

Design and Verification of a Pipelined Synchronous 8-Bit Carry Select Adder

Aswin Natesh Venkatesh^a, Aishwarya Gandhi^b
Department of Electrical and Computer Engineering^{ab}
Stony Brook University^{ab}, Stony Brook, NY, USA
[{aswinnatesh.venkatesh,aishwarya.gandhi}^b](mailto:{aswinnatesh.venkatesh,aishwarya.gandhi}@stonybrook.edu)@stonybrook.edu

Abstract— This project presents the design, layout and performance evaluation of pipelined synchronous 8-bit carry select adder. The adder is designed and implemented using 45nm CMOS process technology. The carry-select adder is implemented using 4-bit ripple carry adders and an array of 2→1 multiplexer's used to select the carry. The design is pipelined at the primary inputs and outputs to reach higher clock rates. The layout design is efficiently optimized in terms of area following CMOS 45nm technology micron rules. The design is validated for functionality, various performance parameters and power. The maximum design objective reached is 7.70×10^{16} Hz/Wm² at a frequency of 3.5GHz. The total layout area is found to be 758.60μm² and consumes 1.650μW power.

Index Terms— 8-Bit, Carry Select Adder, Pipelined Adder, Cadence Virtuoso, 45nm Technology.

I. INTRODUCTION

Adders are one of the most widely used digital component in the integrated circuit design and are the highly necessary part of Digital Signal Processing (DSP) applications. With the advances in technology, researchers have tried and are trying to design adders which offer either high speed, low power consumption, less area or the combination of them. Ripple Carry Adder (RCA), Carry Skip Adder (CSkA), Carry Increment Adder (CIA), Carry Look Ahead Adder (CLA), Carry Save Adder (CSA), Carry Select Adder (CSIA) and Carry Bypass Adder (CBA) are some of the different adder topologies. Every adder is named based on the propagation of carry between the stages. Few of the existing adder topologies are discussed below.

Ripple Carry Adder(RCA) contains series structure of Full Adders (FA); each FA is used to add two bits along with carry bit. The carry generated from each full adder is given to next full adder and so on. Hence, the carry is propagated in a serial computation. Delay is more as the number of bits is increased.

Carry Skip Adder(CSkA) uses skip logic in the propagation of carry. It is designed to speed up the addition operation by adding a propagation of carry bit around a portion of entire adder. It shortens the critical path by computing the group propagate signals for each carry chain and using this to skip over long carry ripples.

The design of Carry Increment Adder (CIA) consists of RCA's and incremental circuitry. The incremental circuit can be designed using HA's in ripple carry chain with a sequential order. The addition operation is done by dividing total number of bits in to group of 4bits and addition operation is done using several 4-bit RCA's.

The carry-look ahead adder (CLA) is similar to the carry-skip adder, but computes group generated signals as well as group propagate signals to avoid waiting for a ripple to determine if the first group generates a carry. In the generation and propagation stage, the generation values and propagation values are computed. Internal carry generation is calculated in second stage. And in final stage, the sum is calculated. This adder reduces the carry delay by reducing the number of gates through which a carry signal must propagate.

In Carry Save Adder (CSA), three bits are added parallelly at a time. In this scheme, the carry is not propagated through the stages. Instead, carry is stored in present stage, and updated as addend value in the next stage. Hence, the delay due to the carry is reduced in this scheme.

The Carry Select Adder (CSIA) architecture consists of independent generation of sum and carry i.e., Cin=1 and Cin=0 are executed parallelly. Depending upon Cin, the external multiplexers select the carry to be propagated to next stage. Further, based on the carry input, the sum will be selected. Hence, the delay is reduced. However, the structure is increased due to the complexity of multiplexers.

In Carry Bypass Adder (CBA), RCA is used to add 4-bits at a time and the carry generated will be propagated to next stage with help of multiplexer using select input as Bypass logic. By pass logic is formed from the product values as it is calculated in the CLA. Depending on the carry value and by pass logic, the carry is propagated to the next stage.

II. CARRY SELECT ADDER DESIGN

The 8- bit Carry Select Adder is designed and implemented using several 1-bit full adders, flip-flops and 2→1 Multiplexors building blocks. The design of each of these building blocks are discussed in detail in the subsections below.

A. Block Level Architecture

The 8-bit carry select adder is split into two 4-bit groups. The lower order bits (a3, a2, a1, a0) and (b3, b2, b1, b0) are fed to the first 4-bit adder to produce the sum bits (s3, s2, s1, s0) and a carry-out bit (c4). The higher order bits (a7, a6, a5, a4 & b7 b6, b5, b4) are fed as inputs to the second and third 4-bit adder. The second Adder calculates the sum with a carry-in of c=0, while the third adder calculates with a carry-in of c=1. The results of the two adders (second and third) are fed as inputs to an array of 2:1 multiplexor's. The carry bit from the c4 of first adder is used as the select signal for the multiplexors. When c4 is low, then the results of second adder are passed to the output,

while when c_4 is high, the results of third adder are passed as outputs. The carry-out bit c_8 is also selected by the MUX array.

In order to increase the operational frequency of the adder, the design has been pipelined by placing flip flops at the primary inputs and outputs. The Critical path of the design is from the Cin of the first adder to the Cout of the final result. The block level architecture of the adder is shown in Figure 1.

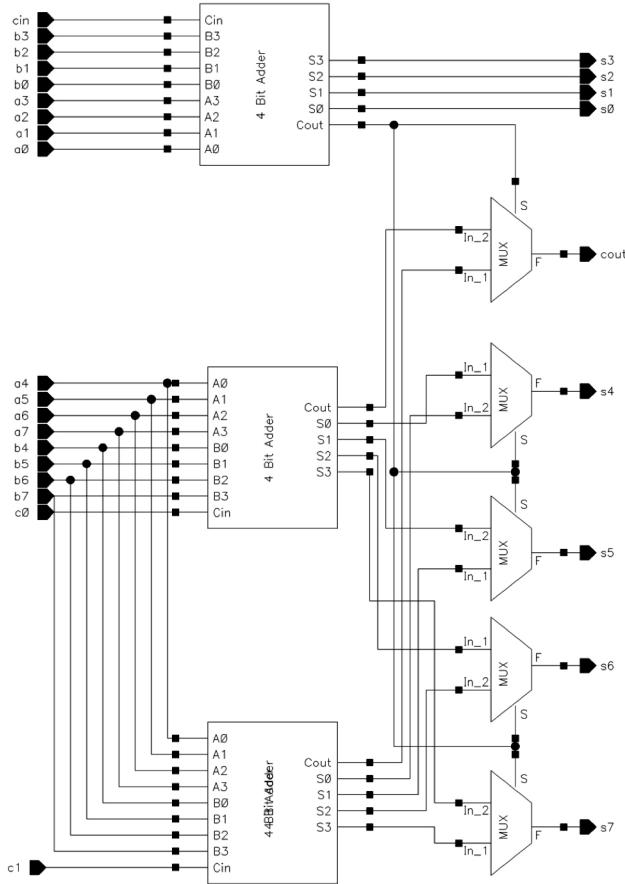


Fig. 1. Block Level Architecture

B. Flip-Flops

In order to increase the operational frequency of the adder, the design is pipelined using flip-flops at the primary inputs and outputs. A CMOS positive edge triggered master-slave D type flip-flop with asynchronous reset is implemented in the design. The D flip-flop is implemented using CMOS inverters, transmission gates, NAND gates, tri-state NAND and tri-state Inverters.

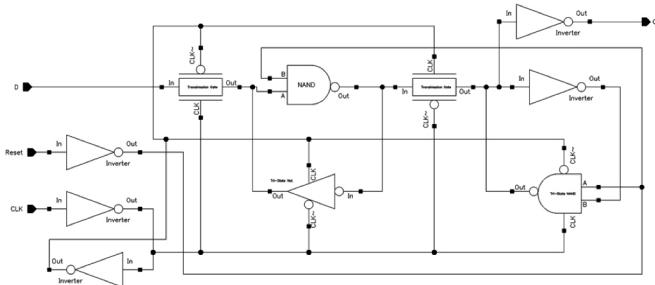


Fig. 2. D Flip Flop Schematic

At the first stage, the master, consists of a NAND gate for Reset and the tri-state Inverter is connected in a positive feedback configuration. At the second stage, the slave, consists of an inverter which gives the output and which is passed to tri-state NAND for feedback signal. When the clock goes low, the D input is transmitted to the first stage, and the second stage is connected in positive feedback to ensure that the output still does not change. When the clock goes high again, the second stage is set to the same state as the first, changing the output. The output changes only when the clock has a transits from low to high. The Schematic diagram of D Flip Flop is shown in Figure 2.

C. 1-Bit Adder

The final adder comprises of several small 1-bit adder building blocks. Various design methodologies like static CMOS, CPL, TG or domino logic can be used for full adder design. Static CMOS seems to be most promising among all other architectures with respect to speed, power dissipation and power delay product. Domino logic designs are good for high speed and not for low power applications. In this design Static CMOS logic has been implemented for the 1-bit adder consisting of 28 transistors. It is a mirror adder as the pMOS network is identical to the nMOS network rather than being the conduction complement. The design is made compact by reusing the Cout term to compute S and is represented by equation (1) & (2).

$$S = ABC_{in} + (A + B + C)\bar{C}_{out} \quad (1)$$

$$C_{out} = AB + BC_{in} + AC_{in} \quad (2)$$

A and B are two single-bit input signals, C_{in} is a single-bit input carry signal and S and C_{out} are two single-bit output signals. The Schematic diagram of 1-Bit Adder is shown in Figure 3.

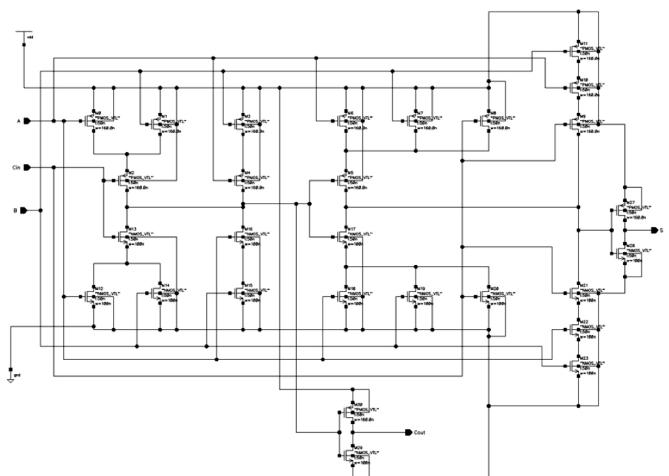


Fig. 3. 1-Bit Adder Schematic

D. 4-Bit Adder

The 4-bit adder block used in CSA is a ripple carry adder. In ripple carry adder each carry bit from a full adder "ripples" to the next full adder. Cin is the input carry (a_3, a_2, a_1, a_0 & b_3, b_2, b_1, b_0) represents two 4-bit input binary numbers. Thus, Cout of the first adder goes to Cin of the second adder and Cout of the second adder goes to Cin of the third adder and so on. C_4

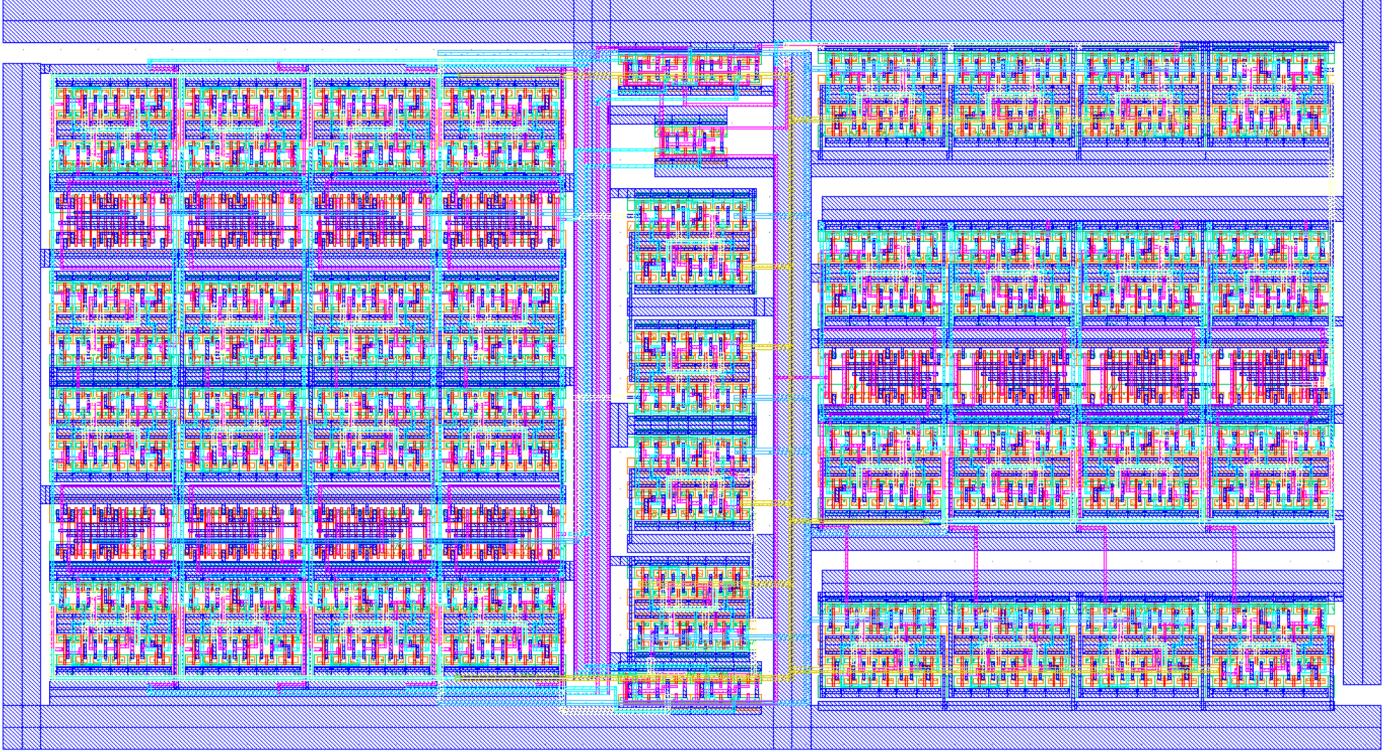


Fig. 6. Layout of 8-Bit Carry Select Adder Schematic

Is the output carry and (s_3, s_2, s_1, s_0) is the output sum. The schematic diagram of the 4-bit adder is shown in Figure 4.

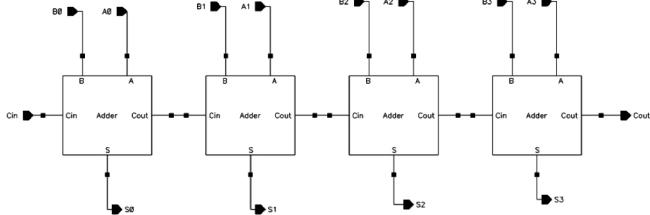


Fig. 4. 4-Bit Ripple Carry Adder Schematic

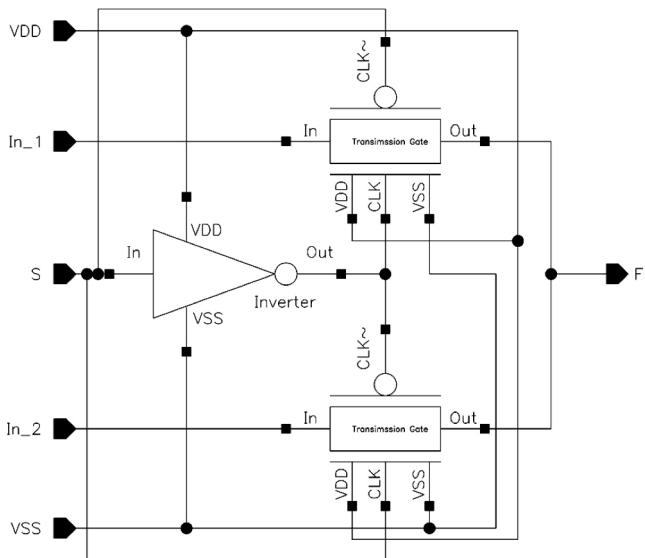


Fig. 5. $2 \rightarrow 1$ Multiplexer Schematic Diagram

E. $2 \rightarrow 1$ Multiplexer

A Transmission gate based multiplexer is implemented in the adder design for the following functionality. The higher order sum bits (s_4, s_5, s_6, s_7) are selected from the second or third adder based on the select signal from the first adder computing the first order bits. To perform this operation a Transmission gate (TG) based multiplexor is implemented considering its low area requirement. Two transmission gates are used to select between two inputs based on the select signals S. In this design, when S is high, output of third adder is passed, while when S is low, the output of second adder is passed. The schematic diagram of the $2 \rightarrow 1$ Multiplexer Schematic Diagram is shown in Figure 4.

III. LAYOUT DESIGN AND SIMULATION

A. Layout Design

The 8-Bit Carry Select Adder is implemented using CMOS 45nm technology. Figure 6 shows the layout of 8-bit adder. The rectangular dimensions of the layout are $36.96\mu\text{m} \times 20.525\mu\text{m}$, with total area of $758.60 \mu\text{m}^2$. The floor plan of the layout consists of 12 1-bit-adder, 12 D-Flip-Flop and 5 Multiplexers. All of these blocks are designed individually and integrated to form the schematic of the complete 8-Bit Adder. The layout is efficiently designed in terms of area by keeping minimum distance between different layers according to the micron rules specified for 45nm CMOS process technology.

B. Post-Layout Simulation

The 8-Bit Carry Select Adder is simulated with the supply voltage of 1.1V. Simulations were carried out till 4 GHz of operation and found to be working. Waveforms are attached for 3.5 GHz of operations. The test specifications of the 4-bit

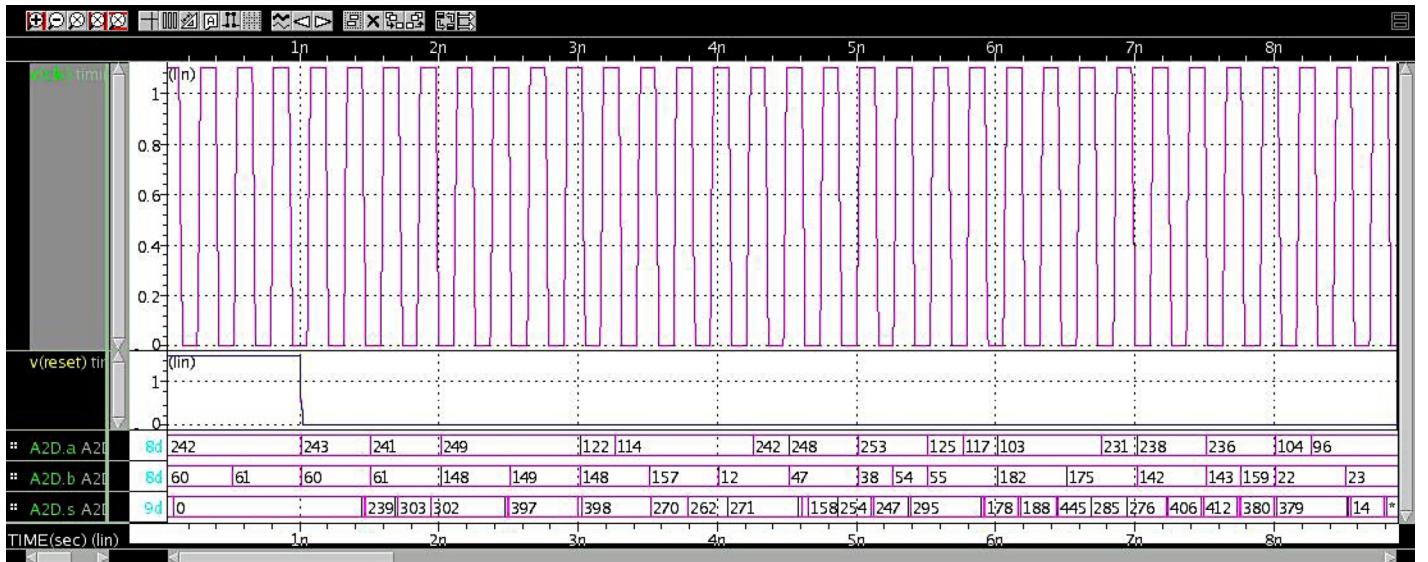


Fig. 7. 8-Bit Adder Simulation Waveform

Adder is tabulated in Table 1. Waveforms showing functional verification of 8-Bit Adder is shown in Figure 7.

TABLE 1: TEST SPECIFICATIONS

Load Capacitance (s0....s7, Cout)	5 fF
Clock Frequency	4 GHz
Duty Cycle	50%
Clock Rise Time	25 ps
Clock Fall Time	25 ps
Input Rise Time	25 ps
Input Fall Time	25 ps

IV. RESULTS AND DISCUSSION

The design objective of this project is to maximize the result of the expression given in equation (3).

$$\text{Design Objective} = \frac{f_{\text{clock}}}{P \times A^{0.5}} \left(\frac{\text{Hz}}{\text{Wm}^2} \right) \quad (3)$$

where f_{clock} is the clock frequency, P is average power consumption, and A is the layout area. The Power and Area for 3 different clock frequencies were observed and the design objective was calculated. The results are tabulated in Table 2. It is observed that the average power utilization is 1.892mW at 4GHz frequency which is the maximum operational frequency of the design to perform correct functionality.

TABLE 2: DESIGN PERFORMANCE

Frequency (Hz)	Area (μm^2)	Power (mW)	Design Objective (Hz / Wm^2)
2.0	758.60	1.111	6.53×10^{16}
2.5	758.60	1.298	6.99×10^{16}
3.0	758.60	1.518	7.15×10^{16}
3.5	758.60	1.650	7.70×10^{16}
4.0	758.60	1.892	7.78×10^{16}

The design is evaluated till 4 GHz clock speed and exceeding this rate the number of glitches in the waveform is found to increase. It can be observed that Power utilization from design performance table, doubles 1.11 mW to 1.65 mW with increase in frequency from 2 GHz to 3.5 GHz.

V. CONCLUSION

In this project a pipelined synchronous 8-bit carry select adder was designed and verified in 45nm CMOS process technology. The design is validated for functionality, performance and power. The design reaches a maximum design objective of $7.70 \times 10^{16} \text{ Hz/Wm}^2$ at 3.5GHz operating frequency. The layout design of the 8-bit adder is efficiently optimized in terms of area, which is $758.60 \mu\text{m}^2$. The average power utilization is 1.650mW.

APPENDIX

Successful Design Rule Check (DRC) and Layer Vs Schematic (LVS) are attached with this report.

REFERENCES

- [1] Neil H. E. Weste, Kamran Eshraghian, " Principles of CMOS VLSI design", A systems perspective, 4th edition.
- [2] Jacob R Baker, "CMOS Circuit, Design, Layout and Simulation" IEEE Series on Microelectronic Systems, 3rd edition.
- [3] HSPICE User Guide, Synopsys, Version B-2008.09

Appendix

1) Successful DRC Result of 8-Bit Pipelined Carry Select Adder

The figure shows two side-by-side windows of the Calibre DRC tool.

Left Window: Calibre - RVE v2014.4_28.20 : adderfinal.drc.results

- Menu Bar:** File, View, Highlight, Tools, Window, Setup, Help.
- Toolbar:** Includes icons for search, zoom, and file operations.
- Status Bar:** Rule File Pathname: /home/home5/avenkatesh/asviness555/_calibreDRC.rul_. Nwell and Pwell must not overlap.
- Table:** Shows a list of checks with results. All entries have a green checkmark and a result of 0.

Check / Cell	Results
Check Well.1	0
Check Well.2	0
Check Well.4	0
Check Poly.1	0
Check Poly.2	0
Check Poly.3	0
Check Poly.4	0
Check Poly.5	0
Check Poly.6	0
Check Active.1	0
Check Active.2	0
Check Active.3	0
Check Active.4	0
Check Implant.1	0
Check Implant.2	0
Check Implant.3	0
Check Implant.4	0
Check Implant.6	0
Check Contact1	0
Check Contact2	0
Check Contact3	0
Check Contact4	0
Check Contact5	0
Check Contact6	0
Check Metal1.1	0
Check Metal1.2	0
Check Metal1.3	0
Check Metal1.4	0

Right Window: DRC Summary Report - adderfinal.drc.summary

```

=====
*** CALIBRE - DRC-H SUMMARY REPORT
=====
Execution Date/Time: Thu Dec 7 23:56:17 2017
Calibre Version: v2014_4_28_20
Thu Dec 4 12:46:29 PST 2014
Rule File Pathname: /home/home5/avenkatesh/asviness555/_calibreDRC.rul_
Rule File Title:
Layout System: GDS
Layout Path(s): adderfinal.calibre.db
Primary Library Cell: adderfinal
Current Directory: /home/home5/avenkatesh/asviness555
User Name: avenkatesh
Maximum Results/RuleCheck: 1000
Max Result Value: 1000
Max Result Database: adderfinal.drc.results (ASCII)
Layout Depth: ALL
Text Depth: PRIMARY
Summary Report File: adderfinal.drc.summary (REPLACE)
Geometry Flagging: ACUTE = NO SKEW = NO ANGLED = NO OFFGRID = NO
NONSIMPLE POLYGON = NO NONSIMPLE PATH = NO
Excluded Cells:
CheckText Mapping: COMMENT TEXT + RULE FILE INFORMATION
Layers: MEMORY-BASED
Keep Empty Checks: YES
--- RUNTIME WARNINGS
---
--- ORIGINAL LAYER STATISTICS
---
LAYER pwell ..... TOTAL Original Geometry Count = 19 (1206)
LAYER nwell ..... TOTAL Original Geometry Count = 19 (1206)
LAYER active ..... TOTAL Original Geometry Count = 24 (1860)
LAYER via1 ..... TOTAL Original Geometry Count = 12 (930)
LAYER implant ..... TOTAL Original Geometry Count = 12 (930)
LAYER nimplant ..... TOTAL Original Geometry Count = 12 (930)
LAYER vth ..... TOTAL Original Geometry Count = 0 (0)
LAYER via2 ..... TOTAL Original Geometry Count = 0 (0)
LAYER metal1 ..... TOTAL Original Geometry Count = 270 (11500)
LAYER contact ..... TOTAL Original Geometry Count = 58 (3931)
LAYER metal2 ..... TOTAL Original Geometry Count = 155 (4041)
LAYER metal3 ..... TOTAL Original Geometry Count = 118 (2099)
LAYER metal4 ..... TOTAL Original Geometry Count = 84 (959)
LAYER metal5 ..... TOTAL Original Geometry Count = 23 (80)
LAYER metal6 ..... TOTAL Original Geometry Count = 28 (42)
LAYER metal7 ..... TOTAL Original Geometry Count = 0 (0)
LAYER metal8 ..... TOTAL Original Geometry Count = 0 (0)
LAYER metal9 ..... TOTAL Original Geometry Count = 0 (0)
LAYER metal10 ..... TOTAL Original Geometry Count = 0 (0)
LAYER via3 ..... TOTAL Original Geometry Count = 1 (1418)
LAYER via4 ..... TOTAL Original Geometry Count = 2 (652)
LAYER via5 ..... TOTAL Original Geometry Count = 1 (233)
LAYER via6 ..... TOTAL Original Geometry Count = 1 (34)
LAYER via7 ..... TOTAL Original Geometry Count = 1 (15)
LAYER via8 ..... TOTAL Original Geometry Count = 0 (0)
LAYER via9 ..... TOTAL Original Geometry Count = 0 (0)
LAYER via10 ..... TOTAL Original Geometry Count = 0 (0)

```

2) Successful LVS Result of 8-Bit Pipelined Carry Select Adder

The screenshot shows the Calibre RVE interface with two main windows open:

- LVS Report File - adderfinal.lvs.report**: This window displays the LVS report file content. It includes header information like REPORT FILE NAME, LAYOUT NAME, SOURCE NAME, and CALIBRE VERSION. Below this is the OVERALL COMPARISON RESULTS section, which shows a summary of layout vs source differences.
- Comparison Results**: This window shows a table comparing Layout Cell / Type, Source Cell, Nets, Instances, and Ports for the adderfinal cell. The table indicates 441L, 441S, 536L, 536S, 32L, and 32S respectively.

Left Sidebar (Navigator):

- Results: Extraction Results, Comparison Results
- Reports: Extraction Report, LVS Report
- Rules
- View: Info, Finder, Schematics
- Setup: Options