blocks for parallel processing. Each block computes sums and generates a carry independently, reducing the delay caused by sequential propagation.

- **Ripple Carry Adder within Blocks:**

Each block uses a ripple carry adder to compute the sum and carry for its respective bits. Intermediate carry generation is based on the equations:

$$Sum_i = A_i \oplus B_i \oplus Cin$$

$$Carry_i = (A_i \cdot B_i) + (Cin \cdot (A_i \oplus B_i))$$

- **Carry Propagation Logic:**

Each block evaluates the propagate condition to determine if a carry can skip the block:

$$P_{block} = P_0 \cdot P_1 \cdot \dots \cdot P_{n-1}, \quad \text{where } P_i = A_i \oplus B_i$$

If all bits in a block satisfy the propagate condition, the carry skips to the next block.

- **Carry Skip Logic:**

- **Optimization:**

- **Delay Optimization:**

By adjusting the number of blocks (M) and their sizes, the delay is minimized using the derived equation:

$$T = 2Pk_1 + (M - 2)k_2$$

where $P = n/M$, k_1 is ripple carry delay, and k_2 is multiplexer delay. Optimized $M = \sqrt{\frac{2nk_1}{k_2}}$.

- **Power Optimization:**

Minimizing the number of blocks reduces power consumption, which increases linearly with the block size.

Simulation and Verification:

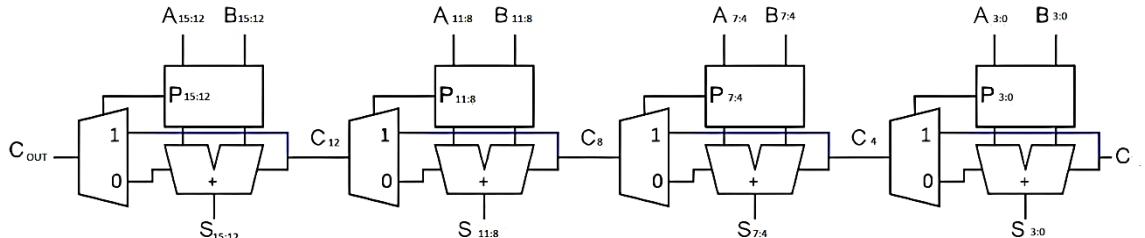
- The design is simulated and verified using directed and layered testbenches to ensure functionality and performance.

Physical Implementation:

- Synthesis, Place, and Route (PnR) are performed while adhering to design constraints, optimizing area utilization, and meeting timing and power budgets.

3.3 Circuit Diagram

Below is the block diagram representation of the 16-bit Carry Skip Adder:



1. Top-Level Structure:

- Input: A[15:0], B[15:0], Cin, and control signal S/U⁻.
- Output: Sum S[15:0] and Carry Out Cout.
- Block-based structure with ripple carry adders and skip logic.

2. Internal Blocks:

- Ripple Carry Adders: Perform addition within each block.
 - Skip Logic: Evaluates the propagate condition and determines carry bypass

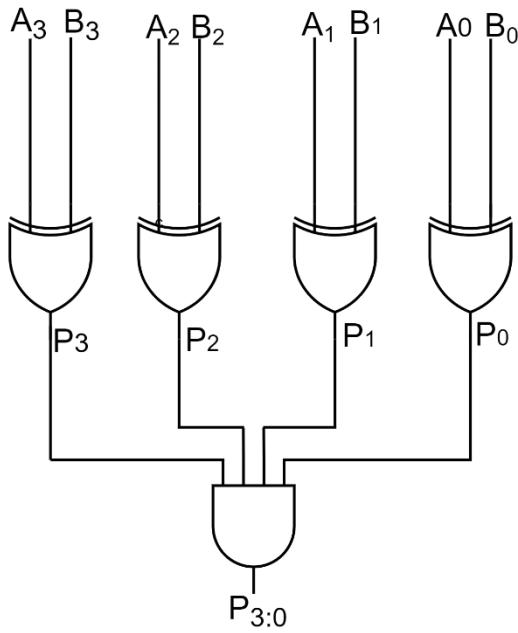


Figure 1 : Carry Propagate Logic

4 Implementation

4.1 RTL Code

```
// Code your design here
module csa_top(
    input wire [15:0] A,
    input wire [15:0] B,
    input wire Cin,
    output wire [15:0] Sum,
    output wire Cout
);
wire cout3,cout2,cout1,cout0;
wire mux_output1,mux_output2,mux_output3;
// RCA4 is 4 bit ripple carry adder
RCA4 rca0(.sum(Sum[3:0]), .cout(cout0), .a(A[3:0]), .b(B[3:0]),.cin(Cin));
RCA4 rca1(.sum(Sum[7:4]), .cout(cout1), .a(A[7:4]), .b(B[7:4]),.cin(mux_output1));
RCA4 rca2(.sum(Sum[11:8]), .cout(cout2), .a(A[11:8]), .b(B[11:8]),.cin(mux_output2));
RCA4 rca3(.sum(Sum[15:12]), .cout(cout3), .a(A[15:12]), .b(B[15:12]),.cin(mux_output3));
// Carry Skip Logic
SkipLogic skip0(.cin_next(mux_output1), .a(A[3:0]), .b(B[3:0]),.cin(Cin),.cout(cout0));
SkipLogic skip1(.cin_next(mux_output2),.a(A[7:4]),.b(B[7:4]),.cin(mux_output1),.cout(cout1));
SkipLogic skip2(.cin_next(mux_output3),.a(A[11:8]),.b(B[11:8]),.cin(mux_output2),.cout(cout2));
SkipLogic skip3(.cin_next(Cout),.a(A[15:12]),.b(B[15:12]),.cin(mux_output3),.cout(cout3));
endmodule
```

```

module FA(input wire a,
           input wire b,
           input wire cin,
           output wire sum,
           output wire cout);
wire w0, w1, w2;

xor (w0, a, b); // a xor b
xor (sum, w0, cin); // a xor b xor cin

and (w1, w0, cin); // (a xor b) cin
and (w2, a, b);
or (cout, w1, w2); // ab + (a xor b) cin
endmodule

module RCA4(input wire [3:0] a,
             input wire [3:0] b,
             input wire cin,
             output wire [3:0] sum,
             output wire cout
            );
wire [3:1] c;

FA fa0(.sum(sum[0]), .cout(c[1]), .a(a[0]), .b(b[0]),
FA fa1(.sum(sum[1]), .cout(c[2]), .a(a[1]), .b(b[1]),
FA fa2(.sum(sum[2]), .cout(c[3]), .a(a[2]), .b(b[2]),
FA fa3(.sum(sum[3]), .cout(cout), .a(a[3]), .b(b[3]),
endmodule

```

```

module SkipLogic(
           input wire [3:0] a, b,
           input wire cin, cout,
           output wire cin_next
          );

wire p0, p1, p2, p3, p;

xor (p0, a[0], b[0]);
xor (p1, a[1], b[1]);
xor (p2, a[2], b[2]);
xor (p3, a[3], b[3]);

and (p, p0, p1, p2, p3);
mux_21
muxs(.a(cout),.b(cin),.s(p),.o(cin_next));

endmodule

module mux_21(
           input wire a,
           input wire b,
           input wire s,
           output wire o
          );
wire s_bar,w1,w2;

not (s_bar,s);
and(w1,s_bar,a);
and(w2,s,b);
or(o,w1,w2);

endmodule

```

4.2 Verification

4.2.1 Direct Testbench

```
module csa_stimulus;
reg [15:0] A;//2's complement input
reg [15:0] B;//2's complement input
reg Cin;
wire[15:0] Sum;//2's complement output
wire Cout;

reg[15:0] Sum_temp;
reg Cout_temp;
reg N;//negative flag
reg Z;//zero flag
reg C;//carry flag
reg V;//Overflow flag
csa_top DUT(
    .A(A),
    .B(B),
    .Cin(Cin),
    .Sum(Sum),
    .Cout(Cout) );
```

```
initial begin
    run();
    #10
    repeat(30) begin
        run();
        #10;
    end
    $finish;
end
```

```
initial begin
    $dumpfile("dump.vcd");
    $dumpvars;
end
task run();
begin
    A=$random % 65536;
    B=$random % 65536;
    Cin=$random % 2;
    //A=63070;
    //B=47030;
    //Cin=1;
    {Cout_temp, Sum_temp}= A+B+Cin;
    #5;
    if(Sum[15])//Negative flag set
        assign N=1'b1;
    else
        assign N=1'b0;
    if(Sum)//Zero flag set
        assign Z=1'b0;
    else
        assign Z=1'b1;
    if((A[15]==B[15]) &
    (A[15]!=Sum[15]))//Overflow flag set
        assign V=1'b1;
    else
        assign V=1'b0;
```

```
#5;
if(Sum != Sum_temp & Cout_temp!=Cout)
    $display("time=%d Failed : A=%d B=%d Cin=%d Expected Sum=%d Resulted Sum=%d
Expected Cout=%d Resulted Cout=%d N=%d Z=%d
V=%d",$time,A,B,Cin,Sum,Sum_temp,Cout,Cout_temp,N,Z,V);
else
    $display("time=%d Passed : A=%d B=%d Cin=%d Expected Sum=%d Resulted Sum=%d Expected
Cout=%d Resulted Cout=%d N=%d Z=%d
V=%d",$time,A,B,Cin,Sum,Sum_temp,Cout,Cout_temp,N,Z,V);
end
endtask
endmodule
```

Output from Cadence:

```
[vlsil3@CadenceServer3 Directed Testbench]$ ncsim csa_stimulus
ncsim(64): 15.10-s015: (c) Copyright 1995-2016 Cadence Design Systems, Inc.
ncsim> run
ncsim: *W,DVEXACC2: some objects excluded from $dumpvars due to -access -R.
      File: ./csa_tb.v, line = 38, pos = 12
      Scope: csa_stimulus
      Time: 0 FS + 0

time=          10  Passed : A=13604 B=24193 Cin=1 Expected Sum=37798 Resulted Sum=37798 Expected Cout=0 Resulte
d Cout=0 N=1 Z=0 V=1
time=          30  Passed : A=22115 B=31501 Cin=1 Expected Sum=53617 Resulted Sum=53617 Expected Cout=0 Resulte
d Cout=0 N=1 Z=0 V=1
time=          50  Passed : A=33893 B=21010 Cin=1 Expected Sum=54904 Resulted Sum=54904 Expected Cout=0 Resulte
d Cout=0 N=1 Z=0 V=0
time=          70  Passed : A=52493 B=61814 Cin=1 Expected Sum=48772 Resulted Sum=48772 Expected Cout=1 Resulte
d Cout=1 N=1 Z=0 V=0
time=          90  Passed : A=22509 B=63372 Cin=1 Expected Sum=20346 Resulted Sum=20346 Expected Cout=1 Resulte
d Cout=1 N=0 Z=0 V=0
time=         110  Passed : A= 9414 B=33989 Cin=0 Expected Sum=43403 Resulted Sum=43403 Expected Cout=0 Resulte
d Cout=0 N=1 Z=0 V=0
time=         130  Passed : A=63461 B=29303 Cin=0 Expected Sum=27228 Resulted Sum=27228 Expected Cout=1 Resulte
d Cout=1 N=0 Z=0 V=0
time=         150  Passed : A=56207 B=27122 Cin=0 Expected Sum=17793 Resulted Sum=17793 Expected Cout=1 Resulte
d Cout=1 N=0 Z=0 V=0
time=         170  Passed : A=31464 B=20165 Cin=0 Expected Sum=51629 Resulted Sum=51629 Expected Cout=0 Resulte
d Cout=0 N=1 Z=0 V=1
time=         190  Passed : A=10429 B=22573 Cin=1 Expected Sum=33003 Resulted Sum=33003 Expected Cout=0 Resulte
d Cout=0 N=1 Z=0 V=1
time=         210  Passed : A=25187 B=34570 Cin=0 Expected Sum=59757 Resulted Sum=59757 Expected Cout=0 Resulte
d Cout=0 N=1 Z=0 V=0
time=         230  Passed : A= 8480 B=17834 Cin=1 Expected Sum=26315 Resulted Sum=26315 Expected Cout=0 Resulte
d Cout=0 N=0 Z=0 V=0
time=         250  Passed : A=16022 B=47123 Cin=1 Expected Sum=63146 Resulted Sum=63146 Expected Cout=0 Resulte
d Cout=1 N=1 Z=0 V=0
time=         270  Passed : A=54867 B=56683 Cin=1 Expected Sum=46015 Resulted Sum=46015 Expected Cout=1 Resulte
d Cout=0 N=1 Z=0 V=1
time=         290  Passed : A=18946 B=16046 Cin=1 Expected Sum=34993 Resulted Sum=34993 Expected Cout=0 Resulte
d Cout=0 N=1 Z=0 V=1
time=         310  Passed : A=29391 B=18723 Cin=0 Expected Sum=48114 Resulted Sum=48114 Expected Cout=0 Resulte
d Cout=0 N=0 Z=0 V=0
time=         330  Passed : A= 2762 B=19516 Cin=0 Expected Sum=22278 Resulted Sum=22278 Expected Cout=0 Resulte
d Cout=1 N=0 Z=0 V=0
time=         350  Passed : A=24970 B=45889 Cin=0 Expected Sum= 5323 Resulted Sum= 5323 Expected Cout=1 Resulte
d Cout=0 N=1 Z=0 V=0
time=         370  Passed : A=62328 B= 4745 Cin=1 Expected Sum= 1538 Resulted Sum= 1538 Expected Cout=1 Resulte
d Cout=0 N=0 Z=0 V=0
time=         390  Passed : A= 8480 B=17834 Cin=1 Expected Sum=26315 Resulted Sum=26315 Expected Cout=0 Resulte
d Cout=0 N=0 Z=0 V=0
time=         410  Passed : A=16022 B=47123 Cin=1 Expected Sum=63146 Resulted Sum=63146 Expected Cout=0 Resulte
d Cout=0 N=1 Z=0 V=0
time=         430  Passed : A=54867 B=56683 Cin=1 Expected Sum=46015 Resulted Sum=46015 Expected Cout=1 Resulte
d Cout=0 N=1 Z=0 V=1
time=         450  Passed : A=29391 B=18723 Cin=0 Expected Sum=48114 Resulted Sum=48114 Expected Cout=0 Resulte
d Cout=0 N=1 Z=0 V=1
time=         470  Passed : A=2762 B=19516 Cin=0 Expected Sum=22278 Resulted Sum=22278 Expected Cout=0 Resulte
d Cout=1 N=0 Z=0 V=0
time=         490  Passed : A=24970 B=45889 Cin=0 Expected Sum= 5323 Resulted Sum= 5323 Expected Cout=1 Resulte
d Cout=0 N=1 Z=0 V=0
time=         510  Passed : A=62328 B= 4745 Cin=1 Expected Sum= 1538 Resulted Sum= 1538 Expected Cout=1 Resulte
d Cout=0 N=1 Z=0 V=0
time=         530  Passed : A=18946 B=16046 Cin=1 Expected Sum=34993 Resulted Sum=34993 Expected Cout=0 Resulte
d Cout=1 N=0 Z=0 V=0
time=         550  Passed : A=29391 B=18723 Cin=0 Expected Sum=48114 Resulted Sum=48114 Expected Cout=1 Resulte
d Cout=0 N=0 Z=0 V=0
time=         570  Passed : A=2762 B=19516 Cin=0 Expected Sum=22278 Resulted Sum=22278 Expected Cout=0 Resulte
d Cout=1 N=1 Z=0 V=0
time=         590  Passed : A=24970 B=45889 Cin=0 Expected Sum= 5323 Resulted Sum= 5323 Expected Cout=1 Resulte
d Cout=0 N=1 Z=0 V=0
time=         610  Passed : A=62328 B= 4745 Cin=1 Expected Sum= 1538 Resulted Sum= 1538 Expected Cout=1 Resulte
d Cout=1 N=0 Z=0 V=0
Simulation complete via $finish(1) at time 620 NS + 0
./csa_tb.v:32  $finish;
ncsim> exit
[vlsil3@CadenceServer3 Directed Testbench]$
```

Figure 2 : Direct Testbench Output from Cadence

Waveform from Cadence:

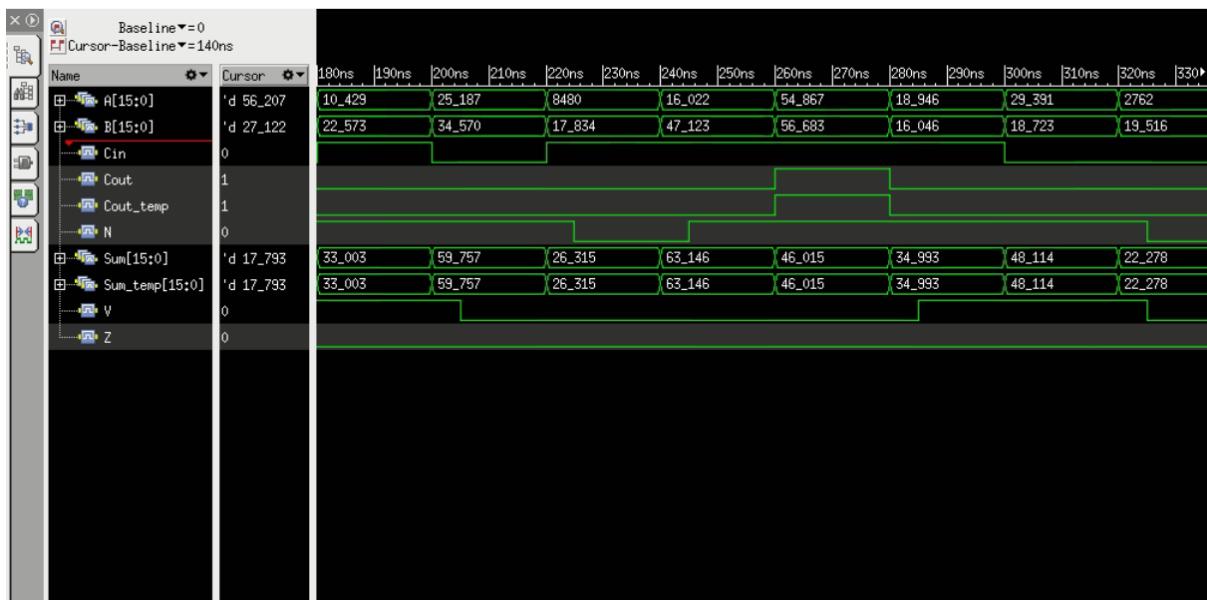
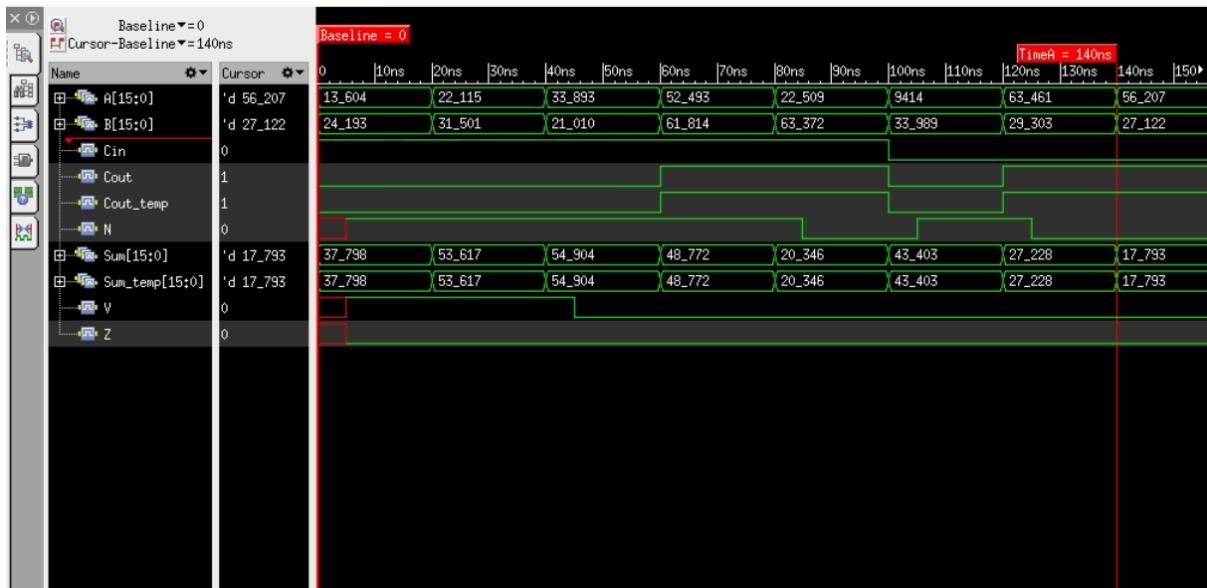


Figure 3 : Direct Testbench Output Waveform of Cadence

4.2.2 Layered

<pre>transaction.sv class transaction; rand bit [15:0] A; rand bit [15:0] B; //rand bit S; rand bit Cin; bit[15:0] Sum; bit Cout; endclass:transaction testcase01.sv `include "environment.sv" program test(input int count, csa_if csaif); environment env; class testcase01 extends transaction; constraint c_s { A inside {[0:65535]}; B inside {[0:65535]}; Cin inside {[0:1]}; //S inside {[0:1]}; //maximizing coverage endclass:testcase01 </pre>	<pre>initial begin //int seed = \$urandom(2998); testcase01 testcase01handle; testcase01handle=new(); env=new(csaif); env.gen.custom_trans=testcase01handle; env.main(count); end endprogram:test interface.sv interface csa_if(input clk); logic [15:0] A,B; logic Cin; logic [15:0] Sum; logic Cout; clocking driver_cb @(negedge clk); default input #1 output #1; output A,B,Cin; endclocking clocking mon_cb @(negedge clk); default input #1 output #1; input A,B,Cin; input Sum,Cout; endclocking modport DRIVER (clocking driver_cb, input clk); modport MONITOR (clocking mon_cb, input clk); </pre>
---	---

```

generator.sv
`include "transaction.sv"
class generator;
mailbox gen2driv;
transaction g_trans, custom_trans;

function new(mailbox gen2driv);
    this.gen2driv=gen2driv;
endfunction

task main(input int count);
repeat(count) begin
    g_trans=new();
    g_trans=new custom_trans;
    assert(g_trans.randomize());

    gen2driv.put(g_trans);
end
endtask:main

endclass:generator

```

driver.sv

```

class driver;
mailbox gen2driv, driv2sb;
virtual csa_if.DRIVER csaif;
transaction d_trans;
event driven;

function new(mailbox gen2driv, driv2sb , virtual
csa_if.DRIVER csaif, event driven);
    this.gen2driv=gen2driv;
    this.csaif=csaif;
    this.driven=driven;
    this.driv2sb=driv2sb;
endfunction

task main(input int count);
repeat(count) begin
    d_trans=new();
    gen2driv.get(d_trans);

    @(csaif.driver_cb);
    csaif.driver_cb.A <= d_trans.A;
    csaif.driver_cb.B <= d_trans.B;
//csaif.driver_cb.S <= d_trans.S;
    csaif.driver_cb.Cin <= d_trans.Cin;
    driv2sb.put(d_trans);
    -> driven;
end

endtask:main

endclass:driver

```

monitor.sv

```

class monitor;
mailbox mon2sb;
virtual csa_if.MONITOR csaif;
transaction m_trans;
event driven;

function new(mailbox mon2sb, virtual csa_if.MONITOR
csaif, event driven);
    this.mon2sb=mon2sb;
    this.csaif=csaif;
    this.driven=driven;
endfunction

task main(input int count);
@(driven);
@(csaif.mon_cb);
repeat(count) begin
    m_trans=new();
    @(posedge csaif.clk);
    m_trans.Sum=csaif.mon_cb.Sum;
    m_trans.Cout=csaif.mon_cb.Cout;
    mon2sb.put(m_trans);
end
endtask:main

```

endclass:monitor

environment.sv

```

`include "generator.sv"
`include "driver.sv"
`include "monitor.sv"
`include "scoreboard.sv"

class environment;
mailbox gen2driv;
mailbox driv2sb;
mailbox mon2sb;

generator gen;
driver drv;
monitor mon;
scoreboard scb;

event driven;

virtual csa_if csaif;

function new(virtual csa_if csaif);
    this.csaif=csaif;
    gen2driv=new();
    driv2sb=new();
    mon2sb=new();

```

<pre> gen=new(gen2drv); drv=new(gen2drv,drv2sb,csaif.DRIVER,driven); mon=new(mon2sb,csaif.MONITOR,driven); scb=new(driv2sb,mon2sb); endfunction task main(input int count); fork gen.main(count); drv.main(count); mon.main(count); scb.main(count); join \$finish; endtask:main endclass:environment </pre>	<p>scoreboard.sv</p> <pre> /// class scoreboard; mailbox driv2sb; mailbox mon2sb; transaction d_trans; transaction m_trans; logic N; logic Z; reg V; //logic [15:0] Flags [7:0]='{default: 8'b0}; real Flags [8]='{default: 64'b0}; real Pass [8]='{default: 64'b0}; real Fail [8]='{default: 64'b0}; real pc [8],f_pc [8], p_pc [8]; logic [2:0] X,Y; event driven; function new(mailbox driv2sb, mon2sb); this.driv2sb=driv2sb; this.mon2sb=mon2sb; endfunction </pre>
--	---

Main task :

```

task main(input int count);
  $display("-----Scoreboard Test Starts-----");
  repeat(count) begin
    m_trans=new();
    mon2sb.get(m_trans);
    report();

    //#10;
    Y={N,d_trans.Cout,V};
    if(m_trans.Sum != d_trans.Sum | m_trans.Cout!=d_trans.Cout)
      begin
        $display("time=%d Failed : A=%d B=%d Cin=%d Expected Sum=%d Resulted Sum=%d Expected Cout=%d N=%d Resulted Cout=%d V=%d Z=%d
", $time,d_trans.A,d_trans.B,d_trans.Cin,d_trans.Sum,m_trans.Sum,d_trans.Cout,N,m_trans.Cout,V,Z);
        if(Y==3'b000)
          Fail[0]=Fail[0]+1;
        else if(Y==3'b001)
          Fail[1]=Fail[1]+1;
        else if(Y==3'b010)
          Fail[2]=Fail[2]+1;
        else if(Y==3'b011)
          Fail[3]=Fail[3]+1;
        else if(Y==3'b100)
          Fail[4]=Fail[4]+1;
      end
  end
endtask

```

```

else if(Y==3'b101)
  Fail[5]=Fail[5]+1;
else if(Y==3'b110)
  Fail[6]=Fail[6]+1;
else if(Y==3'b111)
  Fail[7]=Fail[7]+1;

end

else
begin
$display("time=%d Passed : A=%d B=%d Cin=%d Expected Sum=%d Resulted Sum=%b Expected
Cout=%d N=%d Resulted Cout=%d V=%d Z=%d
",$time,d_trans.A,d_trans.B,d_trans.Cin,d_trans.Sum,m_trans.Sum,d_trans.Cout,N,m_trans.Cout,V,Z);
if(Y==3'b000)
  Pass[0]=Pass[0]+1;
else if(Y==3'b001)
  Pass[1]=Pass[1]+1;
else if(Y==3'b010)
  Pass[2]=Pass[2]+1;
else if(Y==3'b011)
  Pass[3]=Pass[3]+1;
else if(Y==3'b100)
  Pass[4]=Pass[4]+1;
else if(Y==3'b101)
  Pass[5]=Pass[5]+1;
else if(Y==3'b110)
  Pass[6]=Pass[6]+1;
else if(Y==3'b111)
  Pass[7]=Pass[7]+1;
end

end

pc[0]=(Flags[0]*100)/count;
pc[1]=(Flags[1]*100)/count;
pc[2]=(Flags[2]*100)/count;
pc[3]=(Flags[3]*100)/count;
pc[4]=(Flags[4]*100)/count;
pc[5]=(Flags[5]*100)/count;
pc[6]=(Flags[6]*100)/count;
pc[7]=(Flags[7]*100)/count;

p_pc[0]=(Pass[0]*100)/Flags[0];
p_pc[1]=(Pass[1]*100)/Flags[1];
p_pc[2]=(Pass[2]*100)/Flags[2];
p_pc[3]=(Pass[3]*100)/Flags[3];
p_pc[4]=(Pass[4]*100)/Flags[4];
p_pc[5]=(Pass[5]*100)/Flags[5];
p_pc[6]=(Pass[6]*100)/Flags[6];
p_pc[7]=(Pass[7]*100)/Flags[7];

f_pc[0]=(Fail[0]*100)/Flags[0];
f_pc[1]=(Fail[1]*100)/Flags[1];
f_pc[2]=(Fail[2]*100)/Flags[2];
f_pc[3]=(Fail[3]*100)/Flags[3];

```

```

f_pc[4]=(Fail[4]*100)/Flags[4];
f_pc[5]=(Fail[5]*100)/Flags[5];
f_pc[6]=(Fail[6]*100)/Flags[6];
f_pc[7]=(Fail[7]*100)/Flags[7];

$display("\n\n-----Displaying Coverage Results-----\n\n");

$display("# of Case Tested for Type 1: {NCV}='000'=%0.0f cases with percentage=%f.Pass rate=%f,Fail
rate=%f",Flags[0],pc[0],p_pc[0],f_pc[0]);
$display("# of Case Tested for Type 2: {NCV}='001'=%0.0f cases with percentage=%f.Pass rate=%f,Fail
rate=%f",Flags[1],pc[1],p_pc[1],f_pc[1]);
$display("# of Case Tested for Type 3: {NCV}='010'=%0.0f cases with percentage=%f.Pass rate=%f,Fail
rate=%f",Flags[2],pc[2],p_pc[2],f_pc[2]);
$display("# of Case Tested for Type 4: {NCV}='011'=%0.0f cases with percentage=%f.Pass rate=%f,Fail
rate=%f",Flags[3],pc[3],p_pc[3],f_pc[3]);
$display("# of Case Tested for Type 5: {NCV}='100'=%0.0f cases with percentage=%f.Pass rate=%f,Fail
rate=%f",Flags[4],pc[4],p_pc[4],f_pc[4]);
$display("# of Case Tested for Type 6: {NCV}='101'=%0.0f cases with percentage=%f.Pass rate=%f,Fail
rate=%f",Flags[5],pc[5],p_pc[5],f_pc[5]);
$display("# of Case Tested for Type 7: {NCV}='110'=%0.0f cases with percentage=%f.Pass rate=%f,Fail
rate=%f",Flags[6],pc[6],p_pc[6],f_pc[6]);
$display("# of Case Tested for Type 8: {NCV}='111'=%0.0f cases with percentage=%f.Pass rate=%f,Fail
rate=%f\n\n",Flags[7],pc[7],p_pc[7],f_pc[7]);

$display("-----Scoreboard Test Ends-----");
endtask:main

task report();
d_trans=new();
driv2sb.get(d_trans);

{d_trans.Cout, d_trans.Sum}= d_trans.A+d_trans.B+d_trans.Cin;

//{Cout_temp, Sum_temp}= A+B+Cin;
if(d_trans.Sum[15])//Negative flag set
    N=1'b1;
else
    N=1'b0;

if(d_trans.Sum)//Zero flag set
    Z=1'b0;
else
    Z=1'b1;

if((d_trans.A[15]==d_trans.B[15]) & (d_trans.A[15]!=d_trans.Sum[15]))//Overflow flag set
    V=1'b1;
else
    V=1'b0;
X={N,d_trans.Cout,V};

if(X==3'b000)
    Flags[0]=Flags[0]+1;

```

```

else if(X==3'b001)
    Flags[1]=Flags[1]+1;
else if(X==3'b010)
    Flags[2]=Flags[2]+1;
else if(X==3'b011)
    Flags[3]=Flags[3]+1;
else if(X==3'b100)
    Flags[4]=Flags[4]+1;
else if(X==3'b101)
    Flags[5]=Flags[5]+1;
else if(X==3'b110)
    Flags[6]=Flags[6]+1;
else if(X==3'b111)
    Flags[7]=Flags[7]+1;

endtask:report
endclass:scoreboard

```

testbench.sv

```

`include " testcase01.sv"
`include " interface.sv"

```

```

module csa_top_tb;
    bit clk;

initial begin
    forever #5 clk =~clk;
end

int count=1000;
csa_if csaif(clk);

test test01(count,csaif);

initial begin
    $dumpfile("dump.vcd");
    $dumpvars;
end

csa_top DUT (
    .A(csaif.A),
    .B(csaif.B),
    // .S(csaif.S),
    .Cin(csaif.Cin),
    .Sum(csaif.Sum),
    .Cout(csaif.Cout)
);

endmodule

```

2. Output from EDA Playground:

-----Scoreboard Test Starts-----

```
time=      25  Passed : A=43748 B= 6894 Cin=0 Expected Sum=50642 Resulted Sum=50642 Expected
Cout=0 N=1 Resulted Cout=0 V=0 Z=0
time=      35  Passed : A= 5418 B=42976 Cin=0 Expected Sum=48394 Resulted Sum=48394 Expected
Cout=0 N=1 Resulted Cout=0 V=0 Z=0
time=      45  Passed : A=62600 B=18031 Cin=1 Expected Sum=15096 Resulted Sum=15096 Expected
Cout=1 N=0 Resulted Cout=1 V=0 Z=0
time=      55  Passed : A=20985 B=14448 Cin=0 Expected Sum=35433 Resulted Sum=35433 Expected
Cout=0 N=1 Resulted Cout=0 V=1 Z=0
time=      65  Passed : A=35080 B=38843 Cin=0 Expected Sum= 8387 Resulted Sum= 8387 Expected
Cout=1 N=0 Resulted Cout=1 V=1 Z=0
time=      75  Passed : A=19064 B=56466 Cin=1 Expected Sum= 9995 Resulted Sum= 9995 Expected
Cout=1 N=0 Resulted Cout=1 V=0 Z=0
time=      85  Passed : A=60493 B=36641 Cin=1 Expected Sum=31599 Resulted Sum=31599 Expected
Cout=1 N=0 Resulted Cout=1 V=1 Z=0
time=      95  Passed : A= 4847 B= 6722 Cin=1 Expected Sum=11570 Resulted Sum=11570 Expected
Cout=0 N=0 Resulted Cout=0 V=0 Z=0
time=     105  Passed : A=30469 B=60654 Cin=0 Expected Sum=25587 Resulted Sum=25587 Expected
Cout=1 N=0 Resulted Cout=1 V=0 Z=0
time=     115  Passed : A=50399 B=37019 Cin=1 Expected Sum=21883 Resulted Sum=21883 Expected
Cout=1 N=0 Resulted Cout=1 V=1 Z=0
time=     125  Passed : A=15205 B=37822 Cin=0 Expected Sum=53027 Resulted Sum=53027 Expected
Cout=0 N=1 Resulted Cout=0 V=0 Z=0
time=     135  Passed : A=55108 B=30756 Cin=0 Expected Sum=20328 Resulted Sum=20328 Expected
Cout=1 N=0 Resulted Cout=1 V=0 Z=0
time=     145  Passed : A=48367 B=19736 Cin=1 Expected Sum= 2568 Resulted Sum= 2568 Expected
Cout=1 N=0 Resulted Cout=1 V=0 Z=0
time=     155  Passed : A=46083 B=56738 Cin=0 Expected Sum=37285 Resulted Sum=37285 Expected
Cout=1 N=1 Resulted Cout=1 V=0 Z=0
time=     165  Passed : A=46005 B=25223 Cin=0 Expected Sum= 5692 Resulted Sum= 5692 Expected
Cout=1 N=0 Resulted Cout=1 V=0 Z=0
time=     175  Passed : A=41795 B= 3005 Cin=1 Expected Sum=44801 Resulted Sum=44801 Expected
Cout=0 N=1 Resulted Cout=0 V=0 Z=0
time=     185  Passed : A=60900 B=26354 Cin=0 Expected Sum=21718 Resulted Sum=21718 Expected
Cout=1 N=0 Resulted Cout=1 V=0 Z=0
time=     195  Passed : A=37037 B=50235 Cin=1 Expected Sum=21737 Resulted Sum=21737 Expected
Cout=1 N=0 Resulted Cout=1 V=1 Z=0
time=     205  Passed : A= 3651 B=35359 Cin=1 Expected Sum=39011 Resulted Sum=39011 Expected
Cout=0 N=1 Resulted Cout=0 V=0 Z=0
time=     215  Passed : A=51587 B=31788 Cin=1 Expected Sum=17840 Resulted Sum=17840 Expected
Cout=1 N=0 Resulted Cout=1 V=0 Z=0
time=     225  Passed : A=52047 B=60390 Cin=1 Expected Sum=46902 Resulted Sum=46902 Expected
Cout=1 N=1 Resulted Cout=1 V=0 Z=0
time=     235  Passed : A=27387 B=18962 Cin=1 Expected Sum=46350 Resulted Sum=46350 Expected
Cout=0 N=1 Resulted Cout=0 V=1 Z=0
time=     245  Passed : A=11375 B=25992 Cin=0 Expected Sum=37367 Resulted Sum=37367 Expected
Cout=0 N=1 Resulted Cout=0 V=1 Z=0
time=     255  Passed : A=44635 B= 89 Cin=1 Expected Sum=44725 Resulted Sum=44725 Expected
Cout=0 N=1 Resulted Cout=0 V=0 Z=0
time=     265  Passed : A=65331 B=52313 Cin=1 Expected Sum=52109 Resulted Sum=52109 Expected
Cout=1 N=1 Resulted Cout=1 V=0 Z=0
```

time= 275 Passed : A=34623 B=41949 Cin=1 Expected Sum=11037 Resulted Sum=11037 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 285 Passed : A=46999 B=12201 Cin=1 Expected Sum=59201 Resulted Sum=59201 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 295 Passed : A=60720 B=57047 Cin=0 Expected Sum=52231 Resulted Sum=52231 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 305 Passed : A=44819 B=22833 Cin=1 Expected Sum= 2117 Resulted Sum= 2117 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 315 Passed : A=22635 B=27986 Cin=0 Expected Sum=50621 Resulted Sum=50621 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 325 Passed : A=13900 B=33275 Cin=1 Expected Sum=47176 Resulted Sum=47176 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 335 Passed : A= 8199 B= 5634 Cin=1 Expected Sum=13834 Resulted Sum=13834 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 345 Passed : A=23691 B=39338 Cin=1 Expected Sum=63030 Resulted Sum=63030 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 355 Passed : A=58174 B=18322 Cin=0 Expected Sum=10960 Resulted Sum=10960 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 365 Passed : A=19285 B=45534 Cin=1 Expected Sum=64820 Resulted Sum=64820 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 375 Passed : A=23048 B=47676 Cin=0 Expected Sum= 5188 Resulted Sum= 5188 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 385 Passed : A= 4554 B=47511 Cin=0 Expected Sum=52065 Resulted Sum=52065 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 395 Passed : A= 9059 B=14513 Cin=0 Expected Sum=23572 Resulted Sum=23572 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 405 Passed : A=54677 B=25983 Cin=0 Expected Sum=15124 Resulted Sum=15124 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 415 Passed : A=50117 B=11873 Cin=1 Expected Sum=61991 Resulted Sum=61991 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 425 Passed : A=13180 B=32598 Cin=1 Expected Sum=45779 Resulted Sum=45779 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 435 Passed : A=29224 B=12074 Cin=0 Expected Sum=41298 Resulted Sum=41298 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 445 Passed : A= 819 B=25805 Cin=1 Expected Sum=26625 Resulted Sum=26625 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 455 Passed : A=53167 B=15414 Cin=1 Expected Sum= 3046 Resulted Sum= 3046 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 465 Passed : A=55441 B= 4202 Cin=0 Expected Sum=59643 Resulted Sum=59643 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 475 Passed : A=51447 B=61712 Cin=1 Expected Sum=47624 Resulted Sum=47624 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 485 Passed : A= 631 B=61328 Cin=0 Expected Sum=61959 Resulted Sum=61959 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 495 Passed : A=38081 B=30767 Cin=1 Expected Sum= 3313 Resulted Sum= 3313 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 505 Passed : A=41013 B=49057 Cin=1 Expected Sum=24535 Resulted Sum=24535 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 515 Passed : A=12838 B=31661 Cin=0 Expected Sum=44499 Resulted Sum=44499 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 525 Passed : A=31961 B=12754 Cin=0 Expected Sum=44715 Resulted Sum=44715 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 535 Passed : A=18719 B= 3678 Cin=0 Expected Sum=22397 Resulted Sum=22397 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 545 Passed : A=21042 B=59241 Cin=0 Expected Sum=14747 Resulted Sum=14747 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0

time= 555 Passed : A=57421 B=21924 Cin=0 Expected Sum=13809 Resulted Sum=13809 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 565 Passed : A= 2656 B=52829 Cin=0 Expected Sum=55485 Resulted Sum=55485 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 575 Passed : A=44014 B=30668 Cin=0 Expected Sum= 9146 Resulted Sum= 9146 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 585 Passed : A=13274 B=12463 Cin=1 Expected Sum=25738 Resulted Sum=25738 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 595 Passed : A=25679 B=46715 Cin=1 Expected Sum= 6859 Resulted Sum= 6859 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 605 Passed : A=56369 B=61731 Cin=1 Expected Sum=52565 Resulted Sum=52565 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 615 Passed : A=54132 B=24223 Cin=1 Expected Sum=12820 Resulted Sum=12820 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 625 Passed : A=25135 B=52398 Cin=1 Expected Sum=11998 Resulted Sum=11998 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 635 Passed : A=55825 B=16351 Cin=0 Expected Sum= 6640 Resulted Sum= 6640 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 645 Passed : A=15453 B=14581 Cin=1 Expected Sum=30035 Resulted Sum=30035 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 655 Passed : A=24795 B= 4161 Cin=0 Expected Sum=28956 Resulted Sum=28956 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 665 Passed : A=23198 B=12060 Cin=1 Expected Sum=35259 Resulted Sum=35259 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 675 Passed : A=40095 B=44564 Cin=1 Expected Sum=19124 Resulted Sum=19124 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 685 Passed : A= 859 B= 9442 Cin=0 Expected Sum=10301 Resulted Sum=10301 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 695 Passed : A=44825 B=60885 Cin=1 Expected Sum=40175 Resulted Sum=40175 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 705 Passed : A=53730 B=18481 Cin=1 Expected Sum= 6676 Resulted Sum= 6676 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 715 Passed : A=48851 B=41753 Cin=0 Expected Sum=25068 Resulted Sum=25068 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 725 Passed : A=52067 B=64394 Cin=1 Expected Sum=50926 Resulted Sum=50926 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 735 Passed : A=27301 B=62500 Cin=0 Expected Sum=24265 Resulted Sum=24265 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 745 Passed : A=62886 B=19976 Cin=1 Expected Sum=17327 Resulted Sum=17327 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 755 Passed : A=17388 B=54342 Cin=1 Expected Sum= 6195 Resulted Sum= 6195 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 765 Passed : A=24090 B=56203 Cin=1 Expected Sum=14758 Resulted Sum=14758 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 775 Passed : A=14452 B=14412 Cin=1 Expected Sum=28865 Resulted Sum=28865 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 785 Passed : A=29616 B=48220 Cin=1 Expected Sum=12301 Resulted Sum=12301 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 795 Passed : A=25150 B=41457 Cin=1 Expected Sum= 1072 Resulted Sum= 1072 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 805 Passed : A=30510 B=16047 Cin=1 Expected Sum=46558 Resulted Sum=46558 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 815 Passed : A= 1779 B=12368 Cin=0 Expected Sum=14147 Resulted Sum=14147 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 825 Passed : A=29753 B=50849 Cin=0 Expected Sum=15066 Resulted Sum=15066 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0

time= 835 Passed : A=22878 B=36350 Cin=0 Expected Sum=59228 Resulted Sum=59228 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 845 Passed : A=37695 B=13290 Cin=1 Expected Sum=50986 Resulted Sum=50986 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 855 Passed : A=13186 B=18382 Cin=0 Expected Sum=31568 Resulted Sum=31568 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 865 Passed : A=30390 B=11129 Cin=0 Expected Sum=41519 Resulted Sum=41519 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 875 Passed : A=39119 B=43571 Cin=0 Expected Sum=17154 Resulted Sum=17154 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 885 Passed : A= 8216 B= 9117 Cin=1 Expected Sum=17334 Resulted Sum=17334 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 895 Passed : A=24203 B=23224 Cin=0 Expected Sum=47427 Resulted Sum=47427 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 905 Passed : A=54046 B= 2445 Cin=1 Expected Sum=56492 Resulted Sum=56492 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 915 Passed : A=23395 B=37048 Cin=1 Expected Sum=60444 Resulted Sum=60444 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 925 Passed : A=41469 B=36820 Cin=0 Expected Sum=12753 Resulted Sum=12753 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 935 Passed : A=16085 B=16046 Cin=1 Expected Sum=32132 Resulted Sum=32132 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 945 Passed : A=20339 B=26609 Cin=1 Expected Sum=46949 Resulted Sum=46949 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 955 Passed : A=11893 B=28324 Cin=0 Expected Sum=40217 Resulted Sum=40217 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 965 Passed : A=32137 B=44365 Cin=1 Expected Sum=10967 Resulted Sum=10967 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 975 Passed : A=33183 B=63673 Cin=1 Expected Sum=31321 Resulted Sum=31321 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 985 Passed : A=46856 B= 1075 Cin=0 Expected Sum=47931 Resulted Sum=47931 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 995 Passed : A=43268 B=62154 Cin=0 Expected Sum=39886 Resulted Sum=39886 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 1005 Passed : A=64082 B=42824 Cin=1 Expected Sum=41371 Resulted Sum=41371 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 1015 Passed : A=33131 B=22593 Cin=0 Expected Sum=55724 Resulted Sum=55724 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1025 Passed : A=50465 B=44639 Cin=0 Expected Sum=29568 Resulted Sum=29568 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 1035 Passed : A=44226 B=31510 Cin=0 Expected Sum=10200 Resulted Sum=10200 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1045 Passed : A=53794 B=57786 Cin=0 Expected Sum=46044 Resulted Sum=46044 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 1055 Passed : A= 9421 B= 1795 Cin=0 Expected Sum=11216 Resulted Sum=11216 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1065 Passed : A=26642 B=31849 Cin=1 Expected Sum=58492 Resulted Sum=58492 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 1075 Passed : A=32640 B=41811 Cin=1 Expected Sum= 8916 Resulted Sum= 8916 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1085 Passed : A=63683 B= 5603 Cin=0 Expected Sum= 3750 Resulted Sum= 3750 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1095 Passed : A=15129 B=47941 Cin=1 Expected Sum=63071 Resulted Sum=63071 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1105 Passed : A=13542 B=28920 Cin=1 Expected Sum=42463 Resulted Sum=42463 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0

time= 1115 Passed : A=11095 B=37626 Cin=0 Expected Sum=48721 Resulted Sum=48721 Expected
 Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1125 Passed : A=28409 B=46310 Cin=0 Expected Sum= 9183 Resulted Sum= 9183 Expected
 Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1135 Passed : A=60994 B= 912 Cin=1 Expected Sum=61907 Resulted Sum=61907 Expected
 Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1145 Passed : A=62818 B=40719 Cin=0 Expected Sum=38001 Resulted Sum=38001 Expected
 Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 1155 Passed : A= 354 B=42668 Cin=1 Expected Sum=43023 Resulted Sum=43023 Expected
 Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1165 Passed : A=23092 B= 4837 Cin=1 Expected Sum=27930 Resulted Sum=27930 Expected
 Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1175 Passed : A= 8262 B=15882 Cin=0 Expected Sum=24144 Resulted Sum=24144 Expected
 Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1185 Passed : A=46190 B=23918 Cin=0 Expected Sum= 4572 Resulted Sum= 4572 Expected
 Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1195 Passed : A= 4627 B=27316 Cin=0 Expected Sum=31943 Resulted Sum=31943 Expected
 Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1205 Passed : A=43489 B=48206 Cin=1 Expected Sum=26160 Resulted Sum=26160 Expected
 Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 1215 Passed : A=21243 B=31851 Cin=0 Expected Sum=53094 Resulted Sum=53094 Expected
 Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 1225 Passed : A=19532 B=12180 Cin=0 Expected Sum=31712 Resulted Sum=31712 Expected
 Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1235 Passed : A=30753 B=43169 Cin=1 Expected Sum= 8387 Resulted Sum= 8387 Expected
 Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1245 Passed : A= 8692 B=27588 Cin=1 Expected Sum=36281 Resulted Sum=36281 Expected
 Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 1255 Passed : A=36157 B=61801 Cin=1 Expected Sum=32423 Resulted Sum=32423 Expected
 Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 1265 Passed : A= 2896 B=15638 Cin=0 Expected Sum=18534 Resulted Sum=18534 Expected
 Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1275 Passed : A=60367 B=62713 Cin=0 Expected Sum=57544 Resulted Sum=57544 Expected
 Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 1285 Passed : A=24373 B=56334 Cin=0 Expected Sum=15171 Resulted Sum=15171 Expected
 Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1295 Passed : A=15587 B=49064 Cin=0 Expected Sum=64651 Resulted Sum=64651 Expected
 Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1305 Passed : A= 4339 B=37943 Cin=0 Expected Sum=42282 Resulted Sum=42282 Expected
 Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1315 Passed : A=32800 B=51663 Cin=1 Expected Sum=18928 Resulted Sum=18928 Expected
 Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 1325 Passed : A= 6179 B=15327 Cin=0 Expected Sum=21506 Resulted Sum=21506 Expected
 Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1335 Passed : A=22100 B=19332 Cin=1 Expected Sum=41433 Resulted Sum=41433 Expected
 Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 1345 Passed : A=14702 B=43767 Cin=0 Expected Sum=58469 Resulted Sum=58469 Expected
 Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1355 Passed : A=36534 B= 12 Cin=0 Expected Sum=36546 Resulted Sum=36546 Expected
 Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1365 Passed : A=43784 B= 4448 Cin=0 Expected Sum=48232 Resulted Sum=48232 Expected
 Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1375 Passed : A=62338 B=53806 Cin=1 Expected Sum=50609 Resulted Sum=50609 Expected
 Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 1385 Passed : A=50602 B=12618 Cin=0 Expected Sum=63220 Resulted Sum=63220 Expected
 Cout=0 N=1 Resulted Cout=0 V=0 Z=0

time= 1395 Passed : A=22028 B= 1877 Cin=0 Expected Sum=23905 Resulted Sum=23905 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1405 Passed : A= 366 B=39105 Cin=1 Expected Sum=39472 Resulted Sum=39472 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1415 Passed : A=52779 B=15822 Cin=0 Expected Sum= 3065 Resulted Sum= 3065 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1425 Passed : A=48575 B= 3533 Cin=1 Expected Sum=52109 Resulted Sum=52109 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1435 Passed : A=61246 B=51499 Cin=0 Expected Sum=47209 Resulted Sum=47209 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 1445 Passed : A=47147 B=32901 Cin=0 Expected Sum=14512 Resulted Sum=14512 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 1455 Passed : A= 5991 B=61045 Cin=0 Expected Sum= 1500 Resulted Sum= 1500 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1465 Passed : A= 312 B=49161 Cin=1 Expected Sum=49474 Resulted Sum=49474 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1475 Passed : A= 3725 B=64440 Cin=0 Expected Sum= 2629 Resulted Sum= 2629 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1485 Passed : A=36507 B=18792 Cin=0 Expected Sum=55299 Resulted Sum=55299 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1495 Passed : A=14978 B=29747 Cin=0 Expected Sum=44725 Resulted Sum=44725 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 1505 Passed : A=11955 B=29064 Cin=1 Expected Sum=41020 Resulted Sum=41020 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 1515 Passed : A=45326 B=25632 Cin=0 Expected Sum= 5422 Resulted Sum= 5422 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1525 Passed : A=12963 B= 280 Cin=0 Expected Sum=13243 Resulted Sum=13243 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1535 Passed : A=24564 B= 3019 Cin=0 Expected Sum=27583 Resulted Sum=27583 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1545 Passed : A=65208 B=47507 Cin=0 Expected Sum=47179 Resulted Sum=47179 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 1555 Passed : A=62492 B=62465 Cin=1 Expected Sum=59422 Resulted Sum=59422 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 1565 Passed : A=20697 B=47280 Cin=0 Expected Sum= 2441 Resulted Sum= 2441 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1575 Passed : A=53197 B= 1463 Cin=1 Expected Sum=54661 Resulted Sum=54661 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1585 Passed : A=53238 B=43165 Cin=0 Expected Sum=30867 Resulted Sum=30867 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 1595 Passed : A=62050 B=65125 Cin=0 Expected Sum=61639 Resulted Sum=61639 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 1605 Passed : A=35255 B=59122 Cin=0 Expected Sum=28841 Resulted Sum=28841 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 1615 Passed : A=31621 B=43202 Cin=0 Expected Sum= 9287 Resulted Sum= 9287 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1625 Passed : A=51922 B=34103 Cin=0 Expected Sum=20489 Resulted Sum=20489 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 1635 Passed : A=35241 B=54239 Cin=0 Expected Sum=23944 Resulted Sum=23944 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 1645 Passed : A=59103 B=24716 Cin=1 Expected Sum=18284 Resulted Sum=18284 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1655 Passed : A=45897 B=41796 Cin=0 Expected Sum=22157 Resulted Sum=22157 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 1665 Passed : A=53334 B= 4960 Cin=0 Expected Sum=58294 Resulted Sum=58294 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0

time= 1675 Passed : A=21004 B=46583 Cin=0 Expected Sum= 2051 Resulted Sum= 2051 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1685 Passed : A=46276 B=45155 Cin=1 Expected Sum=25896 Resulted Sum=25896 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 1695 Passed : A=12498 B=37783 Cin=0 Expected Sum=50281 Resulted Sum=50281 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1705 Passed : A=54961 B=10261 Cin=0 Expected Sum=65222 Resulted Sum=65222 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1715 Passed : A=12502 B=25493 Cin=1 Expected Sum=37996 Resulted Sum=37996 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 1725 Passed : A= 9812 B=17566 Cin=0 Expected Sum=27378 Resulted Sum=27378 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1735 Passed : A=19572 B=13053 Cin=1 Expected Sum=32626 Resulted Sum=32626 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1745 Passed : A=30234 B=45669 Cin=0 Expected Sum=10367 Resulted Sum=10367 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1755 Passed : A= 1885 B= 1257 Cin=1 Expected Sum= 3143 Resulted Sum= 3143 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1765 Passed : A=58036 B=36852 Cin=1 Expected Sum=29353 Resulted Sum=29353 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 1775 Passed : A=46713 B=55324 Cin=0 Expected Sum=36501 Resulted Sum=36501 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 1785 Passed : A=62290 B=37473 Cin=0 Expected Sum=34227 Resulted Sum=34227 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 1795 Passed : A=51941 B=26675 Cin=0 Expected Sum=13080 Resulted Sum=13080 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1805 Passed : A=60362 B=61070 Cin=0 Expected Sum=55896 Resulted Sum=55896 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 1815 Passed : A=49931 B=56025 Cin=1 Expected Sum=40421 Resulted Sum=40421 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 1825 Passed : A= 3445 B=21060 Cin=1 Expected Sum=24506 Resulted Sum=24506 Expected Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1835 Passed : A=37587 B=58278 Cin=0 Expected Sum=30329 Resulted Sum=30329 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 1845 Passed : A=42529 B=49168 Cin=1 Expected Sum=26162 Resulted Sum=26162 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 1855 Passed : A=23196 B=62561 Cin=1 Expected Sum=20222 Resulted Sum=20222 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1865 Passed : A=15396 B=43507 Cin=0 Expected Sum=58903 Resulted Sum=58903 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1875 Passed : A=30392 B=11498 Cin=1 Expected Sum=41891 Resulted Sum=41891 Expected Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 1885 Passed : A=62063 B=65327 Cin=0 Expected Sum=61854 Resulted Sum=61854 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 1895 Passed : A=38322 B=40665 Cin=1 Expected Sum=13452 Resulted Sum=13452 Expected Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 1905 Passed : A=47625 B=15101 Cin=0 Expected Sum=62726 Resulted Sum=62726 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1915 Passed : A=22572 B=37243 Cin=1 Expected Sum=59816 Resulted Sum=59816 Expected Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 1925 Passed : A=17641 B=63091 Cin=1 Expected Sum=15197 Resulted Sum=15197 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1935 Passed : A=55144 B=25156 Cin=1 Expected Sum=14765 Resulted Sum=14765 Expected Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 1945 Passed : A=47155 B=63186 Cin=1 Expected Sum=44806 Resulted Sum=44806 Expected Cout=1 N=1 Resulted Cout=1 V=0 Z=0

time= 1955 Passed : A=10251 B= 8455 Cin=0 Expected Sum=18706 Resulted Sum=18706 Expected
 Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1965 Passed : A= 265 B= 1315 Cin=1 Expected Sum= 1581 Resulted Sum= 1581 Expected
 Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1975 Passed : A= 6787 B= 3140 Cin=1 Expected Sum= 9928 Resulted Sum= 9928 Expected
 Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 1985 Passed : A=13603 B=19444 Cin=1 Expected Sum=33048 Resulted Sum=33048 Expected
 Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 1995 Passed : A=59600 B=61062 Cin=1 Expected Sum=55127 Resulted Sum=55127 Expected
 Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 2005 Passed : A= 1073 B=22946 Cin=1 Expected Sum=24020 Resulted Sum=24020 Expected
 Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 2015 Passed : A=35820 B=58175 Cin=0 Expected Sum=28459 Resulted Sum=28459 Expected
 Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 2025 Passed : A=28738 B=58778 Cin=0 Expected Sum=21980 Resulted Sum=21980 Expected
 Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 2035 Passed : A=54919 B=48953 Cin=1 Expected Sum=38337 Resulted Sum=38337 Expected
 Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 2045 Passed : A= 3059 B=54309 Cin=1 Expected Sum=57369 Resulted Sum=57369 Expected
 Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 2055 Passed : A=19455 B=24879 Cin=0 Expected Sum=44334 Resulted Sum=44334 Expected
 Cout=0 N=1 Resulted Cout=0 V=1 Z=0
 time= 2065 Passed : A=55410 B=31486 Cin=0 Expected Sum=21360 Resulted Sum=21360 Expected
 Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 2075 Passed : A=46263 B=24559 Cin=0 Expected Sum= 5286 Resulted Sum= 5286 Expected
 Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 2085 Passed : A=33190 B=47428 Cin=1 Expected Sum=15083 Resulted Sum=15083 Expected
 Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 2095 Passed : A=61333 B=62702 Cin=1 Expected Sum=58500 Resulted Sum=58500 Expected
 Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 2105 Passed : A= 491 B=28053 Cin=1 Expected Sum=28545 Resulted Sum=28545 Expected
 Cout=0 N=0 Resulted Cout=0 V=0 Z=0
 time= 2115 Passed : A=47126 B= 908 Cin=1 Expected Sum=48035 Resulted Sum=48035 Expected
 Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 2125 Passed : A=55264 B=52058 Cin=0 Expected Sum=41786 Resulted Sum=41786 Expected
 Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 2135 Passed : A= 7289 B=49455 Cin=1 Expected Sum=56745 Resulted Sum=56745 Expected
 Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 2145 Passed : A=29308 B=49061 Cin=1 Expected Sum=12834 Resulted Sum=12834 Expected
 Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 2155 Passed : A=54666 B=57697 Cin=0 Expected Sum=46827 Resulted Sum=46827 Expected
 Cout=1 N=1 Resulted Cout=1 V=0 Z=0
 time= 2165 Passed : A=14060 B=38215 Cin=1 Expected Sum=52276 Resulted Sum=52276 Expected
 Cout=0 N=1 Resulted Cout=0 V=0 Z=0
 time= 2175 Passed : A=30472 B=46953 Cin=0 Expected Sum=11889 Resulted Sum=11889 Expected
 Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 2185 Passed : A=64646 B=10230 Cin=1 Expected Sum= 9341 Resulted Sum= 9341 Expected
 Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 2195 Passed : A=35345 B=46885 Cin=1 Expected Sum=16695 Resulted Sum=16695 Expected
 Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 2205 Passed : A=50074 B=44563 Cin=1 Expected Sum=29102 Resulted Sum=29102 Expected
 Cout=1 N=0 Resulted Cout=1 V=1 Z=0
 time= 2215 Passed : A=20623 B=65029 Cin=0 Expected Sum=20116 Resulted Sum=20116 Expected
 Cout=1 N=0 Resulted Cout=1 V=0 Z=0
 time= 2225 Passed : A= 3444 B=62469 Cin=0 Expected Sum= 377 Resulted Sum= 377 Expected
 Cout=1 N=0 Resulted Cout=1 V=0 Z=0

```

time=      2235  Passed : A=62387 B=51703 Cin=1 Expected Sum=48555 Resulted Sum=48555 Expected
Cout=1 N=1 Resulted Cout=1 V=0 Z=0
time=      2245  Passed : A=21791 B=40273 Cin=0 Expected Sum=62064 Resulted Sum=62064 Expected
Cout=0 N=1 Resulted Cout=0 V=0 Z=0
time=      2255  Passed : A=43679 B=28206 Cin=0 Expected Sum= 6349 Resulted Sum= 6349 Expected
Cout=1 N=0 Resulted Cout=1 V=0 Z=0
time=      2265  Passed : A= 72 B=47022 Cin=0 Expected Sum=47094 Resulted Sum=47094 Expected
Cout=0 N=1 Resulted Cout=0 V=0 Z=0
time=      2275  Passed : A=17088 B=62900 Cin=1 Expected Sum=14453 Resulted Sum=14453 Expected
Cout=1 N=0 Resulted Cout=1 V=0 Z=0
.
.
.
```

Similarly 1000 case tested (not shown everything)

4.2.3 -----Displaying Coverage Results-----

```

# of Case Tested for Type 1: {NCV}='000'=129 cases with percentage=12.900000.Pass rate=100.000000,Fail
rate=0.000000
# of Case Tested for Type 2: {NCV}='001'=0 cases with percentage=0.000000.Pass rate=-nan,Fail rate=-nan
# of Case Tested for Type 3: {NCV}='010'=252 cases with percentage=25.200000.Pass rate=100.000000,Fail
rate=0.000000
# of Case Tested for Type 4: {NCV}='011'=121 cases with percentage=12.100000.Pass rate=100.000000,Fail
rate=0.000000
# of Case Tested for Type 5: {NCV}='100'=235 cases with percentage=23.500000.Pass rate=100.000000,Fail
rate=0.000000
# of Case Tested for Type 6: {NCV}='101'=131 cases with percentage=13.100000.Pass rate=100.000000,Fail
rate=0.000000
# of Case Tested for Type 7: {NCV}='110'=132 cases with percentage=13.200000.Pass rate=100.000000,Fail
rate=0.000000
# of Case Tested for Type 8: {NCV}='111'=0 cases with percentage=0.000000.Pass rate=-nan,Fail rate=-nan
```

-----Scoreboard Test Ends-----
Simulation complete via \$finish(1) at time 10015 NS + 2
./environment.sv:39 \$finish;

3. Output from Cadence:



Figure 4 : Layerd Testbench Output Waveform

4. Coverage Maximization:

In our design, we have considered A[15:0] and B[15:0] as signed numbers. First of all we have identified total no. of types for inputs A[15:0], B[15:0] and Cin according to the set flags N(Negative Flag), Z(Zero Flag), Cout(Output Carry) and V(Overflow Flag).

So, total types of test cases for four bits {NZCV} are=2⁴=16 cases.

But the zero flag Z=1 only when A[15:0]=16'b0, B[15:0]=16'b0 and Cin=1'b0. Probability for Z=1 is =(1/(2³³))= 1.16E-10 (tends to zero).

So, we can neglect the zero flag condition. So, total types of test cases for three bits {NCV} are=2³=8 cases. We'll analyze each type of case here considering 4 bit A,B and 1 bit Cin for simplicity. We know, for 4 bit signed numbers, from

-2⁽⁴⁻¹⁾= -8 to (2⁽⁴⁻¹⁾)-1= +7 representations are possible only.

5. Case 1: {NCV}='000'

$$\begin{array}{l} A=2 \longrightarrow 0010 (2) \\ B=4 \longrightarrow 0100 (4) \end{array}$$

$$\text{Sum} \longrightarrow 0110 (6) \quad \text{within the range [-8 to 7]}$$

Here, N=0, C=0, V=0. In this type of case, no special operations needed to utilize Sum in the next stage.

6. Case 2: $\{NCV\} = '001'$

$$\begin{array}{lll} A=+ve \longrightarrow & 0xxx & \text{or} \\ B=+ve \longrightarrow & 0xxx & \text{or} \end{array} \quad \begin{array}{lll} A=-ve \longrightarrow & 1xxx \\ B=-ve \longrightarrow & 1xxx \end{array}$$

Let, Sum = -ve \rightarrow 1xxx ($V=1$)

In this case, N becomes 1

So, $\{NCV\} = '001'$ not possible

Let, Sum = +ve \longrightarrow 0xxx ($V=1$)

In this case, C has to be =1, otherwise

actual Sum cannot be corrected.

So, $\{NCV\} = '001'$ not possible

7. Case 3: $\{NCV\} = '010'$

$$A=9 \longrightarrow 1001 (-7)$$

$$B=7 \longrightarrow 0111 (+7)$$

$$\text{Sum} \longrightarrow 1_0000 (0) \text{ within the range } [-8 \text{ to } 7]$$

Here, N=0, C=1, V=0. In this type of case, Carry C should be discarded at output. And $\{NCV\} = '010'$

8. Case 4: $\{NCV\} = '011'$

$$A=9 \longrightarrow 1001 (-7)$$

$$B=13 \longrightarrow 1101 (-3)$$

$$\text{Sum} \longrightarrow 1_0110 (-10) \text{ not within the range } [-8 \text{ to } 7], \text{ So } V=1$$

Here, N=0, C=1, V=1. In this type of case, C=1 should be appended to all the bits appearing before Sum[3] to get the accurate result.

9. Case 5: $\{NCV\} = '100'$

$$A=13 \longrightarrow 1101 (-3)$$

$$B=02 \longrightarrow 0010 (+2)$$

$$\text{Sum} \longrightarrow 1111 (-1) \text{ within the range } [-8 \text{ to } 7], \text{ So } V=0$$

Here, N=1, C=0, V=0. In this type of case, no special operations needed to utilize Sum in the next stage.

10. Case 6: $\{NCV\} = '101'$

$$A=7 \longrightarrow 0111 (+7)$$

$$B=1 \longrightarrow 0001 (+1)$$

$$\text{Sum} \longrightarrow 0_1000 (8) \text{ not within the range } [-8 \text{ to } 7], \text{ So } V=1$$

Here, N=1, C=0, V=1. In this type of case, C=0 should be appended to all the bits appearing before Sum[3] to get the accurate result.

11. Case 7: {NCV}='110'

A=13----> 1101 (-3)

B=14----> 1110 (-2)

Sum----> 1_1011 (-5) within the range [-8 to 7], So V=0

Here, N=1, C=1, V=0. In this type of case, Carry C should be discarded at output. And {NCV}='110'

12. Case 8: {NCV}='111'

A=-ve---->1xxx or, A=+ve-----> 0xxx

B=-ve----> 1xxx or B= +ve-----> 0xxx

Let, Sum=+ve-> 0xxx (V=1)

In this case, N becomes 0

So, {NCV}='111' not possible

Let, Sum= -ve-----> 1xxx (V=1)

In this case, C has to be =0, otherwise
actual Sum cannot be corrected.

So, {NCV}='111' not possible

13. Maximization Steps:

1. Applying constraints in testcase01.sv

```
class testcase01 extends transaction;
  constraint c_s {
    A inside {[0:65535]};//maximizing coverage
    B inside {[0:65535]};//maximizing covergae
    Cin inside {[0:1]};
    //S inside {[0:1]}; //maximizing coverage
  }
endclass:testcase01
```

Here, A[15:0] and B[15:0] are forced to vary between 0 to $2^{16}-1=65535$. So, that for each loop in the testbench, different random values of A and B are created for checking all the eight cases stated above.

2. Count Variation:

Constraints in testcase01.sv allow A and B to be variable among different test cases but they are not randomized among different runs. That's why at every run, number of test cases 'count' will be varied. So, that different values of A[15:0] and B[15:0] are tested and checked.

```
initial begin  
    forever #5 clk =~clk;  
end  
  
int count=100;//# no test cases will be varied manually
```

3. Test Vector approach:

Previously we have tried to implement 4 type of cases using testvector.

C=Carry, V=Overflow

```
Case1: {C,V}=2'b00;// 10 cases  
Case2: {C,V}=2'b01;// 10 cases  
Case3: {C,V}=2'b10;// 10 cases  
Case4: {C,V}=2'b11;// 10 cases
```

Testvector file:

```
0001010100111010 1010101111000000 0  
0001001101111111 0001101000110010 1  
0000111001000011 1000101000001111 1  
1011011101010111 0010111100101001 1  
0111010011100001 0010000100100010 0  
0111111010010111 0101010000001110 0  
1011100000110101 0100011100111010 0  
0100001100000111 0000101000110001 1  
0111100100000011 0010101101000000 1  
0110001000000000 0010100101111111 1  
0010111011100011 1001000111011101 0  
1010010100110011 0101101011011011 0  
0111101010101010 1000010101010101 0  
0001011111101011 1110100010011000 0  
0011000101101101 1100111010010011 0  
0101110011001100 1010001100110100 0  
0001111100111110 1110000011000010 0  
1001100010101101 0110011101010011 0  
1101010101010101 0010101010101010 0  
0111100110100111 1000011001011001 0  
1010111010101010 0101001010101011 0  
0110111100101101 1101000011010110 0
```

```

100110101110110 1001101011110110 0
0101111001010101 1010100110101010 0
1110000110111101 0111000011011110 0
1010101010101010 1100110011001100 0
1000111100110011 1011000011001101 0
1100101101011010 0011010110100101 0
1011101110111011 1011101110111011 0
1111111111111111 0000000000000001 0
1000000000000001 1000000000000001 0
1100000000000000 1100000000000000 0
1011111111111111 1011111111111111 0
0111111111111111 0111111111111111 1
1111100000000000 1111100000000000 0
1000111111111111 1000111111111111 0
1011100000000000 1011100000000000 0
1110000000000000 1110000000000000 0
1100111111111111 1100111111111111 0
1111111111111111 1111111111111111 0

```

But testvector approach only tests few fixed cases. No variability here. That's why this approach was dropped.

4. Adding verification algorithm to check which one of the eight cases are tested each time at scoreboard.sv.

```

real Flags [8]='{default: 64'b0};      // to count the occurrence of each cases
real Pass [8]='{default: 64'b0};       // to count passed occurrence of each test cases
real Fail [8]='{default: 64'b0}        // to count failed occurrence of each test cases
real pc [8],f_pc [8], p_pc [8];      //pc for each testcase percentage occurrence
                                      // f_pc for each testcase passing percentage occurrence
                                      // p_pc for each testcase passing percentage occurrence
logic [2:0] X,Y; // accumulate N, C,V

if(X==3'b000)
    Flags[0]=Flags[0]+1;
else if(X==3'b001)
    Flags[1]=Flags[1]+1;
else if(X==3'b010)
    Flags[2]=Flags[2]+1;
else if(X==3'b011)
    Flags[3]=Flags[3]+1;
else if(X==3'b100)
    Flags[4]=Flags[4]+1;

```

```

else if(X==3'b101)
    Flags[5]=Flags[5]+1;
else if(X==3'b110)
    Flags[6]=Flags[6]+1;
else if(X==3'b111)
    Flags[7]=Flags[7]+1;

endtask:report

Y={N,d_trans.Cout,V};
if(m_trans.Sum != d_trans.Sum | m_trans.Cout!=d_trans.Cout)
begin
$display("time=%d Failed : A=%d B=%d Cin=%d Expected Sum=%d Resulted Sum=%d
Expected      Cout=%d          N=%d      Resulted      Cout=%d      V=%d      Z=%d
",$time,d_trans.A,d_trans.B,d_trans.Cin,d_trans.Sum,m_trans.Sum,d_trans.Cout,N,m_trans.Cout,
V,Z);
if(Y==3'b000)
    Fail[0]=Fail[0]+1;
else if(Y==3'b001)
    Fail[1]=Fail[1]+1;
else if(Y==3'b010)
    Fail[2]=Fail[2]+1;
else if(Y==3'b011)
    Fail[3]=Fail[3]+1;
else if(Y==3'b100)
    Fail[4]=Fail[4]+1;
else if(Y==3'b101)
    Fail[5]=Fail[5]+1;
else if(Y==3'b110)
    Fail[6]=Fail[6]+1;
else if(Y==3'b111)
    Fail[7]=Fail[7]+1;

end

else
begin
$display("time=%d Passed : A=%d B=%d Cin=%d Expected Sum=%d Resulted
Sum=%d      Expected      Cout=%d      N=%d      Resulted      Cout=%d      V=%d      Z=%d
",$time,d_trans.A,d_trans.B,d_trans.Cin,d_trans.Sum,m_trans.Sum,d_trans.Cout,N,m_trans.Cout,
V,Z);
if(Y==3'b000)
    Pass[0]=Pass[0]+1;
else if(Y==3'b001)

```

```

        Pass[1]=Pass[1]+1;
else if(Y==3'b010)
    Pass[2]=Pass[2]+1;
else if(Y==3'b011)
    Pass[3]=Pass[3]+1;
else if(Y==3'b100)
    Pass[4]=Pass[4]+1;
else if(Y==3'b101)
    Pass[5]=Pass[5]+1;
else if(Y==3'b110)
    Pass[6]=Pass[6]+1;
else if(Y==3'b111)
    Pass[7]=Pass[7]+1;
end

end

pc[0]=(Flags[0]*100)/count;
pc[1]=(Flags[1]*100)/count;
pc[2]=(Flags[2]*100)/count;
pc[3]=(Flags[3]*100)/count;
pc[4]=(Flags[4]*100)/count;
pc[5]=(Flags[5]*100)/count;
pc[6]=(Flags[6]*100)/count;
pc[7]=(Flags[7]*100)/count;

p_pc[0]=(Pass[0]*100)/Flags[0];
p_pc[1]=(Pass[1]*100)/Flags[1];
p_pc[2]=(Pass[2]*100)/Flags[2];
p_pc[3]=(Pass[3]*100)/Flags[3];
p_pc[4]=(Pass[4]*100)/Flags[4];
p_pc[5]=(Pass[5]*100)/Flags[5];
p_pc[6]=(Pass[6]*100)/Flags[6];
p_pc[7]=(Pass[7]*100)/Flags[7];

f_pc[0]=(Fail[0]*100)/Flags[0];
f_pc[1]=(Fail[1]*100)/Flags[1];
f_pc[2]=(Fail[2]*100)/Flags[2];
f_pc[3]=(Fail[3]*100)/Flags[3];
f_pc[4]=(Fail[4]*100)/Flags[4];
f_pc[5]=(Fail[5]*100)/Flags[5];
f_pc[6]=(Fail[6]*100)/Flags[6];
f_pc[7]=(Fail[7]*100)/Flags[7];

endtask

```

4.3 Synthesis

4.3.1 TCL Code

```
#run Genus in Legacy UI if Genus is invoked with Common UI ::legacy::set_attribute common_ui false /  
;  
if {[file exists /proc/cpuinfo]} {sh grep "model name" /proc/cpuinfo sh grep "cpu MHz" /proc/cpuinfo}  
} puts "Hostname : [info hostname]"  
#####  
Preset global variables and attributes  
#####  
set DESIGN csa_top  
set SYN_EFF medium  
set MAP_EFF medium  
set OPT_EFF medium  
  
# Directory of PDK  
#set pdk_dir /home/wsadiq/buet_flow/GPDK045  
set pdk_dir /home/cad/VLSI2Lab/Digital/library/  
#set_attribute init_lib_search_path $pdk_dir/gsclib045/timing  
#set_attribute init_hdl_search_path /home/wsadiq/buet_flow/rtl  
set_attribute init_lib_search_path $pdk_dir  
  
#set_attribute init_hdl_search_path ../../rtl  
##Set synthesizing effort for each synthesis stage  
set_attribute syn_generic_effort $SYN_EFF  
set_attribute syn_map_effort $MAP_EFF  
set_attribute syn_opt_effort $OPT_EFF  
  
#set_attribute library "\\  
slow_vdd1v0_basicCells_hvt.lib \  
slow_vdd1v0_basicCells.lib \  
slow_vdd1v0_basicCells_lvt.lib"  
  
set_attribute library "\\  
slow_vdd1v0_basicCells.lib"  
set_dont_use [get_lib_cells CLK*]  
set_dont_use [get_lib_cells SDFF*]  
set_dont_use [get_lib_cells DLY*]  
set_dont_use [get_lib_cells HOLD*]  
  
# If you dont want to use LVT uncomment this line  
#set_dont_use [get_lib_cells *LVT*]  
#####  
Load Design  
#####  
source verilog_files.tcl read_hdl "\${DESIGN}.v"  
elaborate $DESIGN  
puts "Runtime & Memory after 'read_hdl'"  
time_info Elaboration
```

```

check_design -unresolved
#####
# Constraints Setup
#####
read_sdc csa_top.sdc
report timing -encounter >> reports/${DESIGN}_pretim.rpt
#####
# Synthesizing to generic
#####
syn_generic
puts "Runtime & Memory after 'syn_generic'"
time_info GENERIC
report datapath > reports/${DESIGN}_datapath_generic.rpt
generate_reports -outdir reports -tag generic
write_db -to_file ${DESIGN}_generic.db
report timing -encounter >> reports/${DESIGN}_generic.rptF

##This synthesizes your code
synthesize -to_mapped
## This writes all your files
write -mapped > csa_top_synth.vF
## THESE FILES ARE NOT REQUIRED, THE SDC FILE IS A TIMING FILE
write_script > scrip

```

4.3.2 SDC Code

```

# setting up time units
set_units -time 1ns -capacitance pF

# setting the clock period 10ns, as period = 1/freq, here, freq = 100MHz
# set clock_period 10;
set top_module "csa_top"

#set clock_port {clk};
#set reset_port {rst_n};

# setting the input ports in a list to a variable
set input_ports {A, B, Cin} ;

# setting the output ports in a list to a variable
set output_ports {Sum, Cout} ;

# define the clocks
#create_clock -period ${clock_period} -waveform {0 6} -name func_clk
#[get_ports ${clock_port}]

# setting up constraints for the reset signal
#set_multicycle_path -setup 3 -from [get_ports ${reset_port}]
#set_multicycle_path -hold 2 -from [get_ports ${reset_port}]

```

```

# Define input delays
#set_input_delay 0.4 -clock [get_clocks {func_clk}] ${input_ports}
# Define output delays
#set_output_delay 0.6 -clock [get_clocks {func_clk}] ${output_ports}

```

4.3.3 Module

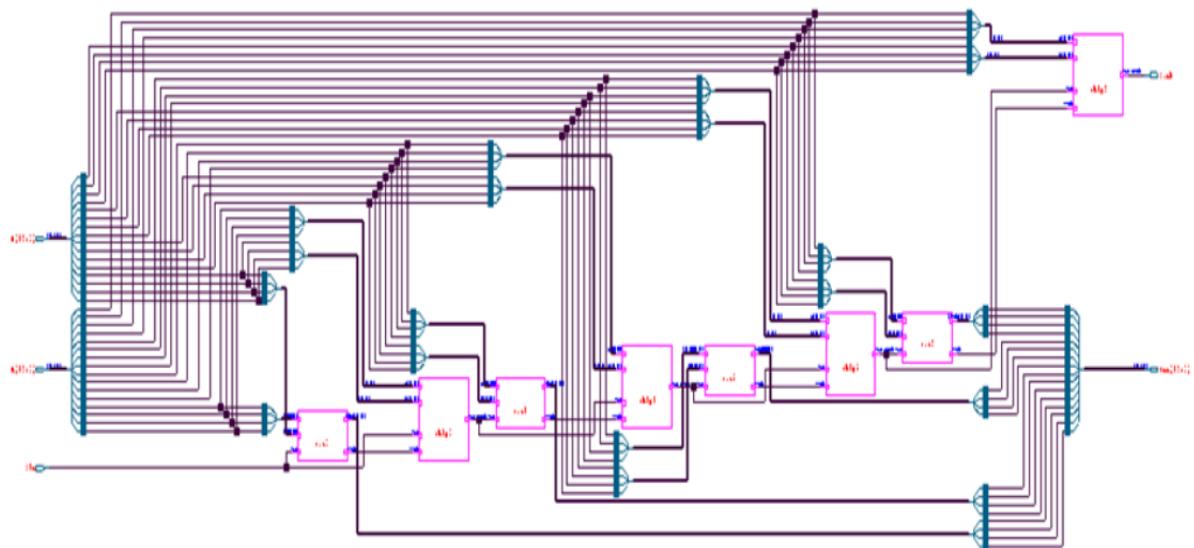


Figure 5 : Synthesis Module for Carry Skip Adder

4.3.4 Submodule

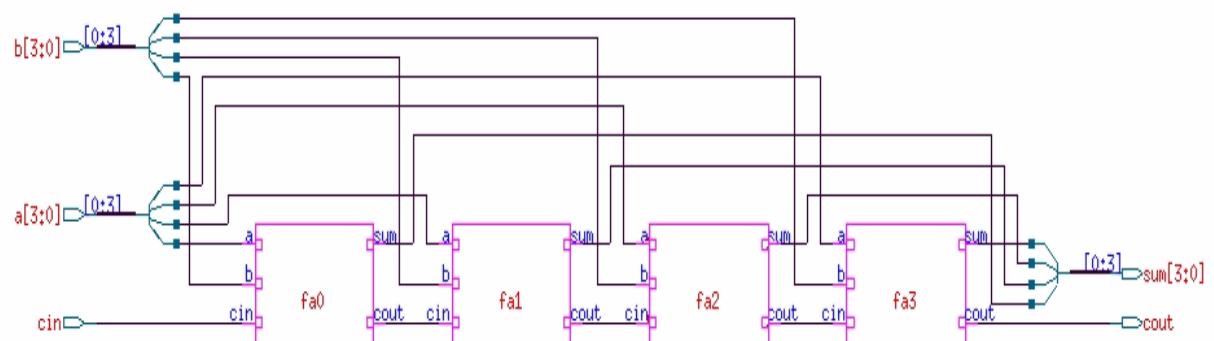


Figure 6 : RCA0 Submodule

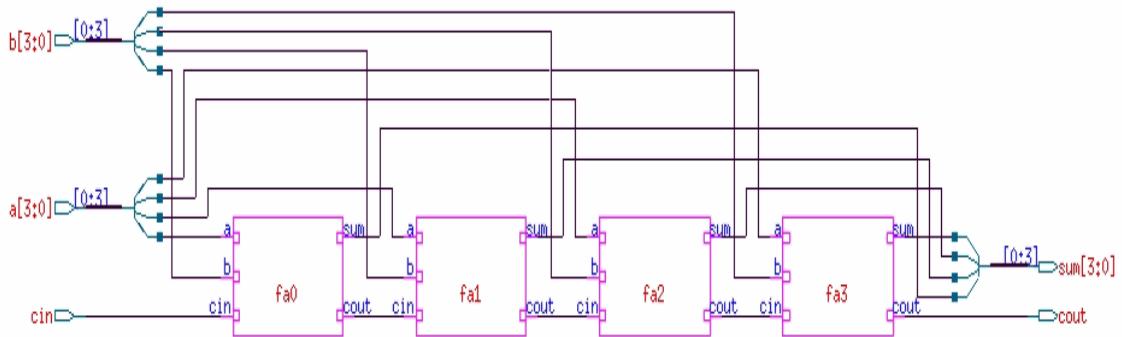


Figure 7 : RCA1 Submodule

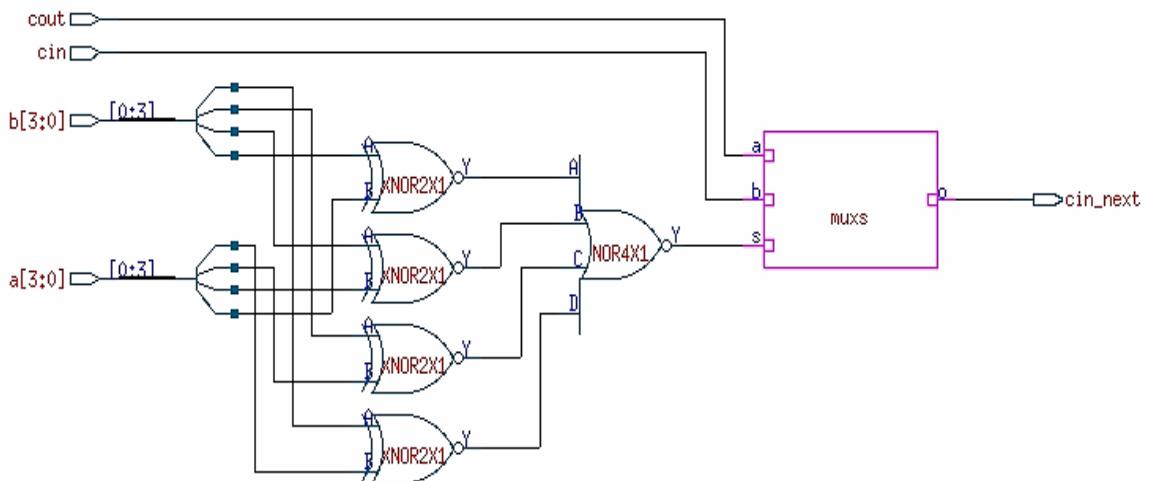


Figure 8 : Skip0 Submodule

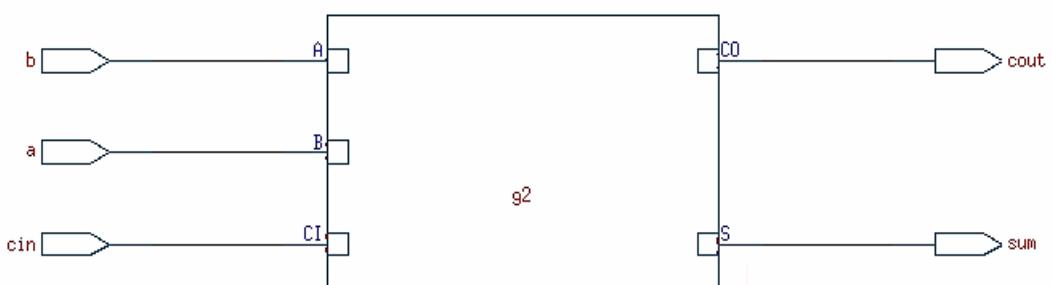


Figure 9 : Full Adder Submodule

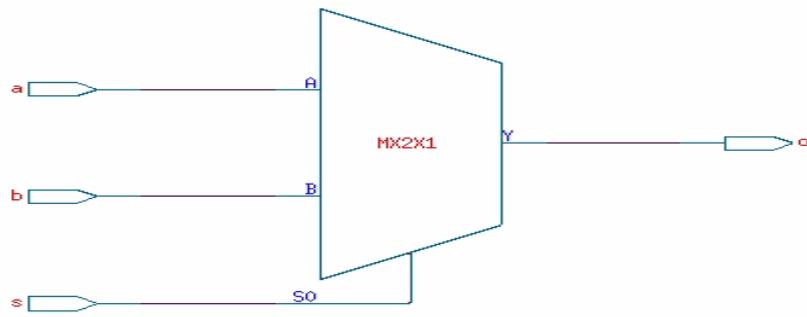


Figure 10 : Mux21 Submodule

4.3.5 Reports

Data for different types of power consumption:

Type	Leakage Power (nW)	Internal Power (nW)	Net Power (nW)	Switching Power (nW)	Total Area
Low	2.93	2451.19	574.17	3025.36	136.80
Medium	2.93	2451.19	574.17	3025.36	136.80
High	2.93	2451.19	574.17	3025.36	136.80

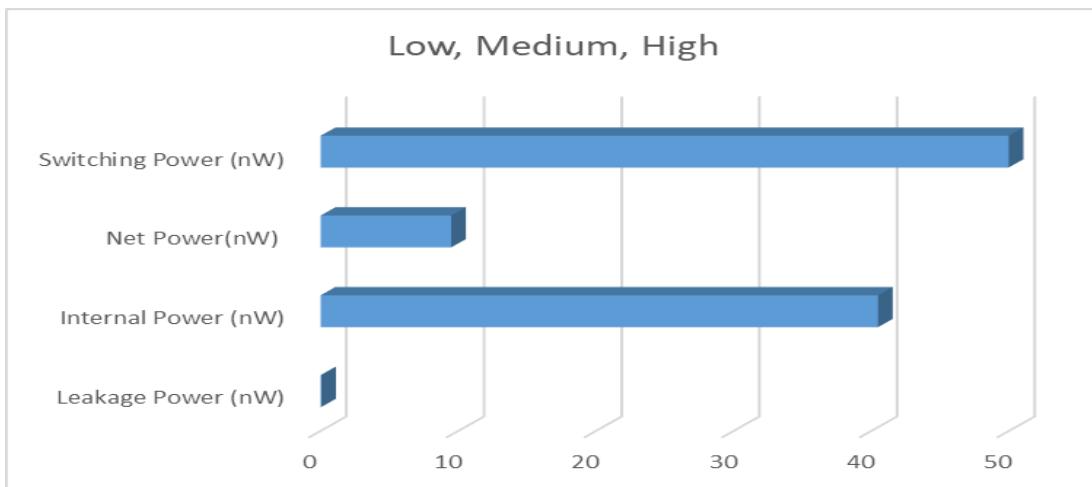


Figure 11: Percentage power consumption for different area

The graph represents the power consumption for different set of areas including switching, net, internal, and leakage. It is obvious that the leakage has the lowest power consumption and the net costs the second lowest power around 8% of the total power. On the other hand, the internal and switching power loss has owned the second maximum and the maximum power consumption respectively. Maximum of the power are being utilized by the net and switching mechanism around 40% and 48% of the total power.

Data of power consumption for clock and without clock:

Set up Time	Hold Time	Leakage Power (nW)	Internal Power (nW)	Net Power (nW)	Switching Power (nW)	Total Area
N/A	N/A	2.93	2451.19	574.17	3025.36	136.80
3	2	2.93	2524.11	589.78	3113.89	136.80
8	6	2.93	2524.11	589.78	3113.89	136.80

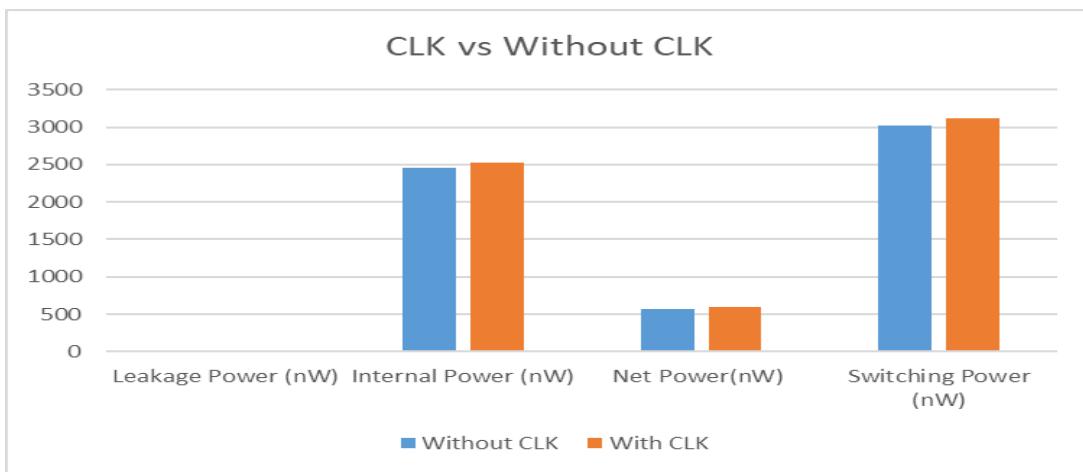


Figure 12 : Comparison of power consumption with and without clock

The above mentioned table and bar graph represent a clear indication about the power consumption on different types of sectors with the presence and the without the clock signals. There is no much difference between these two types of dataset though with presence of the clock signals , the power consumption is greater on each case. With the presence of the clock signal, a change in the hold time and the set-up time do not effect on the power consumption of the components . So , a conclusion may be taken that the hold time and set-up time do not effect the power consumption but adding a clock signal may add a little amount of extra power consumption.

Power Report Summary :

SYN_EFF medium
MAP_EFF medium
OPT_EFF medium

X Report Power

Generated by: Genus(TM) Synthesis Solution 16.13-e036_1 (Dec 20 2016)
Generated on: Dec 05 2024 12:12:55
Module: csa_top
Technology library: slow_vdd1v0 1.0
Operating conditions: PVT_0P9V_125C (balanced_tree)
Wireload mode: enclosed

Instance	Cells	Leakage (nW)	Internal (nW)	Net (nW)	Switching (nW)
csa_top	40	2.93	2451.19	574.17	3025.36
csa_top/rca0	4	0.40	376.85	24.47	401.32
csa_top/rca0/fa0	1	0.10	83.57	6.33	89.90
csa_top/rca0/fa1	1	0.10	92.95	7.59	100.54
csa_top/rca0/fa2	1	0.10	97.92	7.59	105.51
csa_top/rca0/fa3	1	0.10	102.41	2.95	105.37
csa_top/rca1	4	0.40	414.48	28.89	443.15
csa_top/rca1/fa0	1	0.10	102.91	7.59	110.50
csa_top/rca1/fa1	1	0.10	92.85	7.59	100.44
csa_top/rca1/fa2	1	0.10	106.91	10.12	117.04
csa_top/rca1/fa3	1	0.10	111.79	3.38	115.17
csa_top/rca2	4	0.40	476.25	36.28	512.53
csa_top/rca2/fa0	1	0.10	125.85	12.66	138.51
csa_top/rca2/fa1	1	0.10	126.24	11.39	137.63
csa_top/rca2/fa2	1	0.10	117.34	8.86	126.20
csa_top/rca2/fa3	1	0.10	106.82	3.38	110.19
csa_top/rca3	4	0.40	386.07	24.05	410.11

X Report Power

Generated by: Genus(TM) Synthesis Solution 16.13-e036_1 (Dec 20 2016)
Generated on: Dec 05 2024 12:12:55
Module: csa_top
Technology library: slow_vdd1v0 1.0
Operating conditions: PVT_0P9V_125C (balanced_tree)
Wireload mode: enclosed

Instance	Cells	Leakage (nW)	Internal (nW)	Net (nW)	Switching (nW)
csa_top/rca2/fa1	1	0.10	126.24	11.39	137.63
csa_top/rca2/fa2	1	0.10	117.34	8.86	126.20
csa_top/rca2/fa3	1	0.10	106.82	3.38	110.19
csa_top/rca3	4	0.40	386.07	24.05	410.11
csa_top/rca3/fa0	1	0.10	102.83	7.59	110.42
csa_top/rca3/fa1	1	0.10	93.42	6.33	99.75
csa_top/rca3/fa2	1	0.10	88.44	6.33	94.77
csa_top/rca3/fa3	1	0.10	101.38	3.80	105.18
csa_top/skip0	6	0.33	202.27	42.19	244.46
csa_top/skip0/muxs	1	0.05	34.48	11.81	46.29
csa_top/skip1	6	0.33	205.35	43.87	249.22
csa_top/skip1/muxs	1	0.05	39.86	13.50	53.36
csa_top/skip2	6	0.33	191.99	41.34	233.33
csa_top/skip2/muxs	1	0.05	36.64	13.50	50.14
csa_top/skip3	6	0.32	197.95	29.53	227.48
csa_top/skip3/muxs	1	0.04	34.62	0.00	34.62

Figure 13 : Detail power consumption table of the components

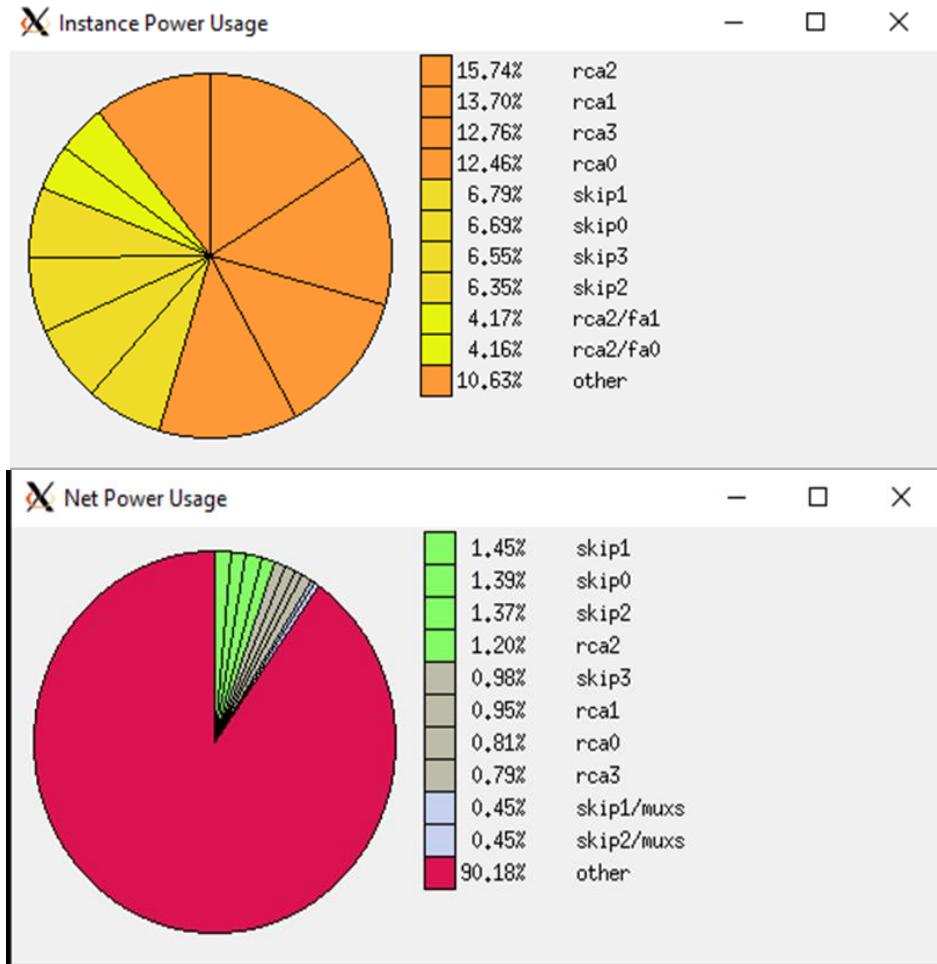


Figure 14 : A pie-chart of detail power consumption table of the components

The table and the pie-chart are the measurement of the power consumption of the different elements and the nets of the module and sub modules. Among all the submodules, the RCA submodules use the most of the power to run and the RCA2 is on the top of these . On the other hand, around 8% power is being consumed by the nets and 90% power is being consumed by the other components.

Area Report Summary:

X Report Area

Generated by: Genus(TM) Synthesis Solution 16.13-e036_1 (Dec 20 2016)
 Generated on: Dec 05 2024 12:08:25
 Module: csa_top
 Technology library: slow_vdd1v0 1.0
 Operating conditions: PVT_0P9V_125C (balanced_tree)
 Wireload mode: enclosed

Instance	Cells	Cell Area	Net Area	Total Area	Wireload	WL Flag
csa_top	40	136.80	0.00	136.80	<none>	(D)
csa_top/rca0	4	20.52	0.00	20.52	<none>	(D)
csa_top/rca0/fa0	1	5.13	0.00	5.13	<none>	(D)
csa_top/rca0/fa1	1	5.13	0.00	5.13	<none>	(D)
csa_top/rca0/fa2	1	5.13	0.00	5.13	<none>	(D)
csa_top/rca0/fa3	1	5.13	0.00	5.13	<none>	(D)
csa_top/rca1	4	20.52	0.00	20.52	<none>	(D)
csa_top/rca1/fa0	1	5.13	0.00	5.13	<none>	(D)
csa_top/rca1/fa1	1	5.13	0.00	5.13	<none>	(D)
csa_top/rca1/fa2	1	5.13	0.00	5.13	<none>	(D)
csa_top/rca1/fa3	1	5.13	0.00	5.13	<none>	(D)
csa_top/rca2	4	20.52	0.00	20.52	<none>	(D)
csa_top/rca2/fa0	1	5.13	0.00	5.13	<none>	(D)
csa_top/rca2/fa1	1	5.13	0.00	5.13	<none>	(D)

X Report Area

Generated by: Genus(TM) Synthesis Solution 16.13-e036_1 (Dec 20 2016)
 Generated on: Dec 05 2024 12:08:25
 Module: csa_top
 Technology library: slow_vdd1v0 1.0
 Operating conditions: PVT_0P9V_125C (balanced_tree)
 Wireload mode: enclosed

Instance	Cells	Cell Area	Net Area	Total Area	Wireload	WL Flag
csa_top/rca2/fa1	1	5.13	0.00	5.13	<none>	(D)
csa_top/rca2/fa2	1	5.13	0.00	5.13	<none>	(D)
csa_top/rca2/fa3	1	5.13	0.00	5.13	<none>	(D)
csa_top/rca3	4	20.52	0.00	20.52	<none>	(D)
csa_top/rca3/fa0	1	5.13	0.00	5.13	<none>	(D)
csa_top/rca3/fa1	1	5.13	0.00	5.13	<none>	(D)
csa_top/rca3/fa2	1	5.13	0.00	5.13	<none>	(D)
csa_top/rca3/fa3	1	5.13	0.00	5.13	<none>	(D)
csa_top/skip0	6	13.68	0.00	13.68	<none>	(D)
csa_top/skip0/muxs	1	2.39	0.00	2.39	<none>	(D)
csa_top/skip1	6	13.68	0.00	13.68	<none>	(D)
csa_top/skip1/muxs	1	2.39	0.00	2.39	<none>	(D)
csa_top/skip2	6	13.68	0.00	13.68	<none>	(D)
csa_top/skip2/muxs	1	2.39	0.00	2.39	<none>	(D)
csa_top/skip3	6	13.68	0.00	13.68	<none>	(D)
csa_top/skip3/muxs	1	2.39	0.00	2.39	<none>	(D)

Figure 15 : Detail area consumption table of the components

X Report Mapped Gates

Generated by: Genus(TM) Synthesis Solution 16.13-s036_1 (Dec 20 2016)
Generated on: Dec 05 2024 12:16:02
Module: csa_top
Technology library: slow_vdd1v0 1.0
Operating conditions: PVT_0P9V_125C (balanced_tree)
Wireload mode: enclosed

Gate	Instances	Area	Library
ADDFX1	16	82.08	slow_vdd1v0
MX2X1	3	7.18	slow_vdd1v0
MX2XL	1	2.39	slow_vdd1v0
NOR4X1	4	6.84	slow_vdd1v0
XNOR2X1	16	38.30	slow_vdd1v0
TOTAL	40	136.79	

Close **Help**

Figure 16 : Detail power consumption table of the instances

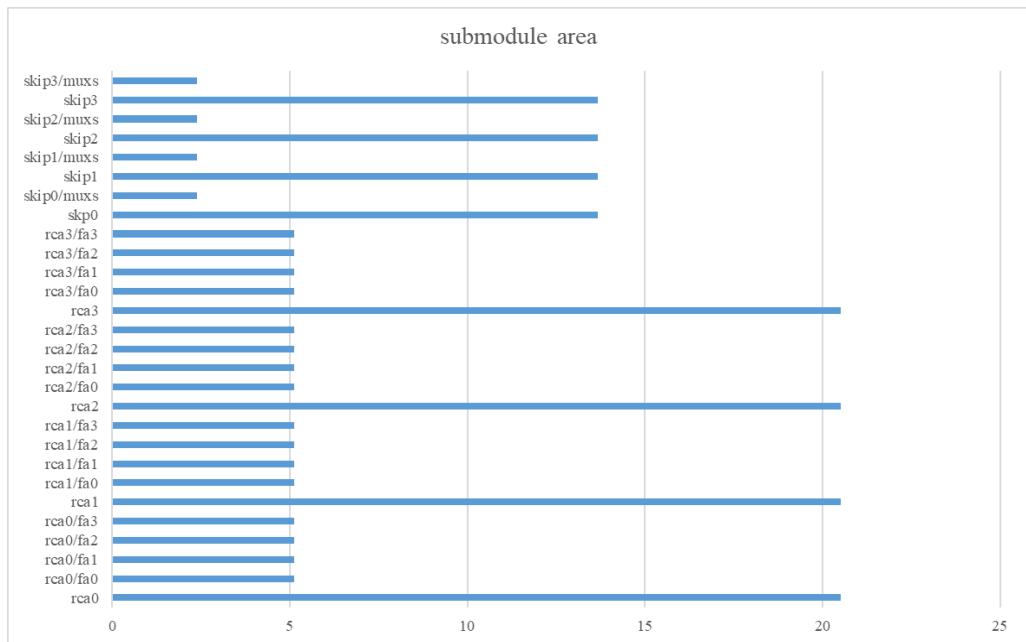


Figure 17 : Bar chart of the different components and modules

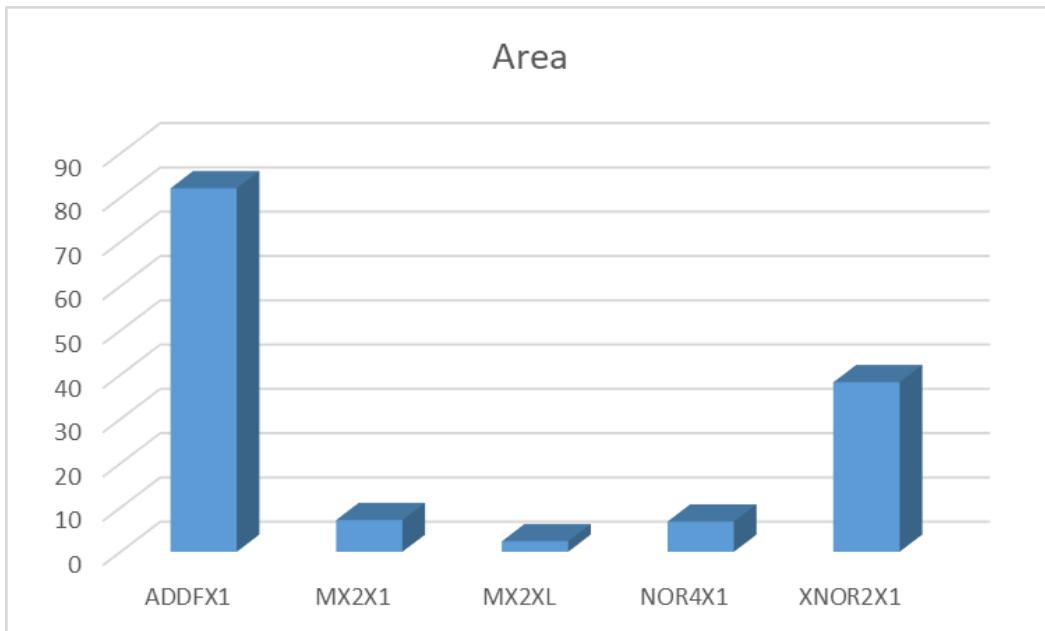


Figure 18 : Area of different submodules of adder and mux

The table and the chart give a pictorial representation of how the submodules are taking area in the module. From the second graph, The adder takes the most of the area among the submodules.

From the first graph, it indicates that the most of the area occupied by the adder are equally distributed among the four RCA submodules area of 20 unit . The skip logic submodule is the second one having the area of about 13 unit each . Full adder sub module is at the third places.

4.3.6 Comparison among Low , Medium , and High SYN_EFF ,MAP_EFF & OPT_EFF



Figure 19 : Comparision of the Syn_EFF, MAP_EFF, OPT_EFF for Low, Medium & High respectively from up to down

The pie-charts shows that there is no impact of Low, Medium and High setup on SYN_EFF , MAP_EFF ,& OPT_EFF . For all cases, the power consumption by the components and the nets connected to the submodules are same by percentage.

4.3.7 Timing Report

Airlines	Endpoint	Slack (ps)	Rise Slew (ps)	Fall Slew (ps)		
opath_1_1	Cout	uncon	14.70	16.00		
Pin	Type	Fanout	Load (#F)	Slew (ps)	Delay (ps)	Arrival (ps)
ca2/cout						
skip2/cout						
muxs/a						
g24/A					0.00	2548.40
g24/Y						
muxs/o						
skip2/cin_next						
ca3/cin						
fa0/cin						
g2/C1					0.00	2717.50
g2/C0	ADDFX1	1	0.60	38.70	184.40	2901.90
fa0/cout						
fa1/cin						
g2/C1					0.00	2901.90
g2/C0	ADDFX1	1	0.60	38.70	186.00	3087.90
fa1/cout						
fa2/cin						
g2/C1					0.00	3087.90
g2/C0	ADDFX1	1	0.60	38.70	186.00	3273.90
fa3/cin						
g2/C1					0.00	3273.90
g2/C0	ADDFX1	1	0.20	30.50	181.40	3455.30
fa3/cout						
ca3/cout						
skip3/cout						
muxs/a						
g24/A					0.00	3455.30
g24/Y						
muxs/o						
skip3/cin_next						
Cout	out port				0.00	3595.70

Figure 20 : Timing Report for High Effort

Airlines	Endpoint	Slack (ps)	Rise Slew (ps)	Fall Slew (ps)		
opath_1_1	Cout	uncon	14.70	16.00		
Pin	Type	Fanout	Load (#F)	Slew (ps)	Delay (ps)	Arrival (ps)
ca2/cout						
skip2/cout						
muxs/a						
g24/A					0.00	2548.40
g24/Y						
muxs/o						
skip2/cin_next						
ca3/cin						
fa0/cin						
g2/C1					0.00	2717.50
g2/C0	ADDFX1	1	0.60	38.70	184.40	2901.90
fa0/cout						
fa1/cin						
g2/C1					0.00	2901.90
g2/C0	ADDFX1	1	0.60	38.70	186.00	3087.90
fa1/cout						
fa2/cin						
g2/C1					0.00	3087.90
g2/C0	ADDFX1	1	0.60	38.70	186.00	3273.90
fa2/cout						
fa3/cin						
g2/C1					0.00	3273.90
g2/C0	ADDFX1	1	0.20	30.50	181.40	3455.30
fa3/cout						
ca3/cout						
skip3/cout						
muxs/a						
g24/A					0.00	3455.30
g24/Y						
muxs/o						
skip3/cin_next						
Cout	out port				0.00	3595.70

Figure 21 : Timing Report for Medium Effort

Timing Report - (id: 1)

Options Endpoint: Cout

Close

Airlines	Endpoint	Slack (ps)	Rise Slew (ps)	Fall Slew (ps)
cpath_1_1	Cout	uncon	14.70	16.00

Pin	Type	Fanout	Load (#F)	Slew (ps)	Delay (ps)	Arrival (ps)
rca2/cout						
skip2/cout	cma_top/rca2/cout					
muxs/a						
g24/A				0.00	2548.40	
g24/Y	MX2X1	2	0.80	35.60	169.10	2717.50
muxs/o						
skip2/cin_next						
rca3/cin						
fa0/cin						
g2/C1				0.00	2717.50	
g2/CO	ADDFX1	1	0.60	38.70	184.40	2901.90
fa0/cout						
fa1/cin						
g2/C1				0.00	2901.90	
g2/CO	ADDFX1	1	0.60	38.70	186.00	3087.90
fa1/cout						
fa2/cin						
g2/C1				0.00	3087.90	
g2/CO	ADDFX1	1	0.60	38.70	188.00	3273.90
fa2/cout						
fa3/cin						
g2/C1				0.00	3273.90	
g2/CO	ADDFX1	1	0.20	30.50	181.40	3455.30
fa3/cout						
rca3/cout						
skip3/cout						
muxs/a						
g24/A				0.00	3455.30	
g24/Y	MX2XL	1	0.00	14.70	140.40	3595.70
muxs/o						
skip3/cin_next						
Cout	out port			0.00	3595.70	

Figure 22 : Timing Report for LOW Effort

From the timing report it is clear that there is no effect of Low , Medium and High effort on delay time of the output of the instances and the module.

4.4 Physical Design

The physical design has been done using virtuoso , nautilus & , innovous interfaces.

4.4.1 Design Import

First, we imported the synthesized netlist from the result of RTL synthesis with Genus in previous part for Carry Skip Adder. The necessary technology LEF file, standard cell LEF file, and standard cell liberty files were also imported.

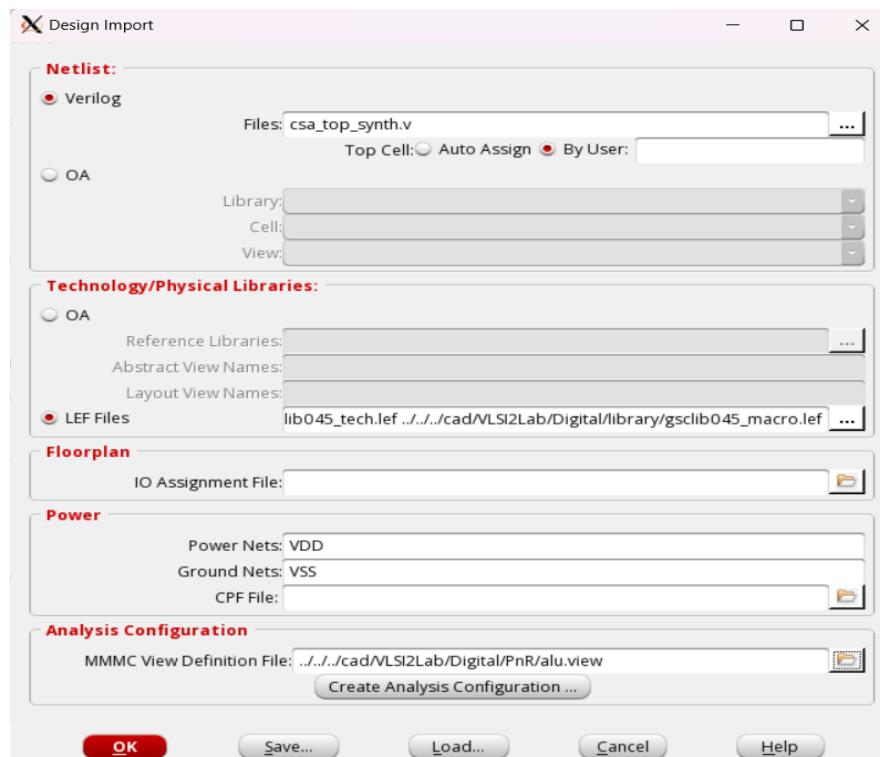


Figure 23 : Design Import Window

4.4.2 Floor planning

Floor planning is the area budgeting of the chip. Here, we specified the floorplan by explicitly providing the width and height of the die. For the CSA, the suggested floorplan area was 14.2×13.68 square microns, but we chose 14.4×15.96 square microns for our final design. Here we use pin spacing of 0.19 micron.

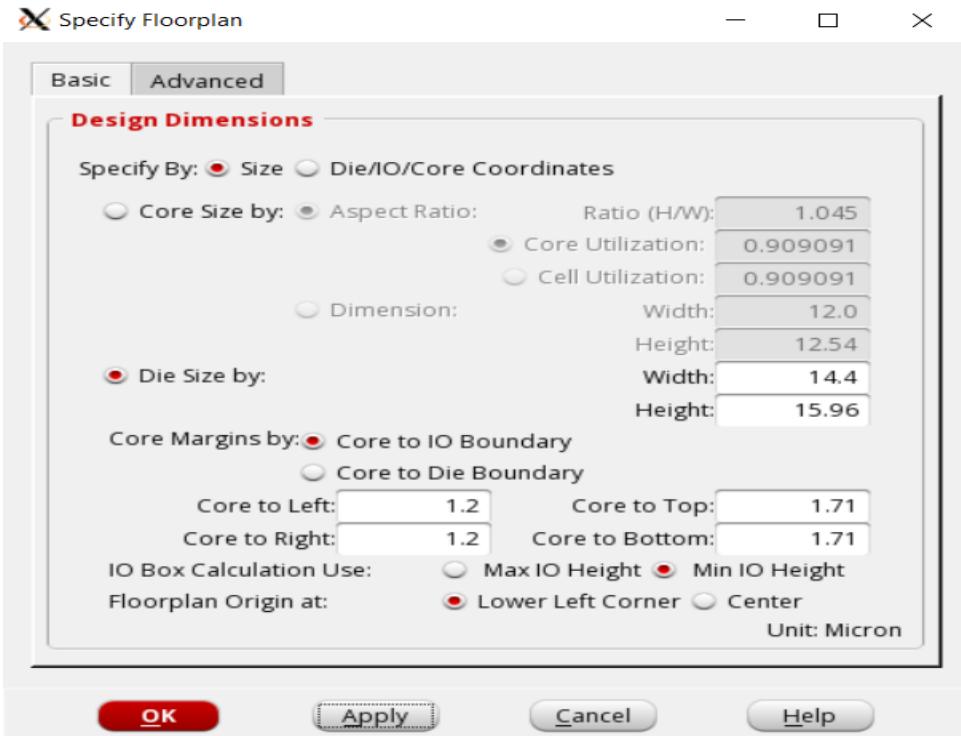


Figure 24 : Floorplan Specification

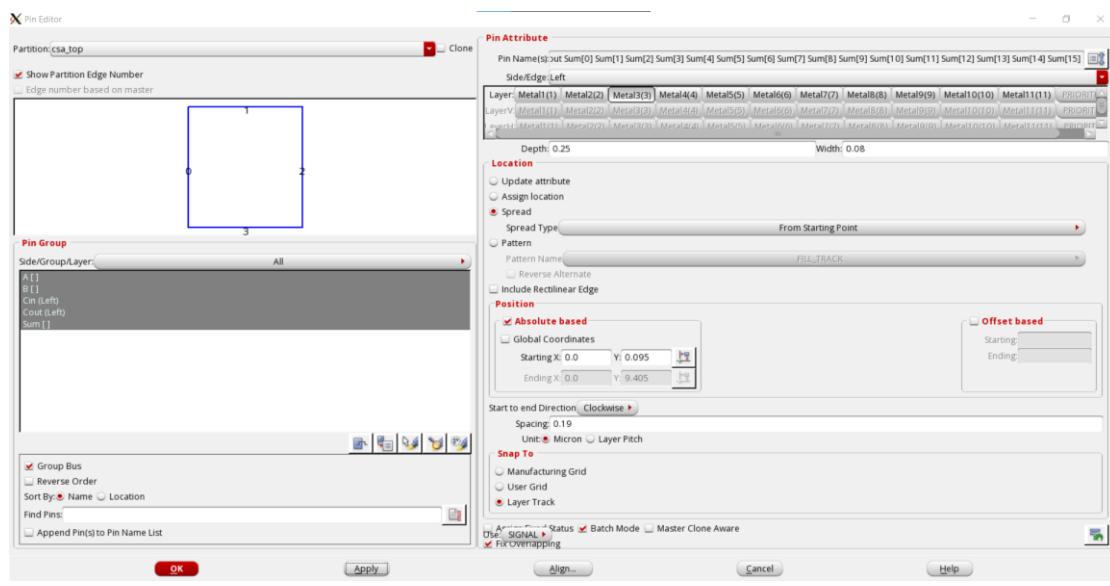


Figure 25 : Creating Pin in "Pin Editor"

4.4.3 Creating Power Mesh

Here we use strip width 0.16 micron and spacing 0.2 micron.



Figure 26 : Add stripes input

After creating power stripes for VDD and VSS, layout is shown below:

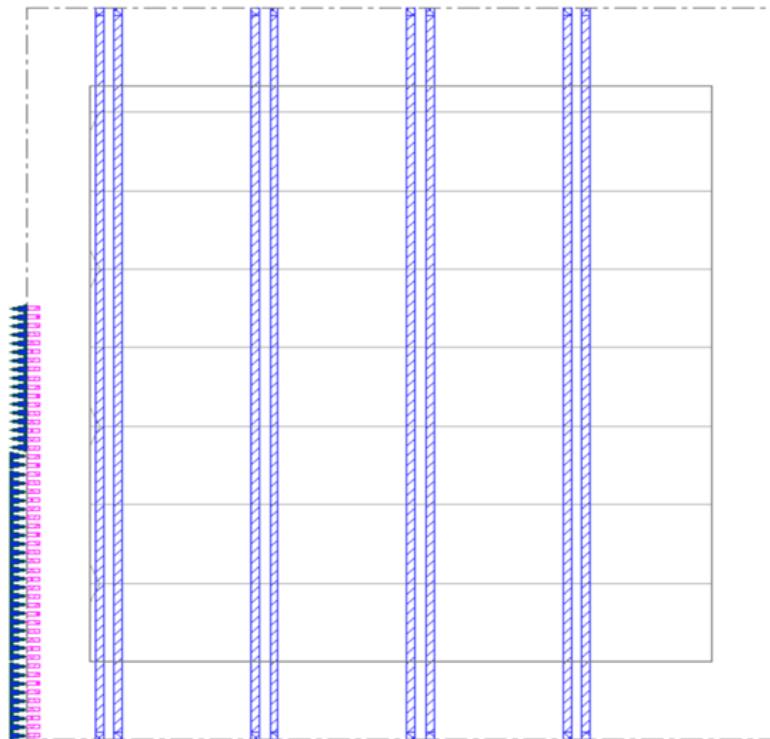


Figure 27 : Adding stripes for power planning

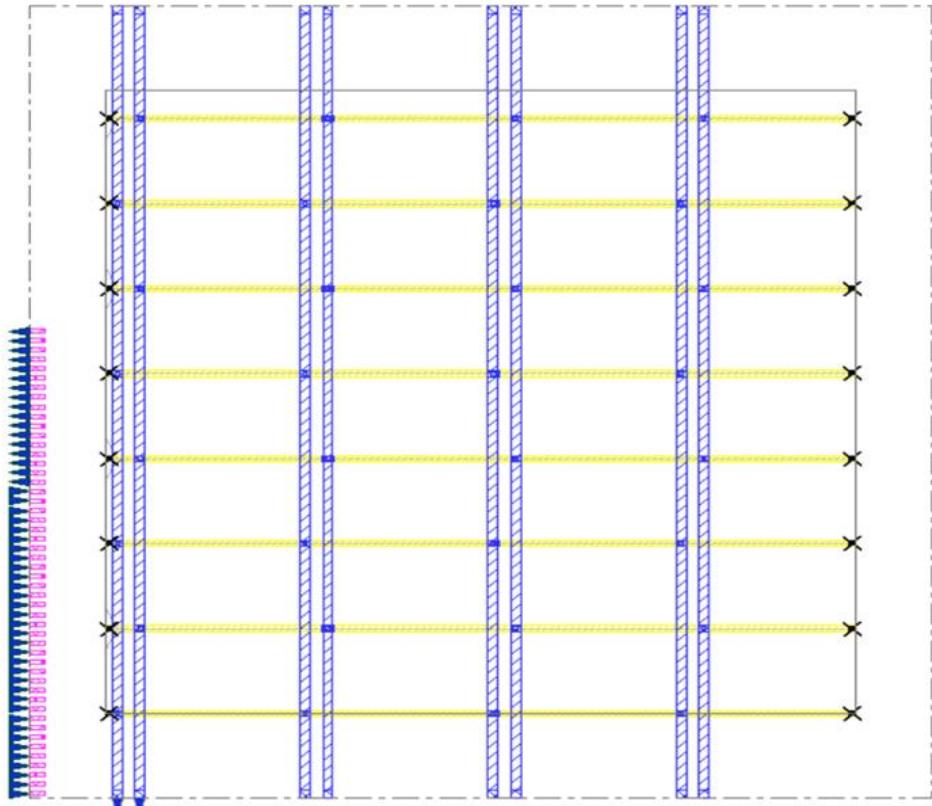


Figure 28 : M1 horizontal rail created

4.4.4 Placement

After placing standard cells in the design and optimization of placement, layout is shown below:

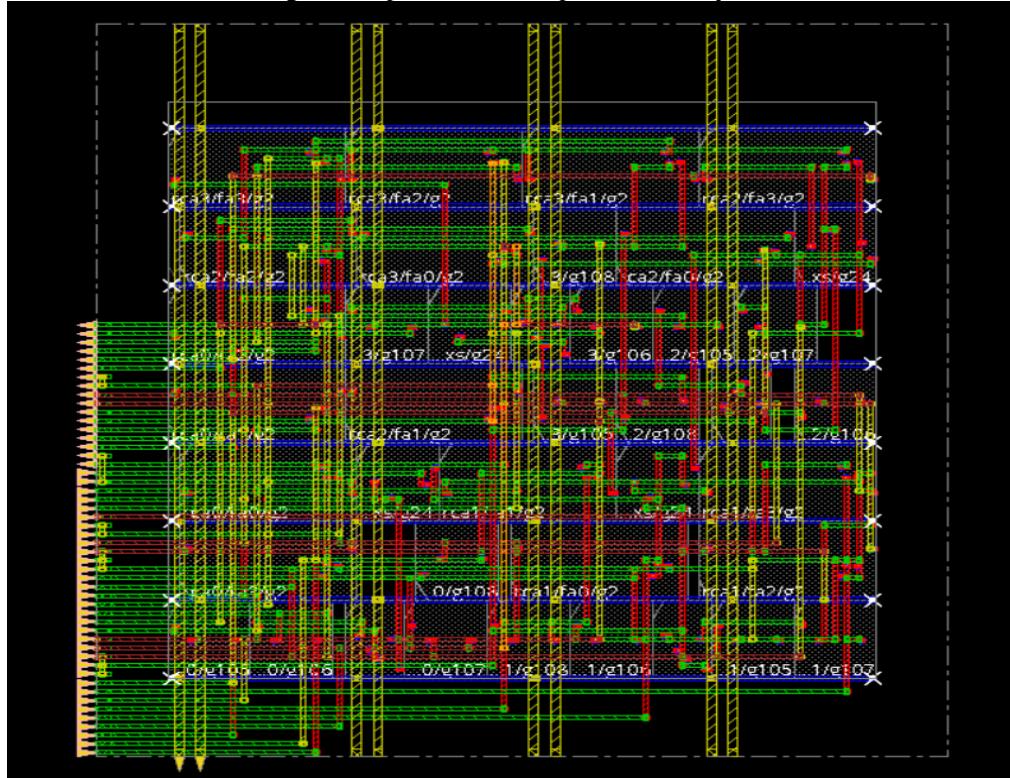


Figure 29 : Layout after optimizing placement

```

----- optDesign Final Summary -----
Setup views included:
func@BC_rcbest0.hold

+-----+
|   Setup mode | all    | default |
+-----+
| WNS (ns): | 0.000 | 0.000 |
| TNS (ns): | 0.000 | 0.000 |
| Violating Paths: | 0 | 0 |
| All Paths: | 0 | 0 |
+-----+ +-----+ +-----+ +-----+ +-----+ +-----+
+-----+ +-----+ +-----+ +-----+ +-----+ +-----+
| DRVs          |           Real           |           Total           |
|                 | Nr nets(terms) | Worst Vio | Nr nets(terms) |
+-----+ +-----+ +-----+ +-----+ +-----+ +-----+
| max_cap       | 0 (0)        | 0.000   | 0 (0)        |
| max_tran      | 0 (0)        | 0.000   | 0 (0)        |
| max_fanout    | 0 (0)        | 0       | 0 (0)        |
| max_length    | 0 (0)        | 0       | 0 (0)        |
+-----+ +-----+ +-----+ +-----+ +-----+ +-----+
Density: 95.238%
Routing Overflow: 0.00% H and 0.00% V
**optDesign ... cpu = 0:00:07, real = 0:00:07, mem = 1368.7M, totSessionCpu=0:00
:55 **
***WARN: (IMPOPT-3195): Analysis mode has changed.
Type 'man IMPOPT-3195' for more detail.
*** Finished optDesign ***
Removing temporary dont_use automatically set for cells with technology sites with no row.
*** Free Virtual Timing Model ... (mem=1368.7M)
**place_opt_design ... cpu = 0:00:10, real = 0:00:11, mem = 1278.3M ***
*** Finished GigaPlace ***

*** Summary of all messages that are not suppressed in this session:
Severity ID          Count  Summary
WARNING IMPEXT-3530    4  The process node is not set. Use the com...
WARNING IMPSP-9025     2  No scan chain specified/traced.
WARNING IMPSP-12502     2  Slack driven placement is disabled beca...
WARNING IMPOPT-3195     2  Analysis mode has changed.
WARNING IMPOPT-3564     1  The following cells are set dont_use tem...
*** Message Summary: 11 warning(s), 0 error(s)

```

Figure 30 : Command Line Report after optimizing placement

Here we get, Density =95.238%

4.4.5 Clock Tree Synthesis (CTS)

After synthesizing clock tree for this CSA design, we get the following timing reports:

```

innovus 18> report_timing
#####
# Generated by: Cadence Innovus 16.10-p004_1
# OS: Linux x86_64(Host ID CadenceServer3.localdomain)
# Generated on: Sat Dec 7 19:13:08 2024
# Design: csa_top
# Command: report_timing
#####
No constrained timing paths found.
Design may not be constrained or library is missing timing information.

```

Figure 31 : Timing reports

4.4.6 Running Nano route

In this step, Nano Route is employed to further refine routing and ensure timing closure in the design, even though timing has already been achieved. This process is essential, as timing closure is not fully reliable until crosstalk has been analyzed and mitigated.

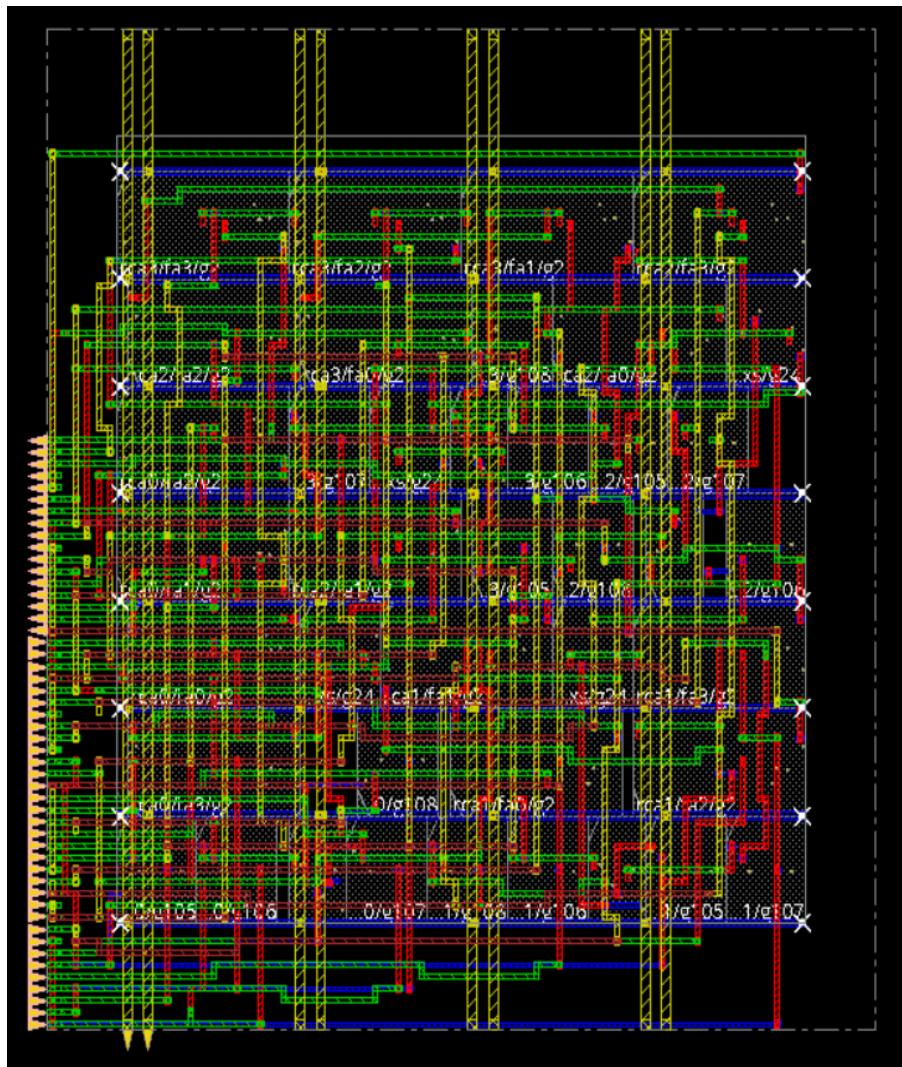


Figure 32 : Layout View after Nano route '

4.4.7 Post-Route timing and SI Optimization

```

----- optDesign Final SI Timing Summary -----
----- Setup views included:
func@BC_rcbest0.hold
Hold views included:
func@BC_rcbest0.hold

+-- Setup mode +--+ all +--+ default +-
| WNS (ns): | 0.000 | 0.000 |
| TNS (ns): | 0.000 | 0.000 |
| Violating Paths: | 0 | 0 |
| All Paths: | 0 | 0 |
+-- Hold mode +--+ all +--+ default +-
| WNS (ns): | 0.000 | 0.000 |
| TNS (ns): | 0.000 | 0.000 |
| Violating Paths: | 0 | 0 |
| All Paths: | 0 | 0 |
+-- DRVs +--+ Real +--+ Total +-
| DRVs | Nr nets(terms) | Worst Vio | Nr nets(terms) |
| max_cap | 0 (0) | 0.000 | 0 (0) |
| max_tran | 0 (0) | 0.000 | 0 (0) |
| max_fanout | 0 (0) | 0 | 0 (0) |
| max_length | 0 (0) | 0 | 0 (0) |
+-- Density: 95.238%
Total number of glitch violations: 0
**optDesign ... cpu = 0:00:11, real = 0:00:12, mem = 1482.9M, totSessionCpu=0:02
:18 **
ReSet Options after AAE Based Opt flow
*** Finished optDesign ***
Removing temporary dont_use automatically set for cells with technology sites wi
th no row.

```

Figure 33 : SI timing summary

```

innovus 24> innovus 24> *** Starting Verify Geometry (MEM: 1483.8) ***
VERIFY GEOMETRY ..... Starting Verification
VERIFY GEOMETRY ..... Initializing
VERIFY GEOMETRY ..... Deleting Existing Violations
VERIFY GEOMETRY ..... Creating Sub-Areas
..... bin size: 1920
VERIFY GEOMETRY ..... SubArea : 1 of 1
VERIFY GEOMETRY ..... Cells : 0 Viols.
VERIFY GEOMETRY ..... SameNet : 0 Viols.
VERIFY GEOMETRY ..... Wiring : 0 Viols.
VERIFY GEOMETRY ..... Antenna : 0 Viols.
VERIFY GEOMETRY ..... Sub-Area : 1 complete 0 Viols. 0 Wrngs.
VG: elapsed time: 0.00
Begin Summary ...
Cells : 0
SameNet : 0
Wiring : 0
Antenna : 0
Short : 0
Overlap : 0
End Summary

Verification Complete : 0 Viols. 0 Wrngs.

*****End: VERIFY GEOMETRY*****
*** verify geometry (CPU: 0:00:00.1 MEM: 169.7M)

```

Figure 34 : Geometry Verification

4.4.8 Adding filler cell and metal filling

To prevent potential DRC errors, it is advisable to insert fill cells to close gaps between placed standard cells. This step not only helps avoid design rule violations but also enhances the mechanical stability and substrate uniformity of the chip, contributing to a more robust final layout.

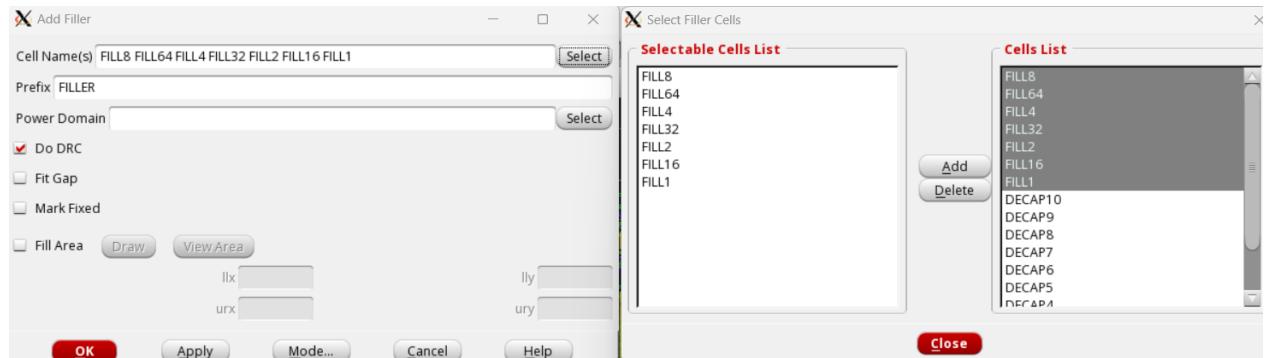


Figure 35 :Adding filler Cell

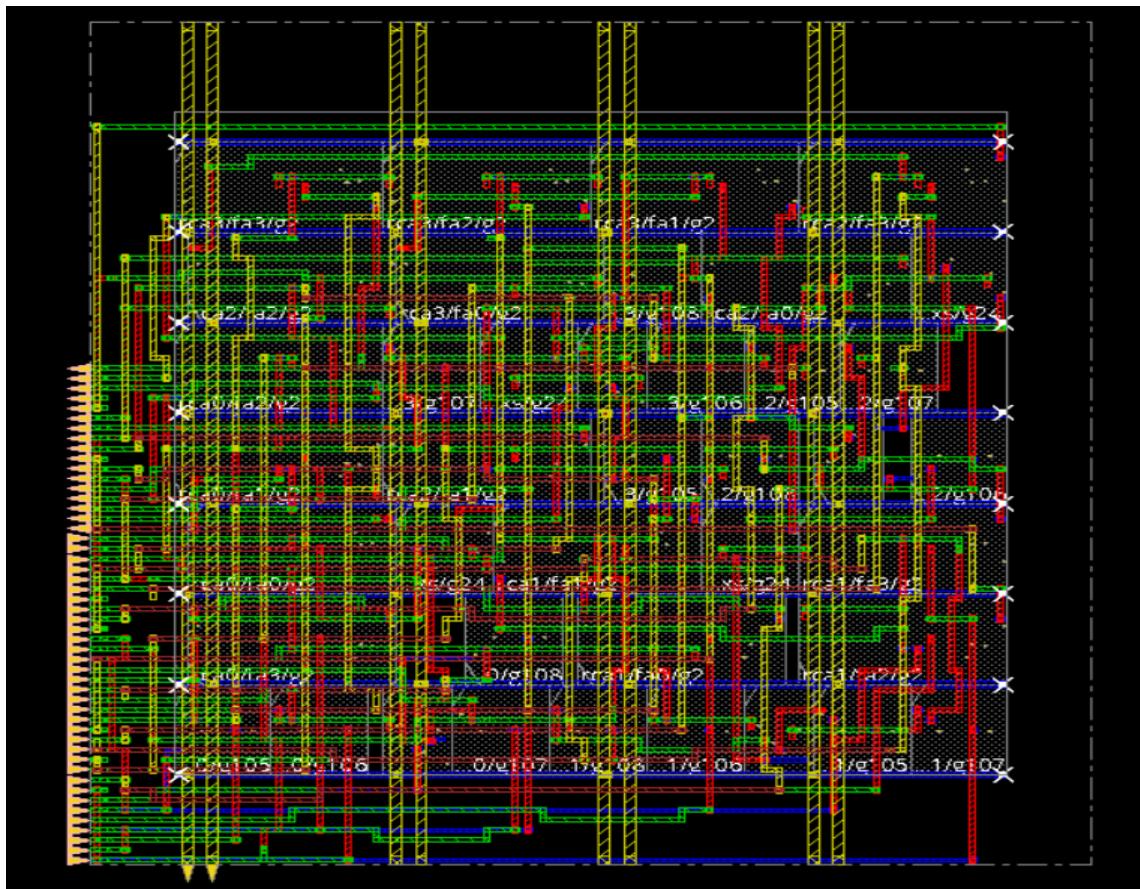


Figure 36 : Layout view after adding filler cell

```

innovus 26> Type 'man IMPSP-5217' for more detail.
*INFO: Adding fillers to top-module.
*INFO: Added 0 filler inst (cell FILL64 / prefix FILLER).
*INFO: Added 0 filler inst (cell FILL32 / prefix FILLER).
*INFO: Added 0 filler inst (cell FILL16 / prefix FILLER).
*INFO: Added 0 filler inst (cell FILL8 / prefix FILLER).
*INFO: Added 2 filler insts (cell FILL4 / prefix FILLER).
*INFO: Added 1 filler inst (cell FILL2 / prefix FILLER).
*INFO: Added 10 filler insts (cell FILL1 / prefix FILLER).
*INFO: Total 13 filler insts added - prefix FILLER (CPU: 0:00:00.0).
For 13 new insts, *** Applied 2 GNC rules (cpu = 0:00:00.0)

```

Figure 37 : List of filler cell

In our layout optimization, seven types of fillers (FILL1, FILL2, FILL4, FILL8, FILL16, FILL32, and FILL64) were available, but only three types (FILL1, FILL2, and FILL4) were utilized. A total of 13 filler instances were added, comprising 10 instances of FILL1, 1 instance of FILL2, and 2 instances of FILL4.

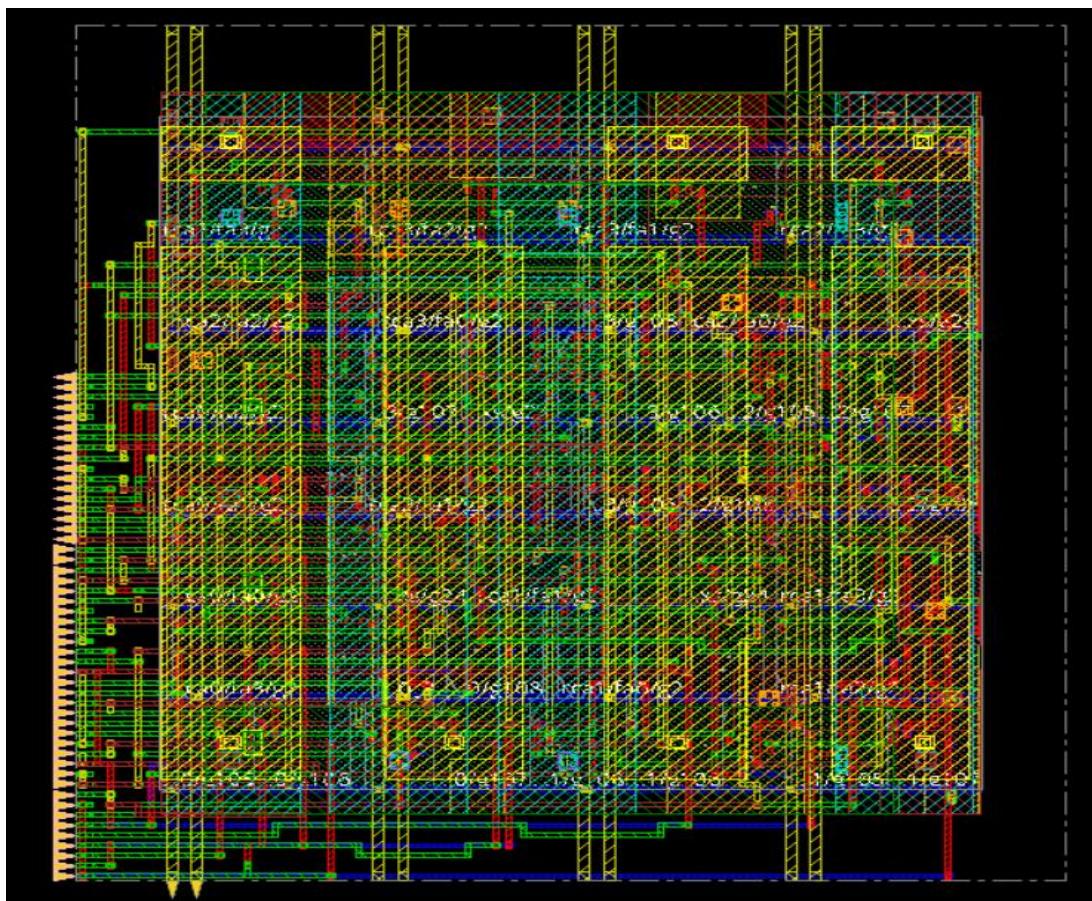


Figure 38 :Layout view after adding metal fill

```

*****End: VERIFY GEOMETRY*****
*** verify geometry (CPU: 0:00:00.1 MEM: 174.3M)

innovus 28> *** Starting Verify Geometry (MEM: 1654.3) **

VERIFY GEOMETRY ..... Starting Verification
VERIFY GEOMETRY ..... Initializing
VERIFY GEOMETRY ..... Deleting Existing Violations
VERIFY GEOMETRY ..... Creating Sub-Areas
..... bin size: 1920
VERIFY GEOMETRY ..... SubArea : 1 of 1
VERIFY GEOMETRY ..... Cells : 0 Viols.
VERIFY GEOMETRY ..... SameNet : 0 Viols.
VERIFY GEOMETRY ..... Wiring : 0 Viols.
VERIFY GEOMETRY ..... Antenna : 0 Viols.
VERIFY GEOMETRY ..... Sub-Area : 1 complete 0 Viols. 0 Wrngs.

VG: elapsed time: 0.00
Begin Summary ...
Cells : 0
SameNet : 0
Wiring : 0
Antenna : 0
Short : 0
Overlap : 0
End Summary

Verification Complete : 0 Viols. 0 Wrngs.

*****End: VERIFY GEOMETRY*****
*** verify geometry (CPU: 0:00:00.1 MEM: 0.0M)

```

Figure 39: Verify Geometry after metal fill

```

***** Start: VERIFY CONNECTIVITY *****
Start Time: Sat Dec 7 19:33:05 2024

Design Name: csa_top
Database Units: 2000
Design Boundary: (0.0000, 0.0000) (14.4000, 15.9600)
Error Limit = 1000; Warning Limit = 50
Check all nets

Begin Summary
Found no problems or warnings.
End Summary

End Time: Sat Dec 7 19:33:05 2024
Time Elapsed: 0:00:00.0

***** End: VERIFY CONNECTIVITY *****
Verification Complete : 0 Viols. 0 Wrngs.
(CPU Time: 0:00:00.0 MEM: 0.000M)

```

Figure 40: Verify Connectivity

4.4.9 Exporting Data

We exported GDS & netlist to run DRC in Virtuoso Assura. The generated file saved as ‘csau.merged.gds’.

4.4.10 Run DRC

The file generated in the previous step was imported into Cadence Virtuoso, where the DRC results were cross-checked, confirming that no errors were found.

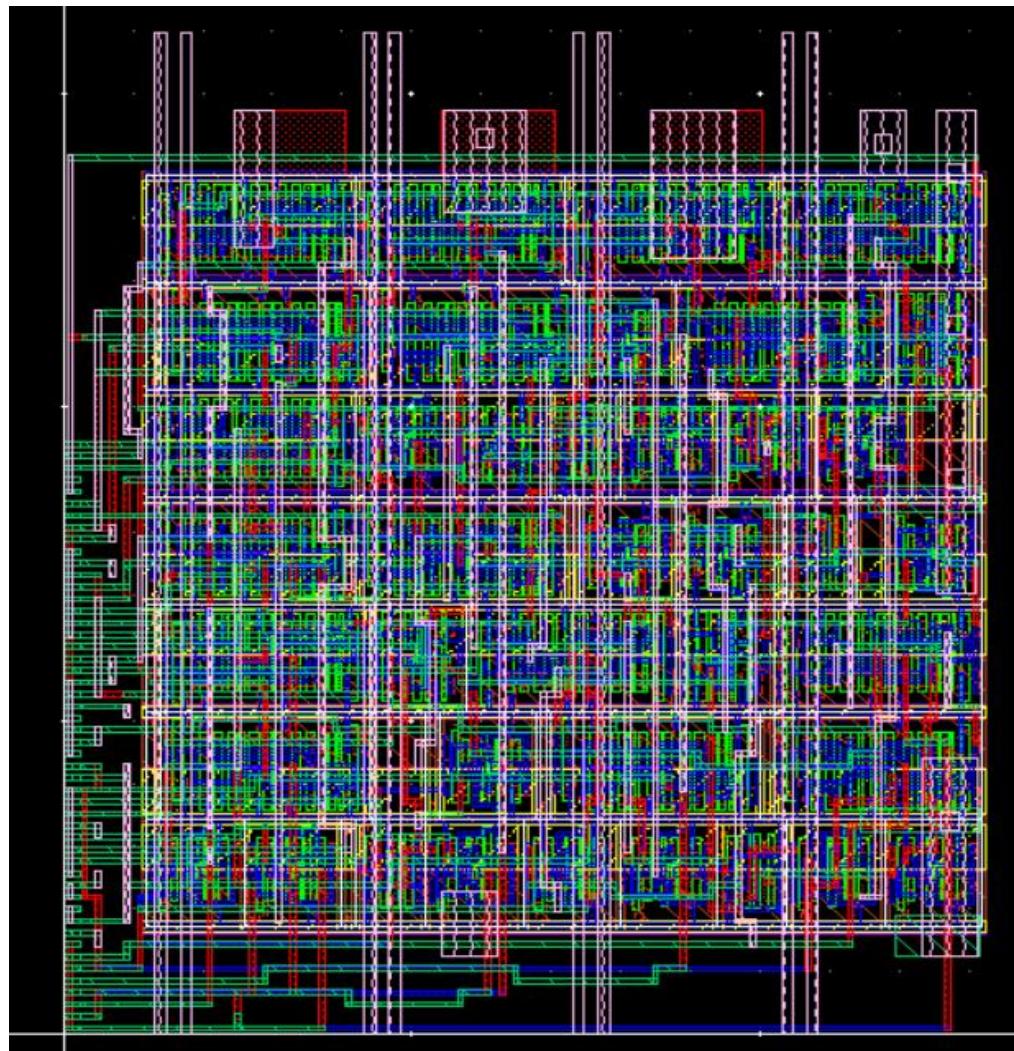


Figure 41: Final Layout at virtuoso



Figure 42: PVS Debug window showing no error

```

#####
# Outputting Results ...
#####

ONE LAYER BOOLEAN: Cumulative Time CPU =      0(s) REAL =      0(s)
TWO LAYER BOOLEAN: Cumulative Time CPU =      0(s) REAL =      0(s)
POLYGON TOPOLOGICAL: Cumulative Time CPU =    0(s) REAL =      0(s)
POLYGON MEASUREMENT: Cumulative Time CPU =    0(s) REAL =      0(s)
      SIZE: Cumulative Time CPU =      0(s) REAL =      0(s)
EDGE TOPOLOGICAL: Cumulative Time CPU =      0(s) REAL =      0(s)
EDGE MEASUREMENT: Cumulative Time CPU =      0(s) REAL =      0(s)
      STAMP: Cumulative Time CPU =      0(s) REAL =      0(s)
ONE LAYER DRC: Cumulative Time CPU =      0(s) REAL =      0(s)
TWO LAYER DRC: Cumulative Time CPU =      0(s) REAL =      0(s)
      NET AREA: Cumulative Time CPU =      0(s) REAL =      0(s)
      DENSITY: Cumulative Time CPU =      0(s) REAL =      0(s)
MISCELLANEOUS: Cumulative Time CPU =      0(s) REAL =      0(s)
      CONNECT: Cumulative Time CPU =      0(s) REAL =      0(s)
      DEVICE: Cumulative Time CPU =      0(s) REAL =      0(s)
      ERC: Cumulative Time CPU =      0(s) REAL =      0(s)
PATTERN_MATCH: Cumulative Time CPU =      0(s) REAL =      0(s)
DFM FILL: Cumulative Time CPU =      0(s) REAL =      0(s)

Total CPU Time           : 1(s)
Total Real Time          : 1(s)
Peak Memory Used         : 20(M)
Total Original Geometry : 1170(6100)
Total DRC RuleChecks    : 562
Total DRC Results        : 0 (0)
Summary can be found in file csa.sum
ASCII report database is /home/vlsi06/eee468_G6_2024/Project/Fnr_1516/csa.drc_errors.ascii
Checking in all SoftShare licenses.

Design Rule Check Finished Normally. Sat Dec 7 19:41:34 2024

```

Figure 43 : DRC Check with clean result

We can see, there was no DRC error in our design.

4.4.11 Utilization Report

Reporting Utilizations.....

```

Core utilization = 95.454545
Effective Utilizations
Average module density = 1.000.
Density for the design = 1.000.
      = stdcell_area 420 sites (144 um^2) / alloc_area 420 sites (144 um^2).
Pin Density = 0.5095.
      = total # of pins 214 / total area 420.
-
```

4.4.12 Power Report

```

Begin Static Power Report Generation
*
```

Total Power

Total Internal Power:	0.00279580	72.0578%
Total Switching Power:	0.00107507	27.7084%
Total Leakage Power:	0.00000907	0.2338%
Total Power:	0.00387994	

```

Ended Static Power Report Generation: (cpu=0:00:00, real=0:00:00,
mem(process/total)=932.21MB/932.21MB)
```

5 Design Analysis and Optimization

5.1 Result Analysis

❖ Speed and Latency

The CSKA significantly reduces propagation delay by allowing carries to skip over blocks when certain conditions are met. This feature drastically minimizes the time required for carry propagation, especially as the number of bits increases. Simulation results show that the CSKA's latency decreases with the implementation of skip logic when compared to the conventional ripple carry adder, where carry propagates sequentially through all bits. The block-based structure of CSKA ensures that carry generation and propagation within individual blocks is independent, contributing to an overall faster computation.

Comparative analysis with traditional RCA and CLA shows that CSKA offers a compromise between the fast carry generation of CLA and the simplicity of RCA. For large bit-width operations, CSKA outperforms RCA by skipping unnecessary carry calculations, which helps reduce the overall time for addition

❖ Power Consumption

Power analysis was performed to assess the energy efficiency of the CSKA. Given that power consumption is a critical factor in modern integrated circuits, the design was optimized for low power while maintaining performance. By reducing the number of blocks and optimizing the size of each block, the CSKA design minimizes power consumption, which increases linearly with the size of the adder. The results demonstrate that CSKA uses less power compared to the CLA and RCA, particularly in large-scale applications, due to the reduced carry propagation.

Power consumption results were also verified using synthesis tools like Cadence and Xilinx, where the CSKA showed lower dynamic power consumption, making it more suitable for battery-operated or energy-efficient devices

❖ Area Efficiency

In terms of area efficiency, the CSKA's block-based architecture ensures optimal utilization of chip area. While the CLA requires a more complex circuit with additional gates to compute carry operations, CSKA maintains a simpler structure by limiting the number of gates needed to handle carries within each block. Area optimization was particularly crucial for minimizing silicon usage, as modern VLSI designs aim for compact implementations without sacrificing performance.

Simulation results show that CSKA, while not as compact as RCA due to its additional skip logic, strikes a good balance by optimizing block sizes to minimize the number of gates without compromising speed. This makes it suitable for large bit-width applications that require both high performance and efficient area usage

❖ Overall Performance

The overall performance of the CSKA is evaluated by combining the results from speed, power, and area analysis. The adder demonstrates high efficiency with reduced propagation delay, lower power consumption, and optimized area usage. These results align with the objectives of the project, showcasing that CSKA is a viable solution for high-performance, energy-efficient, and compact digital systems.

5.2 Novelty

5.2.1 Signed Number Consideration and Flag Addition

In our design, we have considered A[15:0] and B[15:0] as signed numbers. First of all we have identified total no. of types for inputs A[15:0], B[15:0] and Cin according to the set flags N(Negative Flag), Z(Zero Flag), Cout(Output Carry) and V(Overflow Flag).

5.2.2 Coverage

i) Applying constraints in testcase01.sv

```
class testcase01 extends transaction;
constraint c_s {
    A inside {[0:65535]};//maximizing coverage
    B inside {[0:65535]};//maximizing coverage
    Cin inside {[0:1]};
    //S inside {[0:1]}; //maximizing coverage
}
endclass:testcase01
```

Here, A[15:0] and B[15:0] are forced to vary between 0 to $2^{16}-1=65535$. So, that for each loop in the testbench, different random values of A and B are created for checking all the eight cases stated above.

ii) Count Variation:

Constraints in testcase01.sv allow A and B to be variable among different test cases but they are not randomized among different runs. That's why at every run, number of test cases 'count' will be varied. So, that different values of A[15:0] and B[15:0] are tested and checked.

iii) Test Vector approach:

Previously we have tried to implement 4 type of cases using testvector. C=Carry, V=Overflow

Case1: {C,V}=2'b00;// 10 cases
Case2: {C,V}=2'b01;// 10 cases
Case3: {C,V}=2'b10;// 10 cases
Case4: {C,V}=2'b11;// 10 cases

But testvector approach only tests few fixed cases. No variability here. That's why this approach was dropped.

iv) Adding verification algorithm to check which one of the eight cases are tested each time at scoreboard.sv.

```
real Flags [8]='{default: 64'b0}; // to count the occurrence of each cases
real Pass [8]='{default: 64'b0}; // to count passed occurrence of each test cases
real Fail [8]='{default: 64'b0}; // to count failed occurrence of each test cases
```

```

real pc [8],f_pc [8], p_pc [8]; //pc for each testcase percentage occurrence
                                // f_pc for each testcase passing percentage occurrence
                                // p_pc for each testcase passing percentage occurrence
logic [2:0] X,Y; // accumulate N, C,V

```

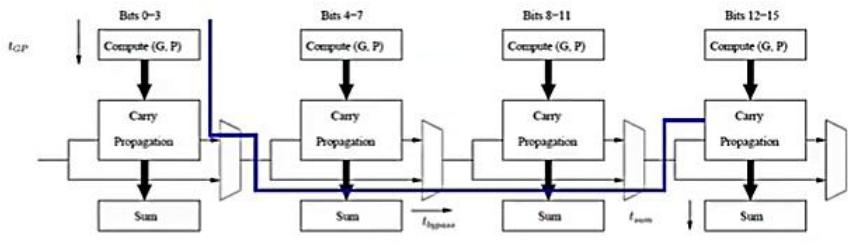
5.3 Design Optimization

5.3.1 Delay minimization, Higher Speed, and Lower Latency

We've achieved this condition by minimizing delay of the adder both theoretically and practically.

Theoretical Delay minimization:

For higher bit addition architectural level optimization is preferable to circuit level optimization. For 16 bit carry skip addition, we need to divide 16 bits into some k bits intermediate ripple carry blocks. If k=1, then area required for synthesis will be larger and create more delay. If k=16 then it will be a ripple carry adder. So, delay could be further minimized. So, partition value k has to be between 1 to 16 and an integer number.



$$t_{carry\text{-}skip} = t_{GP} + Mt_{carry} + \left(\frac{N}{M} - 1\right)t_{bypass} + (M - 1)t_{carry} + t_{sum}$$

- ▶ Still proportional to N , however linear with N/M
- ▶ Still better than ripple adder.
- ▶ Carry skip starts showing lesser delay for $N > 4 - 8$

Figure 44 :Carry Skip Chain

Now we'll minimize delay by plotting tcsa with respect to intermediate block size M/k.

Matlab Code:

```

clc;clear all;close all;
t_pg=4;%4ns
t_carry=2.5;%2.5ns
t_mux=2;%2ns
N=16;
K = linspace(1,16,1000);
t_PG=t_pg*ones(1,1000);
t_csa=t_PG+ (K*t_carry) + ((N./K)-1).*t_mux +(K-1)*t_carry;
figure(1);
% subplot(211);
plot(K,t_csa, 'LineWidth',2);hold on
xlabel('Number of bits per stage(K)');
ylabel('Delay of carry skip adder(t_csa) in ns');
title('Optimization of Carry Skip Adder');

```

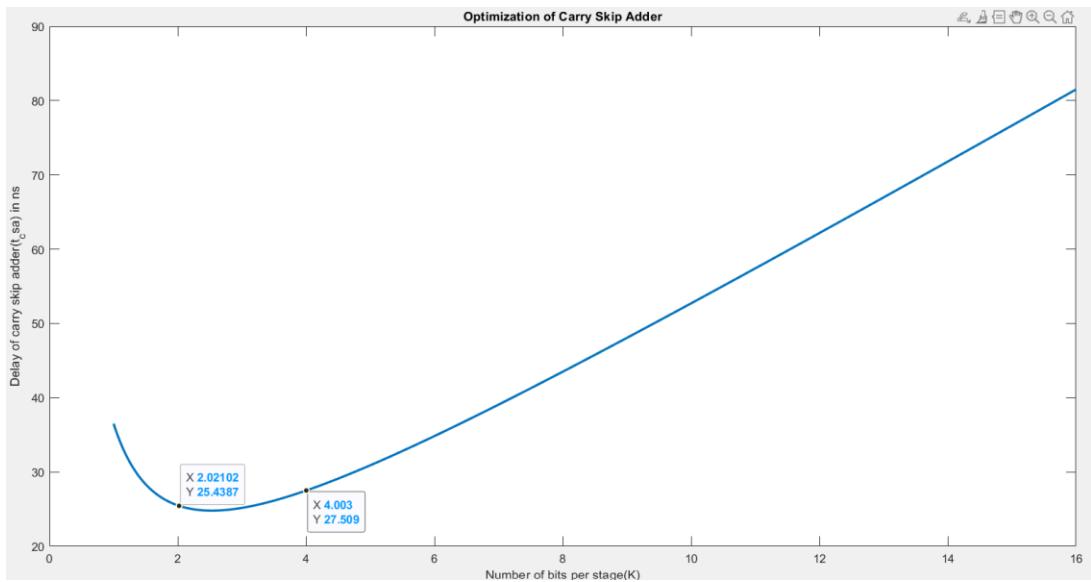


Figure 45 : Delay Vs Block size (k)

Here delay is lowest if $k=2$ or $k=4$ keeping t_{PG} , t_{mux} and t_{carry} constant

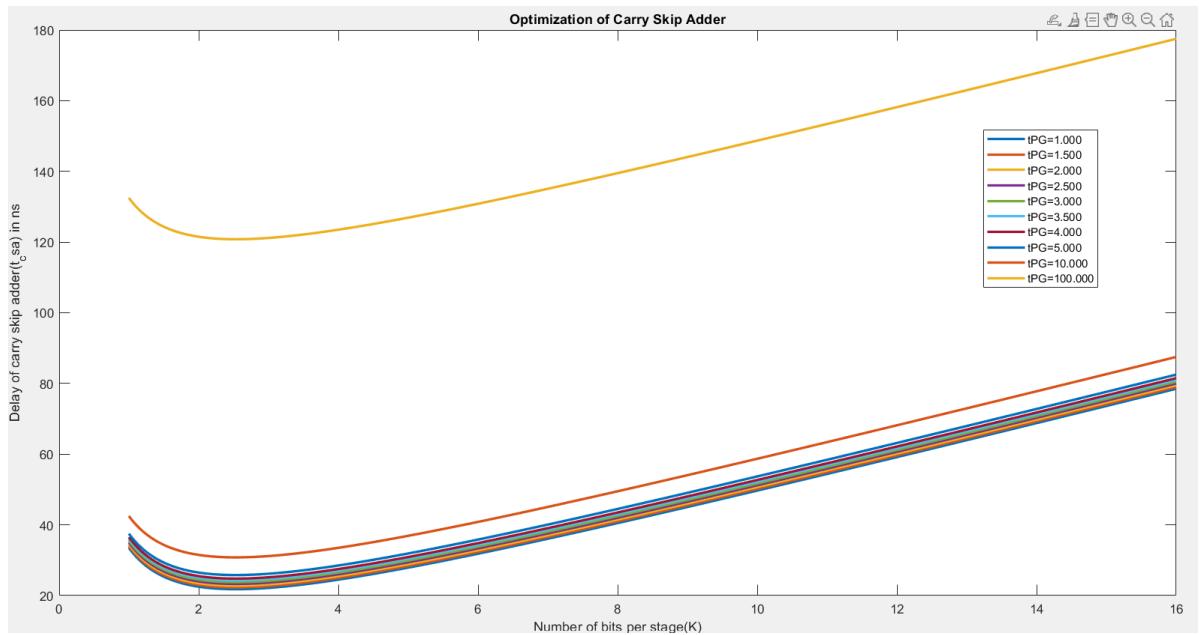


Figure 46 : Delay vs Block Size (k) Varying t_{PG}

Here delay is lowest if $k=2$ or $k=4$ keeping t_{PG} variable, t_{mux} and t_{carry} constant

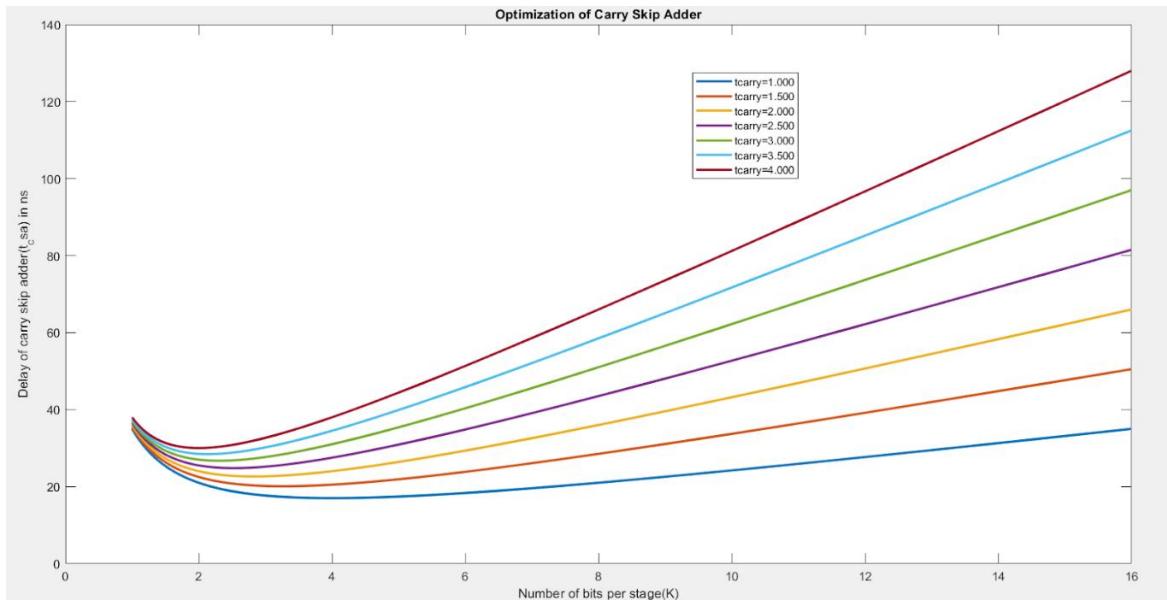


Figure 47;: Delay vs Block Size (k) Varying t_{carry}

Here delay is lowest if $k=2$ or $k=4$ keeping t_{PG} and t_{mux} constant , t_{carry} variable

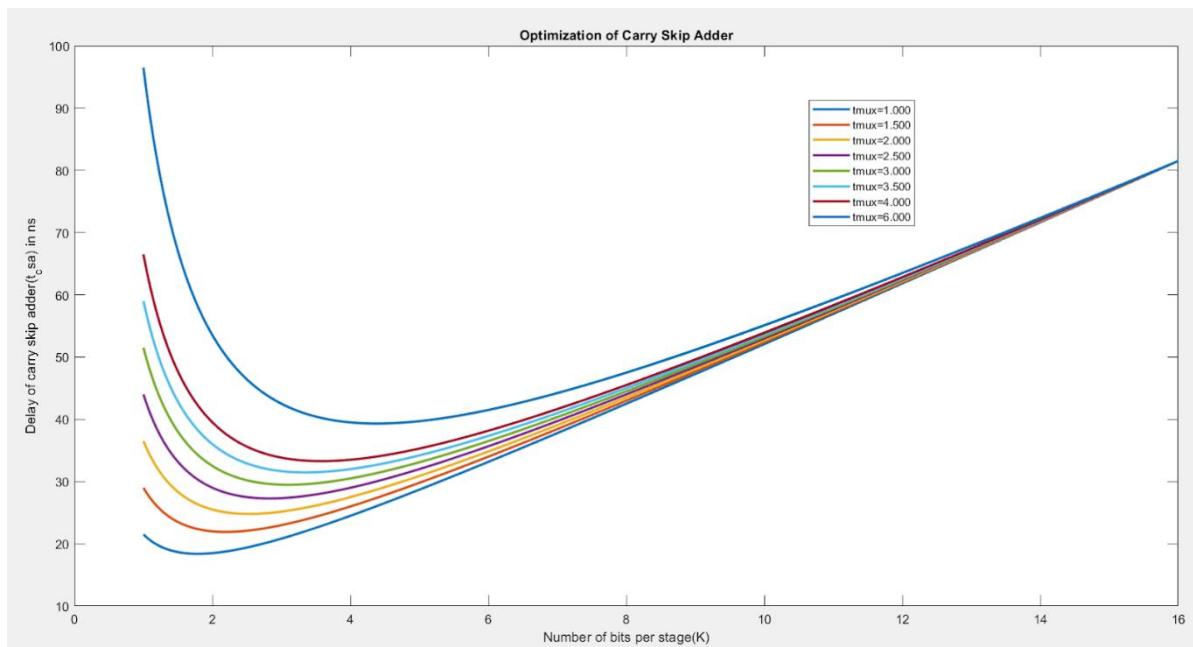


Figure 48 : : Delay vs Block Size (k) Varying t_{MUX}

Here delay is lowest if $k=2$ keeping t_{PG} and t_{carry} constant , t_{mux} variable

Verification With Simulation:

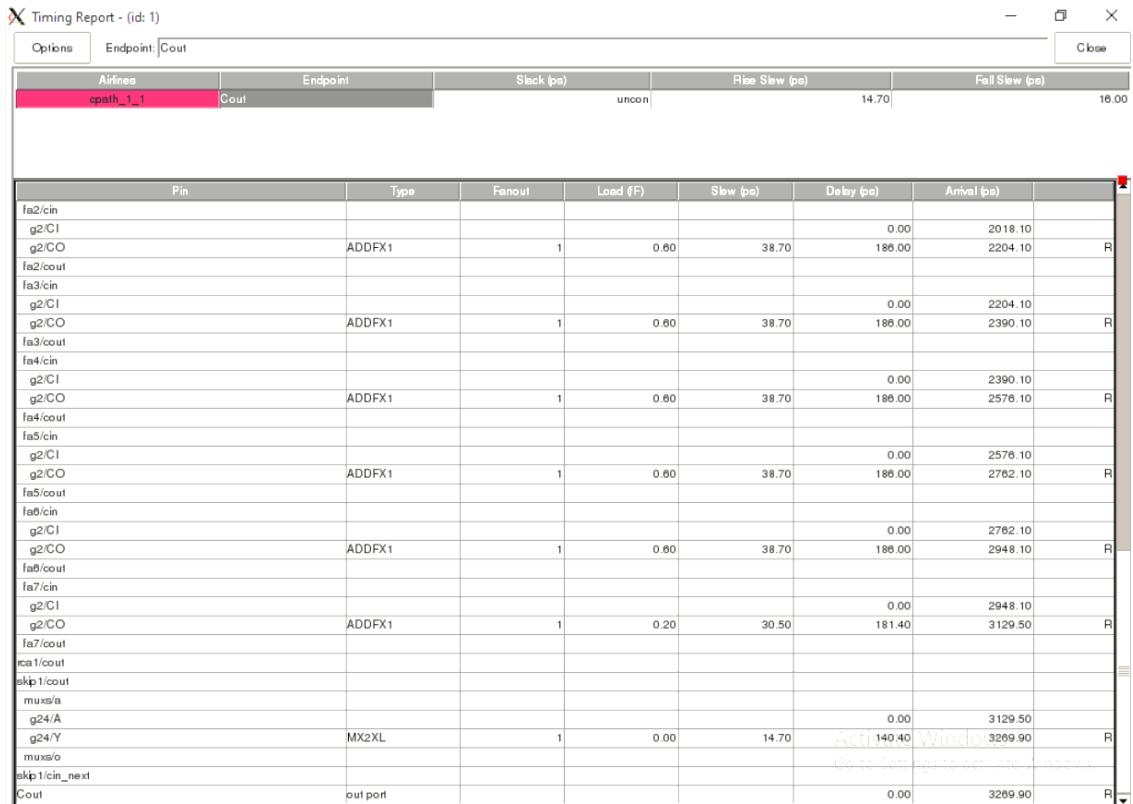


Figure 49: Timing report for $k=2$

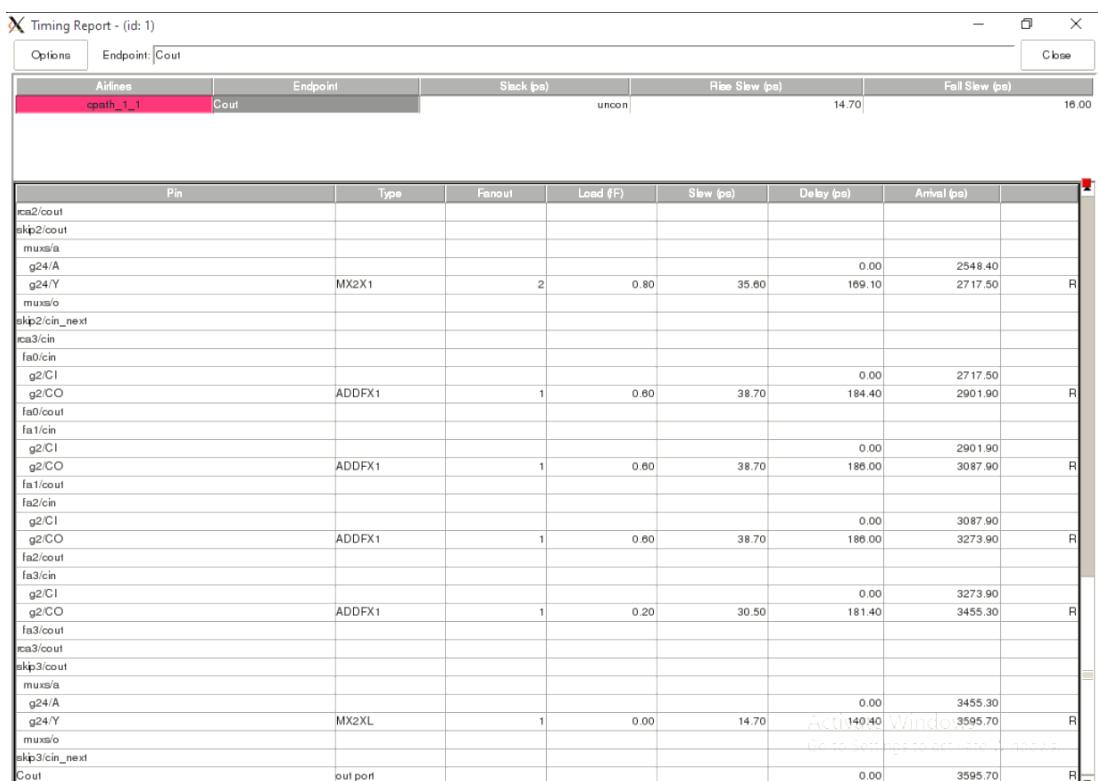


Figure 50: Timing report for $k=4$

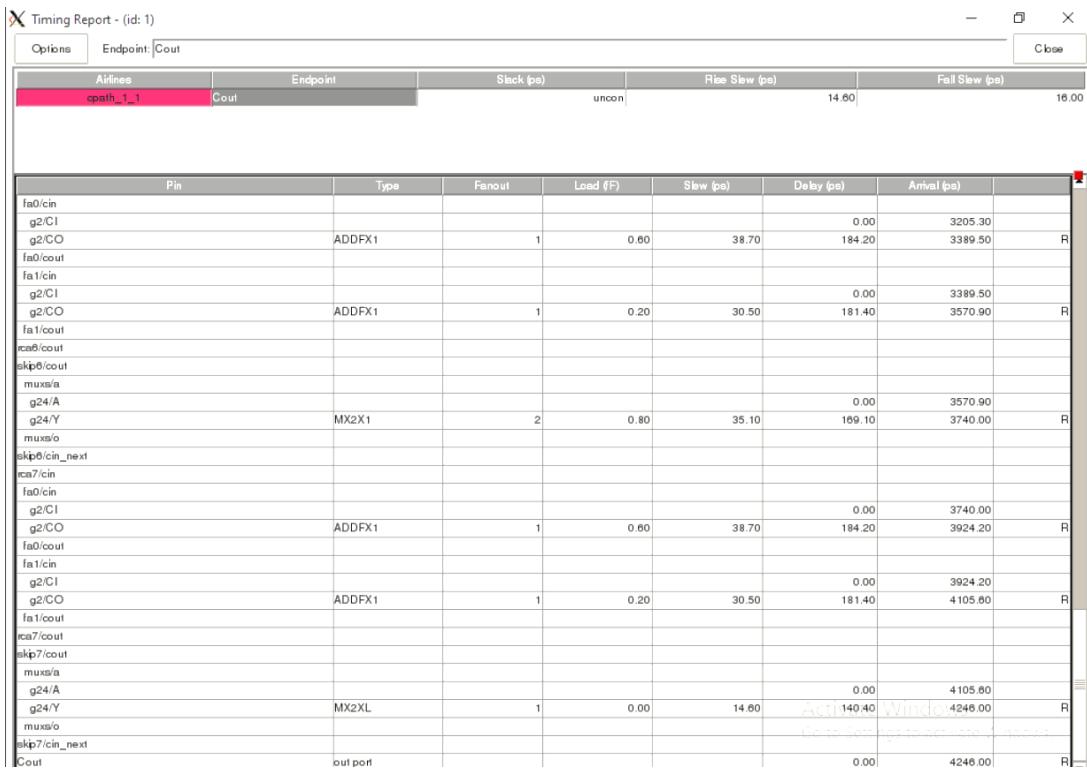


Figure 51 : Timing report for k=8

Here, we can see for k=2 area and power loss is more, for k=8 delay is more. So, optimum design is to go with k=4.

5.3.2 Minimum Power Consumption

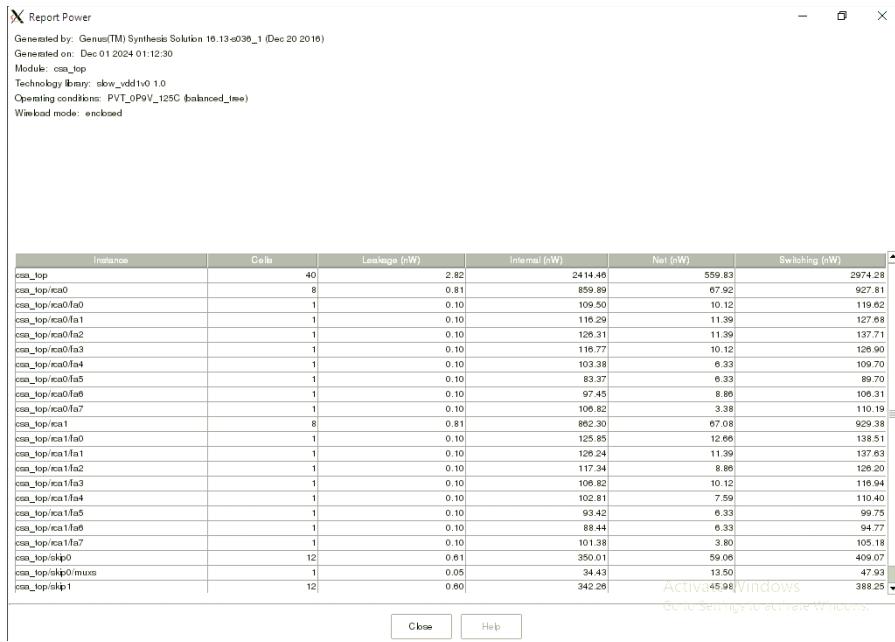


Figure 52: Power Report for k=8

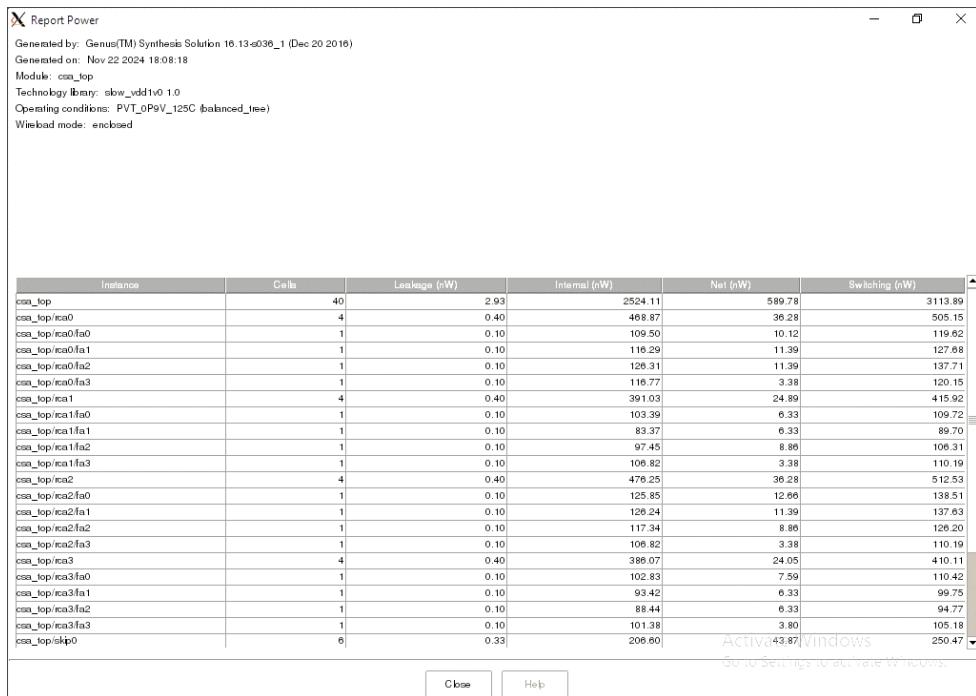


Figure 53 : Power Report for k=4

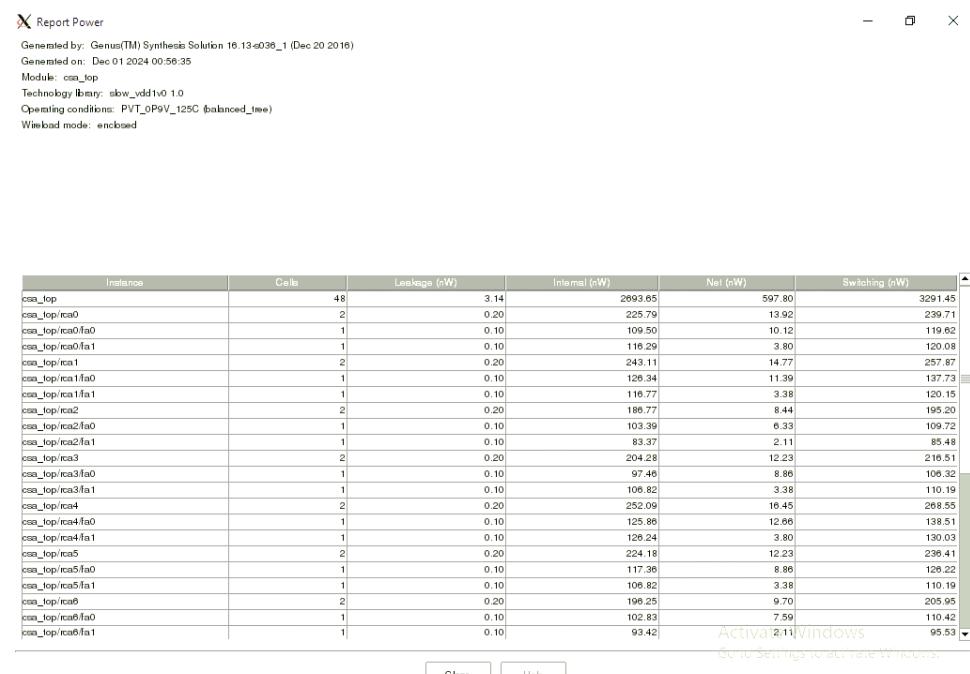


Figure 54 : Power Report for k=2

5.3.3 Area Minimization

Report Area

Generated by: Genus(TM) Synthesis Solution 16.13-e036_1 (Dec 20 2016)
Generated on: Dec 01 2024 01:08:30
Module: csa_top
Technology library: slow_vdd1v0 1.0
Operating conditions: PVT_Op9V_125C (balanced_lres)
Wireload mode: enclosed

Instance	Cells	Cell Area	Net Area	Total Area	Wireload	WL Flag
csa_top	40	134.06	0.00	134.06<none>	(D)	
csa_top/ica0	8	41.04	0.00	41.04<none>	(D)	
csa_top/ica0/fa0	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica0/fa1	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica0/fa2	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica0/fa3	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica0/fa4	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica0/fa5	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica0/fa6	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica0/fa7	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica1	8	41.04	0.00	41.04<none>	(D)	
csa_top/ica1/fa0	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica1/fa1	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica1/fa2	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica1/fa3	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica1/fa4	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica1/fa5	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica1/fa6	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica1/fa7	1	5.13	0.00	5.13<none>	(D)	
csa_top/skip0	12	25.99	0.00	25.99<none>	(D)	
csa_top/skip0/muxs	1	2.39	0.00	2.39<none>	(D)	
csa_top/skip1	12	25.99	0.00	25.99<none>	(D)	

Figure 55 : Area Report for k=8

Report Area

Generated by: Genus(TM) Synthesis Solution 16.13-e036_1 (Dec 20 2016)
Generated on: Nov 22 2024 18:10:55
Module: csa_top
Technology library: slow_vdd1v0 1.0
Operating conditions: PVT_Op9V_125C (balanced_lres)
Wireload mode: enclosed

Instance	Cells	Cell Area	Net Area	Total Area	Wireload	WL Flag
csa_top	40	136.80	0.00	136.80<none>	(D)	
csa_top/ica0	4	20.52	0.00	20.52<none>	(D)	
csa_top/ica0/fa0	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica0/fa1	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica0/fa2	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica0/fa3	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica1	4	20.52	0.00	20.52<none>	(D)	
csa_top/ica1/fa0	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica1/fa1	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica1/fa2	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica1/fa3	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica2	4	20.52	0.00	20.52<none>	(D)	
csa_top/ica2/fa0	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica2/fa1	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica2/fa2	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica2/fa3	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica3	4	20.52	0.00	20.52<none>	(D)	
csa_top/ica3/fa0	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica3/fa1	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica3/fa2	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica3/fa3	1	5.13	0.00	5.13<none>	(D)	
csa_top/skip0	6	13.68	0.00	13.68<none>	(D)	

Figure 56 : Area Report for k=4

X Report Area

Generated by: Genus(TM) Synthesis Solution 16.13<036_1 (Dec 20 2016)

Generated on: Dec 01 2024 00:53:06

Module: csa_top

Technology Library: slow_vdd1v0 1.0

Operating conditions: PVT_0P9V_125C (balanced_free)

Wireload mode: enclosed

Instance	Cells	Cell Area	Net Area	Total Area	Wireload	WL Flag
csa_top	48	147.74	0.00	147.74<none>	(D)	
csa_top/ica0	2	10.26	0.00	10.26<none>	(D)	
csa_top/ica0fa0	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica0fa1	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica1	2	10.26	0.00	10.26<none>	(D)	
csa_top/ica1fa0	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica1fa1	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica2	2	10.26	0.00	10.26<none>	(D)	
csa_top/ica2fa0	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica2fa1	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica3	2	10.26	0.00	10.26<none>	(D)	
csa_top/ica3fa0	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica3fa1	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica4	2	10.26	0.00	10.26<none>	(D)	
csa_top/ica4fa0	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica4fa1	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica5	2	10.26	0.00	10.26<none>	(D)	
csa_top/ica5fa0	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica5fa1	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica6	2	10.26	0.00	10.26<none>	(D)	
csa_top/ica6fa0	1	5.13	0.00	5.13<none>	(D)	
csa_top/ica6fa1	1	5.13	0.00	5.13<none>	(D)	

Close Help Go to Settings to activate Windows DWS

Figure 57 : Area Report for k=2

5.3.4 Overall Optimization of Block Size(k)

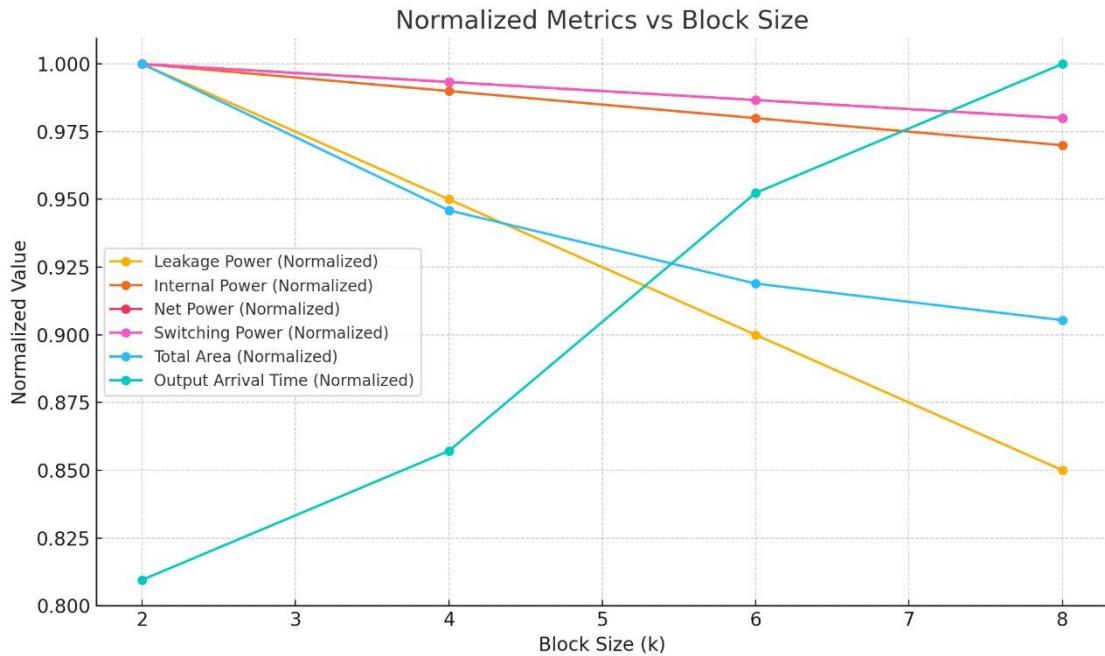


Figure 58 : Normalized Power, Area, & Output Arrival Time

The above graph is the comparison of the different parameter of the design vs the block size of the adder. All the parameter except the output arrival time are decreasing with the increasing the block size while the output arrival time is increasing with the increasing of the block size. We have chosen block size k-4 for optimal parameter condition.

5.3.5 Maximum Utilization of Chip Area

Die width (μm)	Die height (μm)	Die Size (sq μm)	Pin Spacing (μm)	Pin Depth (μm)	Pin Width (μm)	Area Density %	Strip width (μm)	Strip Spacing (μm)	Strip Number
16	19.95	319.2	0.35	0.28	0.14	65.359	0.16	0.2	10
14.2	13.08	185.74	0.2	0.28	0.14	Error	-	-	-
15.3	18.68	285.62	0.19	0.25	0.08	78.125	0.16	0.2	10
15	16.53	247.95	0.19	0.25	0.08	90.703	0.16	0.2	9
14.4	15.96	229.82	0.19	0.25	0.08	95.238	0.16	0.2	8

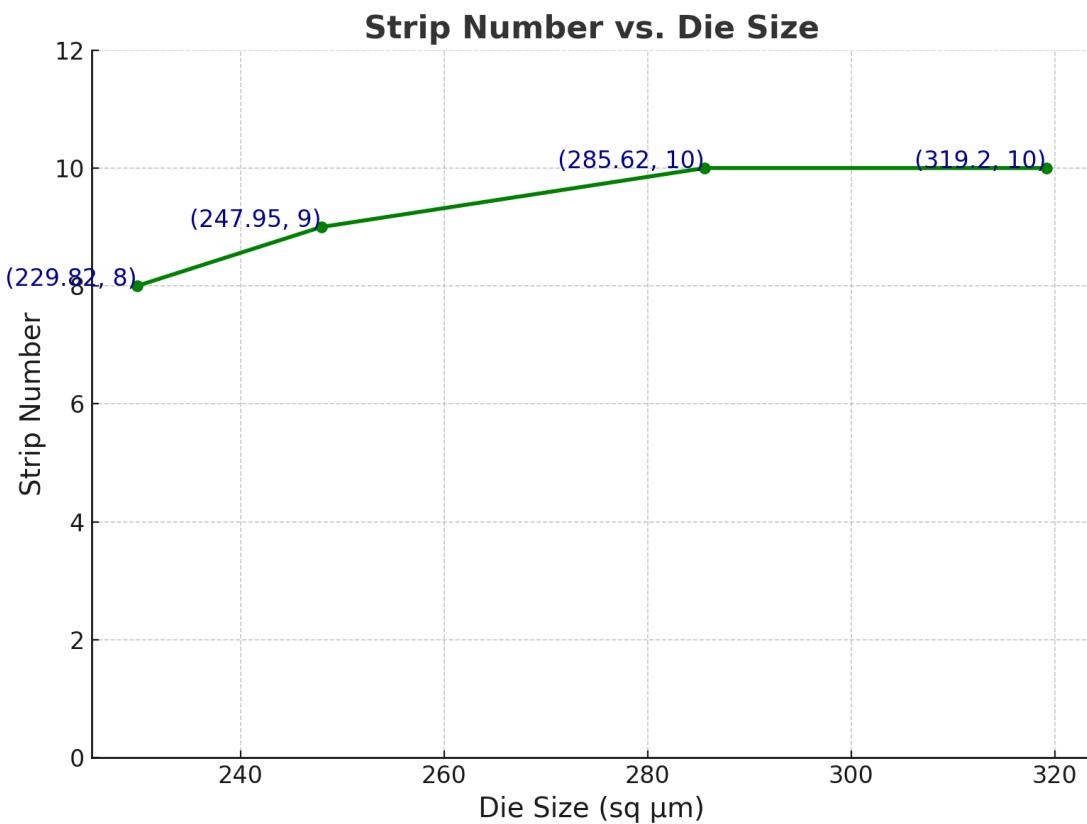


Figure 59 : Strip Number vs Die area

The graph here representing the Strip number with respect to Die area is increasing while the area is increasing . There is a relation of the strip width,depth, spacing , pin width, depth, pin spacing with strip number along with die area.

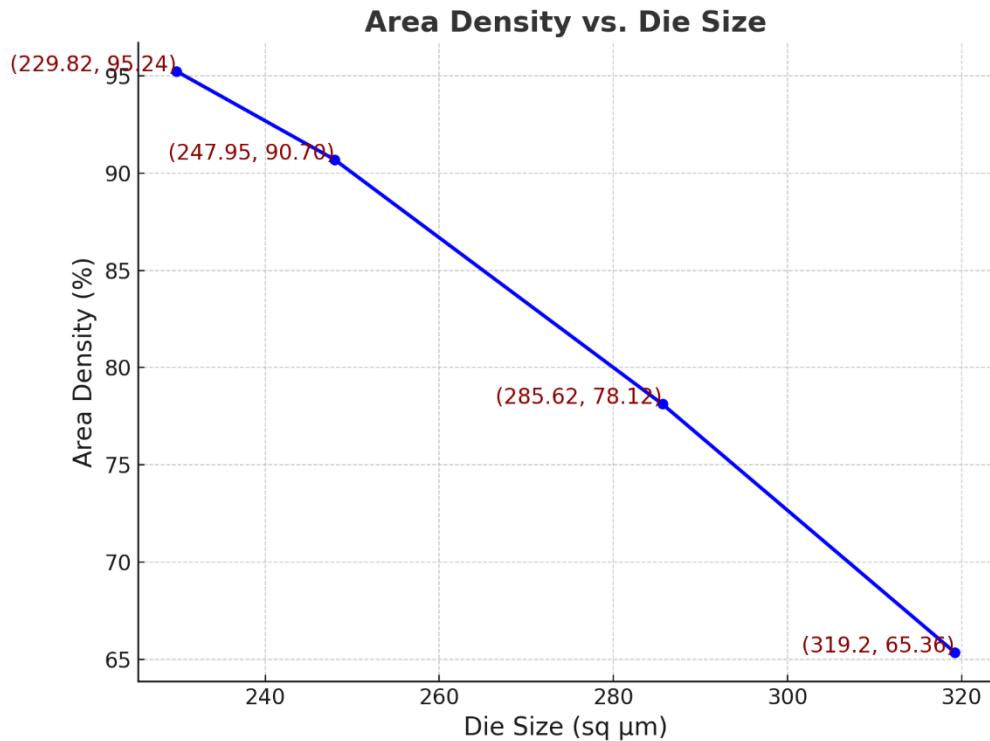


Figure 60 : Area density vs Die size

The graph indicates that if the area of the die is increased then the area density will be decreased. We have got the highest area density 95% at 229.82 sq um .

5.4 Design Considerations

1. Gate level design/RTL coding in design csa_top.v file so that synthesis is realistic and not a single adder/alu module.
2. Signed number implementation so that we can understand how they are handled in a practical digital circuit.
3. N,Z,C,V flag addition and handling for convenient adder operation.
4. Layered testbench implementation so that testbench engineers don't need to bother much if design is modified.
5. Coverage maximization so that the designed adder performs accurately for all possible cases.
6. Delay optimization for faster operation and lower latency.
7. Area and power loss minimization for higher density and stability.
8. PnR design with maximum utilisation.

6 Reflection on Individual and Team work

Functioning effectively as an individual and as a member or leader in diverse teams and multi-disciplinary settings is essential for the success of the Carry Skip Adder (CSA) project. As an individual, team member, or leader, effective communication skills are critical to convey ideas, share insights, and collaborate with colleagues from diverse backgrounds. Clear and open communication fosters a collaborative environment where team members can leverage their unique skills and perspectives.

In multi-disciplinary settings, where expertise from various engineering domains may be required, the ability to integrate insights from different disciplines becomes crucial. Engineers working on the CSA project must be adaptable, willing to learn from others, and capable of applying their expertise in a cross-functional context. Leadership skills are valuable for guiding the team, setting priorities, and ensuring that everyone is aligned with the project's objectives.

Understanding the strengths and weaknesses of team members allows for effective delegation and utilization of skills. Embracing diversity in thought and background enriches the problem-solving process, leading to innovative solutions for the complex challenges associated with CSA design. Team members who function well individually and collaboratively contribute to a dynamic and high-performing project environment.

By fostering an inclusive and collaborative culture, the CSA project benefits from the collective intelligence and diverse perspectives of its team members, ultimately enhancing the project's overall success and the quality of the engineered solution.

6.1 Individual Contribution of Each Member

Individual IDs	Contribution
1806052	Design coding, directed testbench, debugging, report writing and slide preparation.
1906070	Verification by directed and layered testbench, coverage maximization, delay minimization, synthesis.
1906076	RTL code writing, layered testbench writing, Optimization of power, performance and area, debugging.
1906077	Place and Route, chip design and utilization area maximization, report writing and presentation preparation.
1906080	Directed Testbench, Synthesis, timing, power consumption and area optimization, report writing and theoretical analysis.

6.2 Mode of Team Work

Our team work was distributed as:

- ✓ Design and directed testbench.
- ✓ Layered Testbench and coverage.
- ✓ Synthesis.
- ✓ Place and Route
- ✓ Optimization of Power, Performance and Area

Although all the works were distributed equally, there were also individual work distribution as given below:

1806052: Design and directed testbench.

1906070: Layered Testbench and coverage.

1906076: Optimization of Power, Performance and Area

1906077: Place and Route

1906080: Synthesis

6.3 Log Book of Project Implementation

Week	Milestone Achieved	Individual Role	Comments
Week 4	Project Assign	-----	-----
Week 5	Initial team discussion, distribution of tasks	All members participated	Finalized roles and timeline.
Week 6	RTL code writing completed	1906070,1906076	Initial code had minor bugs, debugging initiated.
Week 7	Directed testbench designed	1806052,1906076	Simulated the first set of cases successfully.
Week 8	Debugging of initial design completed	1906076,1906080	Successfully generated netlist; minor timing issues identified.
Week 9	Verification with layered testbench	1906070,1906076	Testbench validated performance metrics.
Week 10	Layered testbench written and coverage increased	1906070,1906080	Achieved 100% functional coverage in simulations.
Week 11	Synthesis of design	1806052, 1906080	Successfully generated
	Place and Route completed	1906076, 1906077	Final design utilization optimized to meet area constraints.
Week 12	Optimization of Power, Performance, and Area (PPA)	All member participated	All group do their part to optimize the output
	Final report and presentation slides completed	1806052,1906080	All member writes their part and mentioned two members do sorting and merging the report and presentation.

7 Summary

This report presents the successful design and implementation of a 16-bit Carry Skip Adder (CSKA) optimized for both signed and unsigned addition operations. The CSKA employs a block-based structure with skip logic to improve speed by reducing carry propagation delay. Key highlights of the project include the following:

Design and Implementation:

The adder was divided into smaller blocks, with ripple carry adders used within each block. Skip logic was implemented to evaluate propagate conditions and enable carry bypassing for enhanced performance.

Verification and Simulation:

Directed and layered testbenches were developed to verify functionality and performance. The simulations confirmed correctness under various input conditions.

Optimization:

The design underwent extensive Power, Performance, and Area (PPA) optimization, ensuring efficient utilization of resources. Delay minimization, power reduction, and area efficiency were achieved through iterative design refinement.

Physical Design:

The synthesis, floorplanning, and routing stages adhered to industry-standard design rules. Clean DRC and LVS reports validated the physical implementation's correctness and manufacturability.

In conclusion, the project demonstrates the practical application of advanced digital design principles in creating a high-performance adder. The CSKA's performance and efficiency make it a viable candidate for integration into modern high-speed digital systems.

8 Conclusion

Our project successfully demonstrates the design and implementation of a 16-bit Carry Skip Adder that achieves significant improvements in speed, power efficiency, and area utilization. The adoption of propagate-generate logic and skip mechanisms enables high-performance addition operations suitable for both signed and unsigned data, addressing diverse computational requirements.

Through rigorous simulation and layered verification, we achieved high functional coverage and ensured the design's correctness under varied test scenarios. The synthesis and Place and Route stages further optimized the design for timing closure and efficient area usage while adhering to industry-standard DRC checks. Our approach ensures the design is not only fast but also energy-efficient, making it ideal for integration into modern high-speed digital systems. Overall, the project outcomes highlight the potential of the Carry Skip Adder architecture in advancing the performance of arithmetic units, paving the way for its application in processors, signal processors, and other computational systems requiring high-speed arithmetic operations.

9 References

- Harris, D. M., & Harris, S. L. (2013). Digital Design and Computer Architecture. Morgan Kaufmann.A comprehensive guide on digital design concepts, including arithmetic circuits like adders, with insights into timing, optimization, and implementation techniques.
- Weste, N. H. E., & Harris, D. (2010). CMOS VLSI Design: A Circuits and Systems Perspective. Addison-Wesley.This book covers advanced design principles for CMOS-based digital circuits, including strategies for optimizing adders like the Carry Skip Adder.
- Shirakol, S. K., et al. (2014). "Design and Implementation of 16-Bit Carry Skip Adder Using Efficient Adders." Elsevier Publications.Discusses the design of CSKA using efficient adder configurations with emphasis on power, performance, and delay optimizations
- ResearchGate. (n.d.). Reversible Logic for 16-bit Carry Skip Adder Design.Focuses on using reversible logic to improve the efficiency of CSKA, detailing implementation and validation using Xilinx and Cadence tools.
- Cadence Design Systems. (n.d.). Digital Logic Design: Practical Simulation Tools.Official documentation and case studies on implementing adders in the Cadence environment, providing insights into physical design, synthesis, and timing analysis.
- Wikipedia. (n.d.). Carry-Skip Adder.Retrieved from https://en.wikipedia.org/wiki/Carry-skip_adder.A concise explanation of the theory, architecture, and applications of carry skip adders.