MINI PROJECT

Design and Simulation of 4-Bit Carry Save Adder Using Reversible Logic Gates in Cadence

ABSTRACT

In modern digital systems, optimizing arithmetic operations for speed, area, and power efficiency is critical. This paper focuses on the design, simulation, and performance analysis of a 4-bit Carry Save Adder (CSA) using two different approaches: traditional logic gates and reversible logic gates. The goal is to evaluate the potential advantages of reversible logic, which is known for its low power dissipation and emerging importance in quantum computing and low-energy applications. Both CSA designs are implemented and synthesized using Cadence Genus for RTL-level design and analyzed using Cadence Innovus for layout, power, area, and timing metrics. A detailed comparative study is carried out to measure key performance parameters including area utilization, power consumption, and delay. The results provide insights into the trade-offs between conventional and reversible logic designs, highlighting the feasibility of reversible logic in energy-efficient VLSI design.

METHODOLOGY

The design process begins by opening a terminal and using the gedit editor to write the Verilog code for both the 4-bit traditional Carry Save Adder (CSA) and the reversible logic-based CSA, along with their respective testbenches. Once the coding is completed and saved, the INCISIVE simulation environment is launched using the nclaunch command. In the pop-up interface, the "multiple step" option is selected, and a cds.lib file is created to manage design libraries. The Verilog files for the design and testbench are then compiled. The compiled testbench is chosen from work.lib for elaboration, and subsequently selected from the snapshots folder for simulation. The waveform viewer is then opened, the simulation is executed, and the resulting output waveform is observed and captured to verify functionality. After verification, the RTL design is saved into a dedicated RTL folder for synthesis purposes.

Next, synthesis is carried out using Cadence Genus. This is done by navigating to the syn directory and launching the Genus tool via the terminal. Within Genus, a series of commands are executed to perform synthesis, generate the netlist, define timing constraints via an SDC file, and produce reports for power, timing, and area. These reports are saved and analyzed for both the traditional and reversible CSA designs.Following synthesis, the physical implementation phase is carried out using Cadence Innovus. A new project directory (e.g., csa_layout) is created, containing essential files such as Default.globals, the synthesized netlist, .view or .tcl script files, and the block-level SDC file. After closing all Genus-related windows, Innovus is launched from the terminal. Once the Innovus GUI opens, commands are entered to initialize the design using the global settings and netlist. Floorplanning is then configured through the GUI, where values

for core area and utilization are adjusted. Global net connections for power (VDD) and ground (VSS) are created and enforced in the design using the tool's "Connect Global Nets" feature.

Power rings are added around the core boundary using the "Power Planning → Add Rings" option, and a special routing is performed to assign initial metal layers to power and ground nets. After this, standard cell placement is executed, including I/O pin placement, by running the full placement feature. The design is then analyzed for timing by generating a setup-time timing report through the GUI. Area and power reports are generated using the terminal commands report_area and report_power respectively. These results for both the traditional and reversible CSA are then tabulated and compared based on power consumption, timing delay, and area occupied, giving a comprehensive insight into the efficiency of reversible logic in VLSI design.

DESIGN AND IMPLEMENTAION

This project focuses on the design and implementation of two different 4-bit Carry Save Adder (CSA) architectures: one using traditional logic gates and the other using reversible logic gates. The objective is to compare the two approaches based on key parameters such as area, power, and timing using industry-standard EDA tools, Cadence Genus and Innovus.

The traditional 4-bit CSA is designed using conventional full adder modules connected in a carry-save configuration. The Verilog HDL code is written for the design as well as the testbench to validate its functionality. Similarly, for the reversible CSA, a custom design is implemented using reversible logic gates such as Feynman, Fredkin, or Peres gates, ensuring no information loss and low power consumption characteristics. The reversible full adder modules are connected in a similar CSA configuration and verified through simulation.Both designs are functionally verified using the **Cadence Incisive** simulation environment. The Verilog files are compiled, elaborated, and simulated, and the outputs are verified using waveform analysis. Once the simulation results confirm correctness, the designs are saved for synthesis.

The synthesis of both traditional and reversible CSAs is carried out using **Cadence Genus**. The RTL is synthesized to obtain a gate-level netlist along with reports that provide estimates for area utilization, total power consumption, and critical path delay. Synthesis constraints are applied using an SDC (Synopsys Design Constraints) file to ensure realistic timing analysis.Post-synthesis, the design is taken into **Cadence Innovus** for physical implementation. A floorplan is created, and power planning is performed by adding power rings and stripes. Global net connections are established for `VDD` and `VSS`. Special routing and standard cell placement are completed, followed by the generation of timing, power, and area reports from the physical design. These reports help in analyzing the real-world implementation cost of both architectures.

The design and implementation phase concludes with a comprehensive comparison of both designs based on synthesis and physical design metrics. The results clearly demonstrate the impact of reversible logic on area, power, and performance, offering insights into its potential advantages in low-power VLSI design.

TRADITIONAL 4- BIT CARRY SAVE ADDER DESIGN PROGRAM:

```
module CSA(x,y,z,s,cout);
input [3:0] x,y,z;
output [4:0] s;
output cout;
wire [3:0] c1,s1,c2;
fulladder fa0(x[0],y[0],z[0],s1[0],c1[0]);
fulladder fa1(x[1],y[1],z[1],s1[1],c1[1]);
fulladder fa2(x[2],y[2],z[2],s1[2],c1[2]);
fulladder fa3(x[3],y[3],z[3],s1[3],c1[3]);
fulladder fa4(s1[1],c1[0],1'b0,s[1],c2[0]);
fulladder fa5(s1[2],c1[1],c2[0],s[2],c2[1]);
fulladder fa6(s1[3],c1[2],c2[1],s[3],c2[2]);
fulladder fa7(1'b0, c1[3],c2[2],s[4],c2[3]);
assign s[0] = s1[0];
assign cout=c2[3];
endmodule
module fulladder(a,b,cin,sum,carry );
input a,b,cin;
output sum,carry;assign sum = a ^ b ^ cin;
assign carry = (a & b) | (cin & b) | (a & cin);
endmodule
```

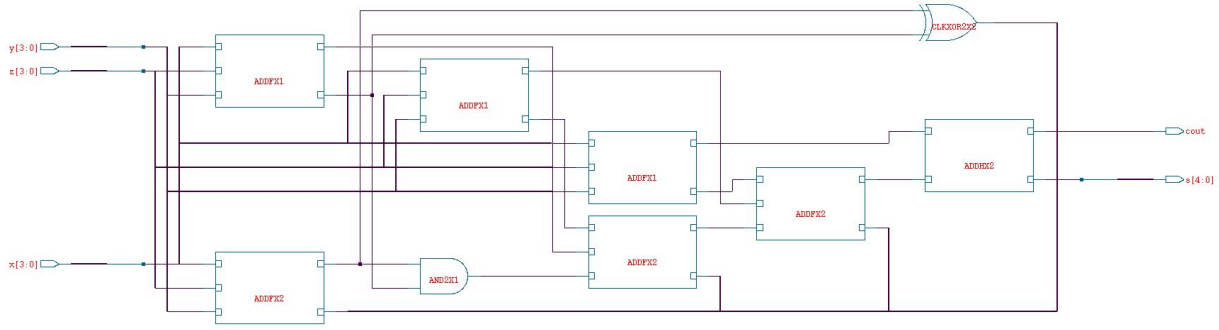
TRADITIONAL 4- BIT CARRY SAVE ADDER TESTBENCH PROGRAM:

```
module CSA_tb();
wire [4:0]s;
wire cout;
reg [3:0]x,y,z;
CSA g1(x,y,z,s,cout);
initial begin

 x=4'b0000;y=4'b0001;z=4'b0010;
#100 x=4'b0001;y=4'b0011;z=4'b0101;
#100 x=4'b1000;y=4'b1011;z=4'b1000;
end
endmodule
```

TRADITIONAL 4- BIT CARRY SAVE ADDER NETLIST:

```
module CSA(x, y, z, s, cout);
input [3:0] x, y, z;
output [4:0] s;
output cout;
wire [3:0] x, y, z;
wire [4:0] s;
wire cout;
wire n_0, n_1, n_2, n_4, n_5, n_6, n_7, n_8;
wire n_10, n_12;
ADDHX2 g480__2398(.A (n_1), .B (n_12), .CO (cout), .S (s[4]));
ADDFX2 g481__5107(.A (n_2), .B (n_4), .CI (n_10), .CO (n_12), .S (s[3]));
ADDFX2 g482__6260(.A (n_5), .B (n_0), .CI (n_8), .CO (n_10), .S (s[2]));
CLKXOR2X2 g483__4319(.A (n_7), .B (n_6), .Y (s[1]));
AND2X1 g484__8428(.A (n_7), .B (n_6), .Y (n_8));
ADDFX1 g487__5526(.A (x[2]), .B (z[2]), .CI (y[2]), .CO (n_4), .S (n_5));
ADDFX2 g486__6783(.A (x[0]), .B (z[0]), .CI (y[0]), .CO (n_7), .S(s[0]));
ADDFX1 g485__3680(.A (x[3]), .B (z[3]), .CI (y[3]), .CO (n_1), .S(n_2));
ADDFX1 g488__1617(.A (x[1]), .B (z[1]), .CI (y[1]), .CO (n_0), .S(n_6));
endmodule
```

WAVEFORM:



REVERSIBLE LOGIC GATES OF 4- BIT CARRY SAVE ADDER DESIGN PROGRAM:

```
module fulladder(input a,b,c,output s,co);
wire p,w1,w2,w3,w4,w5,w6;
nand g1(w1,a,b);
or g2(w2,a,b);
and g3(p,w1,w2);
nand g4(w3,p,c);
or g5 (w4,p,c);
and g6(w5,p,c);
and g7(w6,a,b);
or g8(co,w5,w6);
and g9(s,w3,w4);
endmodule
module CSA(s,cout,x,y,z,);
input [3:0] x,y,z;
output [4:0] s;
output cout;
wire [3:0] c1,s1,c2;
//fulladder
fulladder fa0(x[0],y[0],z[0],s1[0],c1[0]);
fulladder fa1(x[1],y[1],z[1],s1[1],c1[1]);
fulladder fa2(x[2],y[2],z[2],s1[2],c1[2]);
fulladder fa3(x[3],y[3],z[3],s1[3],c1[3]);
fulladder fa4(s1[1],c1[0],1'b0,s[1],c2[0]);
fulladder fa5(s1[2],c1[1],c2[0],s[2],c2[1]);
fulladder fa6(s1[3],c1[2],c2[1],s[3],c2[2]);
fulladder fa7(1'b0, c1[3],c2[2],s[4],c2[3]);
assign s[0] = s1[0];
assign cout=c2[3];
endmodule
```

REVERSIBLE LOGIC GATES OF 4- BIT CARRY SAVE ADDER TESTBENCH PROGRAM:

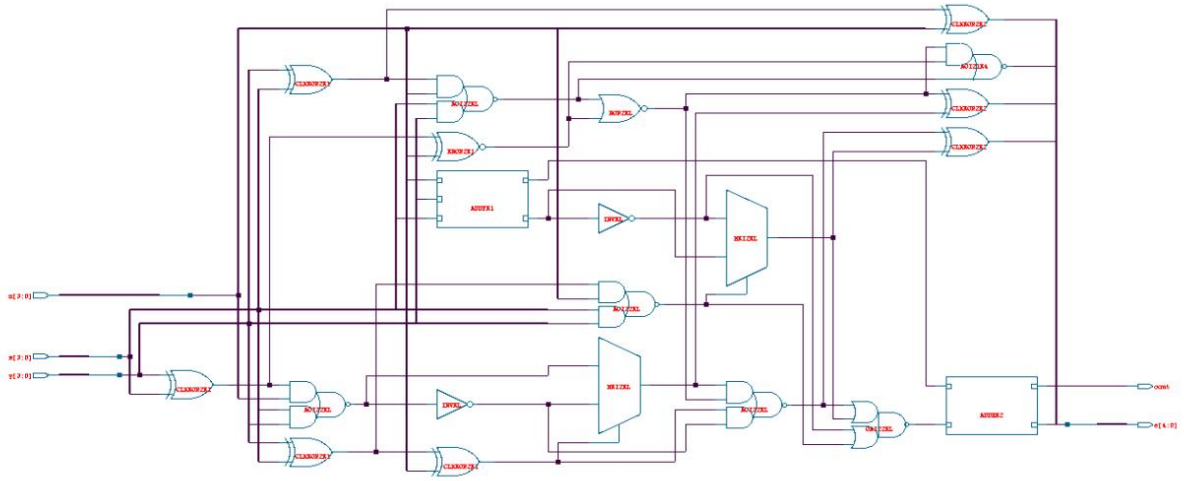
```
module CSA_tb();
wire [4:0]su;wire cout;
reg [3:0]x,y,z;
CSA g1(su,cout,x,y,z);
initial begin
 x=4'b0000;y=4'b0000;z=4'b0000;
#100 x=4'b0001;y=4'b0011;z=4'b0101;
#100 x=4'b1000;y=4'b1011;z=4'b1000;end
endmodule
```

REVERSIBLE LOGIC GATES OF 4- BIT CARRY SAVE ADDER NETLIST:

```
module CSA(s, cout, x, y, z);
input [3:0] x, y, z;
output [4:0] s;
output cout;
wire [3:0] x, y, z;
wire [4:0] s;
wire cout;
wire n_0, n_1, n_4, n_6, n_8, n_9, n_10, n_11;
wire n_13, n_14, n_15, n_16, n_18, n_19, n_21, n_40;
wire n_41;
ADDHX2 g408__2398(.A (n_40), .B (n_21), .CO (cout), .S (s[4]));
OAI22XL g409__5107(.A0 (n_19), .A1 (n_18), .B0 (n_14), .B1 (n_13), .Y(n_21));
CLKXOR2X2 g410__6260(.A (n_19), .B (n_18), .Y (s[3]));
AOI22XL g411__4319(.A0 (n_15), .A1 (n_16), .B0 (n_10), .B1 (n_11), .Y(n_19));
CLKXOR2X2 g412__8428(.A (n_16), .B (n_15), .Y (s[2]));
MXI2XL g414__5526(.A (n_14), .B (n_41), .S0 (n_13), .Y (n_18));
AOI21X4 g413__6783(.A0 (n_9), .A1 (n_8), .B0 (n_16), .Y (s[1]));
MXI2XL g415__3680(.A (n_6), .B (n_11), .S0 (n_10), .Y (n_15));
NOR2XL g416__1617(.A (n_9), .B (n_8), .Y (n_16));
INVXL g422(.A (n_41), .Y (n_14));
INVXL g417(.A (n_6), .Y (n_11));
CLKXOR2X2 g423__2802(.A (n_4), .B (z[0]), .Y (s[0]));
CLKXOR2X1 g424__1705(.A (n_1), .B (z[2]), .Y (n_10));
XNOR2X1 g425__5122(.A (n_0), .B (z[1]), .Y (n_8));
AOI22XL g418__8246(.A0 (n_4), .A1 (z[0]), .B0 (x[0]), .B1 (y[0]), .Y(n_9));
AOI22XL g421__1881(.A0 (n_1), .A1 (z[2]), .B0 (x[2]), .B1 (y[2]), .Y(n_13));
AOI22XL g419__5115(.A0 (n_0), .A1 (z[1]), .B0 (x[1]), .B1 (y[1]), .Y(n_6));
CLKXOR2X1 g428__4733(.A (y[1]), .B (x[1]), .Y (n_0));
CLKXOR2X1 g429__6161(.A (y[2]), .B (x[2]), .Y (n_1));
CLKXOR2X1 g430__9315(.A (y[0]), .B (x[0]), .Y (n_4));
ADDFX1 g2(.A (z[3]), .B (y[3]), .CI (x[3]), .CO (n_40), .S (n_41));
endmodule
```

WAVEFORM:


Baseline = 0
TimeA = 200ns
Name
Cursor
cout
su[4:0]
x[3:0]
y[3:0]
z[3:0]
0
10ns
20ns
30ns
40ns
50ns
60ns
70ns
80ns
90ns
100ns
110ns
120ns
130ns
140ns
150ns
160ns
170ns
180ns
190ns
9
1
8
5



RESULT :

TRADENTIAL 4-BIT CARRY SAVE ADDER ANALYSIS:

AREA REPORT:

```
innovus 3>
innovus 3> report_area
    Hinst Name        Module Name           Inst Count            Total Area
--------------------------------------------------------------------------------
CSA                                                9               152.894
```

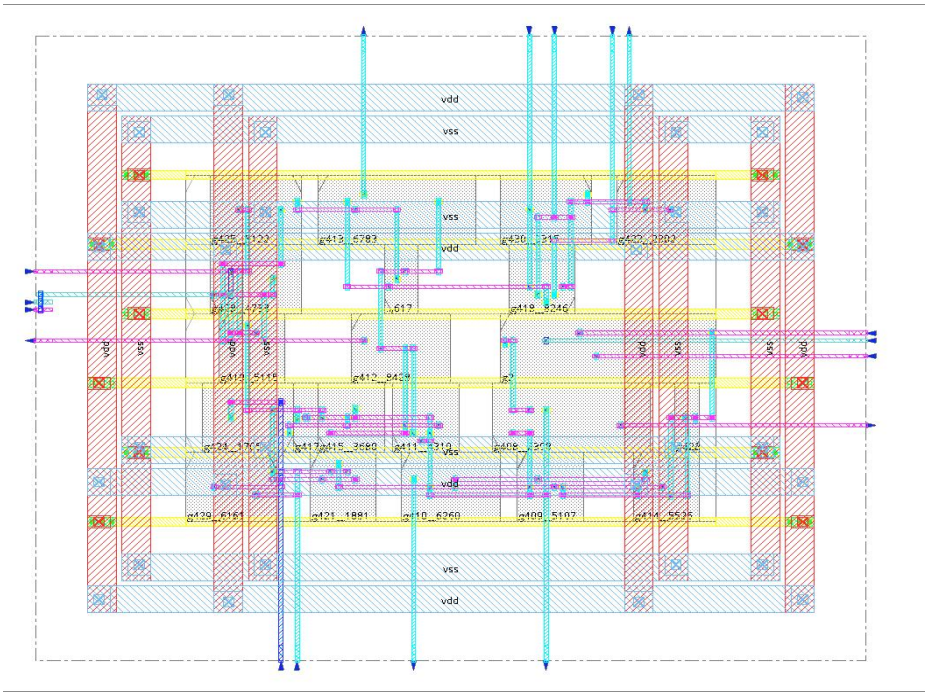
POWER REPORT:

```
Total Power
----------------------------------------------------------------------------------------
Total Internal Power:        0.00550616            20.6708%
Total Switching Power:       0.01977751            74.2470%
Total Leakage Power:         0.00135378             5.0822%
Total Power:                 0.02663745
----------------------------------------------------------------------------------------


Group                           Internal   Switching     Leakage       Total  Percentage
                                Power      Power         Power         Power  (%)
----------------------------------------------------------------------------------------
Sequential                             0           0           0           0           0
Macro                                  0           0           0           0           0
IO                                     0           0           0           0           0
Combinational                   0.005506     0.01978    0.001354     0.02664         100
Clock (Combinational)                  0           0           0           0           0
Clock (Sequential)                     0           0           0           0           0
----------------------------------------------------------------------------------------
Total                           0.005506     0.01978    0.001354     0.02664         100
----------------------------------------------------------------------------------------


Rail                 Voltage   Internal   Switching     Leakage       Total  Percentage
                               Power      Power         Power         Power  (%)
----------------------------------------------------------------------------------------
Default                  1.1   0.005506     0.01978    0.001354     0.02664         100


----------------------------------------------------------------------------------------
*       Power Distribution Summary:
*              Highest Average Power:               g480__2398 (ADDHX2):         0.008332
*              Highest Leakage Power:               g480__2398 (ADDHX2):        0.0002155
*                Total Cap:      9.6604e-13 F
*                Total instances in design:     9
*                Total instances in design with no power:     0
*                Total instances in design with no activty:     0

*                Total Fillers and Decap:     0
----------------------------------------------------------------------------------------

** INFO:  (VOLTUS_POWR-3465): There are 0 decaps and 0 fillers in the design

Ended Static Power Report Generation: (cpu=0:00:00, real=0:00:00,
mem(process/total/peak)=1907.25MB/3686.56MB/1907.25MB)
```

TIMING REPORT:

```
Starting delay calculation for Setup views
AAE_INFO: opIsDesignInPostRouteState() is 0
################################################################################
# Design Stage: PreRoute
# Design Name: CSA
# Design Mode: 90nm
# Analysis Mode: MMMC Non-OCV
# Parasitics Mode: No SPEF/RCDB
# Signoff Settings: SI Off
################################################################################
Calculate delays in BcWc mode...
Start delay calculation (fullDC) (1 T). (MEM=2068.07)
Total number of fetched objects 28
AAE_INFO: Total number of nets for which stage creation was skipped for all views 0
End delay calculation. (MEM=2204.17 CPU=0:00:00.0 REAL=0:00:00.0)
End delay calculation (fullDC). (MEM=2204.17 CPU=0:00:00.0 REAL=0:00:00.0)
*** Done Building Timing Graph (cpu=0:00:00.1 real=0:00:00.0 totSessionCpu=0:00:22.4 mem=2196.2M)

------------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------------

Setup views included:
 best

+--------------------+--------+---------+
|     Setup mode     |   all  | default |
+--------------------+--------+---------+
|           WNS (ns):|  0.000 |  0.000  |
|           TNS (ns):|  0.000 |  0.000  |
|    Violating Paths:|    0   |    0    |
|          All Paths:|    0   |    0    |
+--------------------+--------+---------+

+----------------+-----------------------------+------------------+
|                |             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: 68.707%
Routing Overflow: 0.00% H and 0.00% V
------------------------------------------------------------------
Reported timing to dir timingReports
Total CPU time: 0.47 sec
Total Real time: 1.0 sec
Total Memory Usage: 2173.585938 Mbytes
*** timeDesign #1 [finish] : cpu/real = 0:00:00.2/0:00:00.7 (0.2), totSession cpu/real = 0:00:22.4/0:05:00.8 (0.1), mem = 2173.6M
```

LAYOUT:



REVERSIBLE LOGIC GATES OF 4-BIT CARRY SAVE ADDER ANALYSIS:

AREA REPORT:

```
innovus 3> report_area
    Hinst Name       Module Name          Inst Count           Total Area
------------------------------------------------------------------------
CSA                                            21              167.275
```

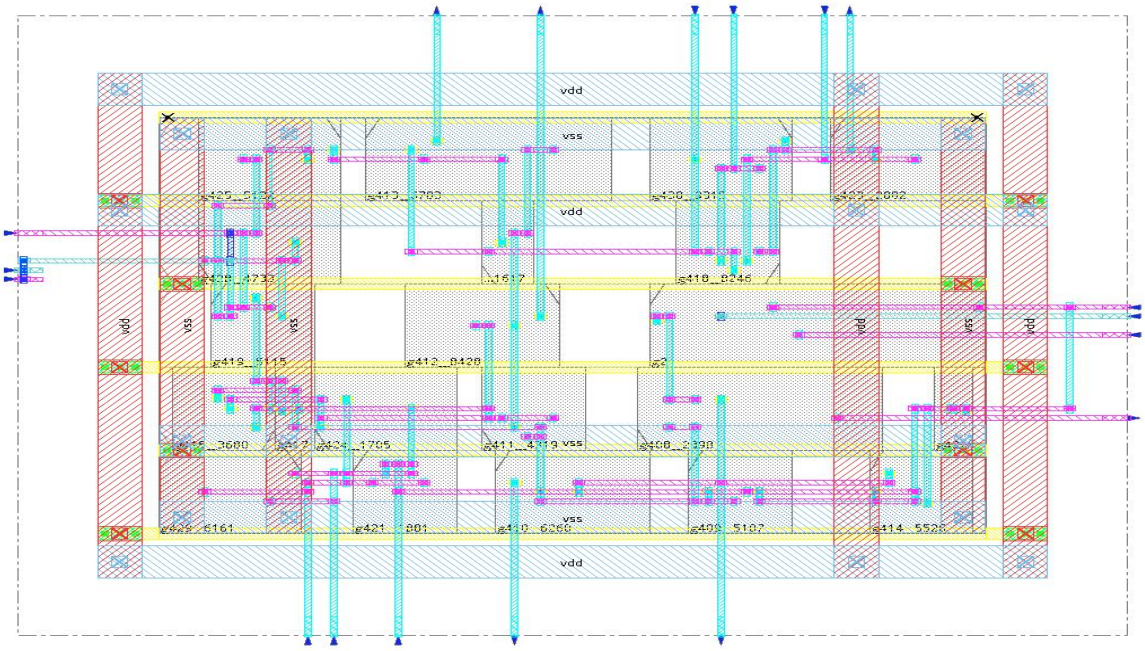
POWER REPORT:

```
Total Power
--------------------------------------------------------------------------------------------
Total Internal Power:        0.00769172            21.1402%
Total Switching Power:       0.02665210            73.2517%
Total Leakage Power:         0.00204044             5.6080%
Total Power:                 0.03638426
--------------------------------------------------------------------------------------------


Group                            Internal   Switching     Leakage       Total  Percentage
                                 Power      Power         Power         Power  (%)
--------------------------------------------------------------------------------------------
Sequential                              0           0           0           0           0
Macro                                   0           0           0           0           0
IO                                      0           0           0           0           0
Combinational                    0.007692     0.02665     0.00204     0.03638         100
Clock (Combinational)                   0           0           0           0           0
Clock (Sequential)                      0           0           0           0           0
--------------------------------------------------------------------------------------------
Total                            0.007692     0.02665     0.00204     0.03638         100
--------------------------------------------------------------------------------------------


Rail                  Voltage    Internal   Switching     Leakage       Total  Percentage
                                 Power      Power         Power         Power  (%)
--------------------------------------------------------------------------------------------
Default                   1.1    0.007692     0.02665     0.00204     0.03638         100


--------------------------------------------------------------------------------------------
*       Power Distribution Summary:
*              Highest Average Power:                g408__2398 (ADDHX2):          0.009931
*              Highest Leakage Power:                g408__2398 (ADDHX2):         0.0002155
*                Total Cap:      1.01612e-12 F
*                Total instances in design:    21
*                Total instances in design with no power:     0
*                Total instances in design with no activty:     0

*                Total Fillers and Decap:     0
--------------------------------------------------------------------------------------------

** INFO:  (VOLTUS_POWR-3465): There are 0 decaps and 0 fillers in the design

Ended Static Power Report Generation: (cpu=0:00:00, real=0:00:00,
mem(process/total/peak)=1662.25MB/3279.73MB/1662.25MB)
```

TIMING REPORT:

```
# Design Mode: 90nm
# Analysis Mode: MMMC Non-OCV
# Parasitics Mode: No SPEF/RCDB
# Signoff Settings: SI Off
################################################################################
Calculate delays in BcWc mode...
Start delay calculation (fullDC) (1 T). (MEM=1658.19)
Total number of fetched objects 35
AAE_INFO: Total number of nets for which stage creation was skipped for all views 0
End delay calculation. (MEM=1793.27 CPU=0:00:00.0 REAL=0:00:00.0)
End delay calculation (fullDC). (MEM=1793.27 CPU=0:00:00.0 REAL=0:00:00.0)
*** Done Building Timing Graph (cpu=0:00:00.1 real=0:00:00.0 totSessionCpu=0:00:28.6 mem=1785.3M)

------------------------------------------------------------------
          timeDesign Summary
------------------------------------------------------------------

Setup views included:
 best

+--------------------+---------+---------+
|     Setup mode     |   all   | default |
+--------------------+---------+---------+
|           WNS (ns):|  0.000  |  0.000  |
|           TNS (ns):|  0.000  |  0.000  |
|    Violating Paths:|    0    |    0    |
|          All Paths:|    0    |    0    |
+--------------------+---------+---------+

+----------------+-------------------------------+------------------+
|                |              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: 69.062%
Routing Overflow: 0.00% H and 0.00% V
------------------------------------------------------------------
Reported timing to dir timingReports
Total CPU time: 0.46 sec
Total Real time: 1.0 sec
Total Memory Usage: 1762.691406 Mbytes
*** timeDesign #1 [finish] : cpu/real = 0:00:00.1/0:00:00.7 (0.2), totSession cpu/real = 0:00:28.6/0:07:07.2 (0.1), mem = 1762.7M
```

LAYOUT:



DISCUSSION:

Comparing the two implementations of the 4-bit carry-save adder, we observe differences in their area and power characteristics, while both designs successfully meet the timing requirements under the analyzed best-case scenario. The reversible gate-based implementation exhibits a larger area footprint, occupying 167.275 units compared to the original design's 152.094 units. This increase in area is a common trade-off when employing reversible logic, which often necessitates additional gates to ensure the one-to-one mapping between inputs and outputs.

In terms of power consumption, the reversible gate-based adder shows a slightly higher total power dissipation of approximately 0.0364 mW compared to the original design's 0.0266 mW. This increase is primarily attributed to a higher switching power component in the reversible gate implementation (0.0261 mW vs. 0.0198 mW). The leakage power is also marginally higher in the reversible design (0.0027 mW vs. 0.0014 mW).

Despite these differences in area and power, both implementations demonstrate satisfactory timing performance, with no setup or hold time violations reported in the best-case analysis. This indicates that both designs can operate at the required speed under these conditions. The layout density and routing overflow are also comparable between the two designs, suggesting similar levels of physical implementation efficiency. The increased component count in the reversible gate design, likely contributing to the larger area, doesn't appear to have significantly impacted the routing complexity.

CONCLUSION

In conclusion, while both the original and the reversible gate-based 4-bit carry-save adders meet the timing requirements, the reversible gate implementation comes with a penalty in terms of area and power consumption. The reversible design occupies a larger area and dissipates more power, particularly switching power, compared to the original implementation. This trade-off between reversibility and increased resource utilization is a crucial consideration in circuit design, especially when area and power efficiency are critical constraints. The choice between the two implementations would ultimately depend on the specific application requirements and the relative importance of reversibility versus area and power optimization. If the benefits of reversible computing, such as potential for ultra-low power in specific paradigms, are paramount, the increased area and power might be acceptable. However, for applications where area and power are strictly limited, the original, non-reversible design appears to be more efficient..

REFERENCE

1. C. Pakkiraiah and R. V. S. Satyanarayana, "Design and FPGA realization of energy efficient reversible full adder for digital computing applications," *J. VLSI Circuits Syst.*, vol. 6, no. 1, pp. 7–18, 2024.

2. C. Pakkiraiah and D. R. Satyanarayana, "An innovative design of low power binary adder based on switching activity," *Int. J. Comput. Digit. Syst.*, vol. 11, no. 1, pp. 861–871, 2022.

3. D. Krishnaveni, N. Kiran, H. Shalini, and M. Geetha Priya, "Reversible FADE gate as decoder, encoder and full adder," in *Advances in Automation, Signal Processing, Instrumentation, and Control*, Springer, pp. 1323–1331, 2021.

4. **S**. Ahmed, M. I. Baba, S. M. Bhat, I. Manzoor, N. Nafees, and S.-B. Ko, "Design of reversible universal and multifunctional gate-based 1-bit full adder and full subtractor in quantum-dot cellular automata nanocomputing," *J. Nanophoton.*, vol. 14, no. 3, p. 036002, 2020.

5. S. Raghuraman, *Efficiency of logic minimization techniques for cryptographic hardware implementation*, Ph.D. dissertation, Virginia Tech, 2019.