

Schematic and Layout Design of 16-bit Carry-Select Adder

Richard Rhee
rheer@purdue.edu

I. INTRODUCTION

Addition is the building block of arithmetic operations used ubiquitous in all areas of electronics, therefore it is extremely important that its performance is optimized as much as possible. The carry-select-adder (CSA) offers much improved propagation delay over the ripple-carry-adder (RCA) which suffers from large propagation delay of having to wait for the carry bit to ripple through the entire device.

II. COMPLEMENTARY METAL-OXIDE-SEMICONDUCTOR (CMOS) INVERTER

A. Optimal Width Sizing

To optimize propagation delay and energy consumption of the Carry-Select Adder (CSA), it is imperative these metrics are optimized starting with the smallest CMOS device, the inverter. Electrons and holes present in metal-oxide-semiconductor field-effect-transistors (MOSFETs) have different mobilities, which need to be accounted and optimized for in P-channel (PMOS) and N-channel MOSFETs (NMOS). Running a parametric analysis of the CMOS inverter with fixed PMOS and NMOS length at 45 nm, and NMOS width at 260 nm, the sizing ratio of PMOS-to-NMOS width of 1.76 is found [Fig. 1].

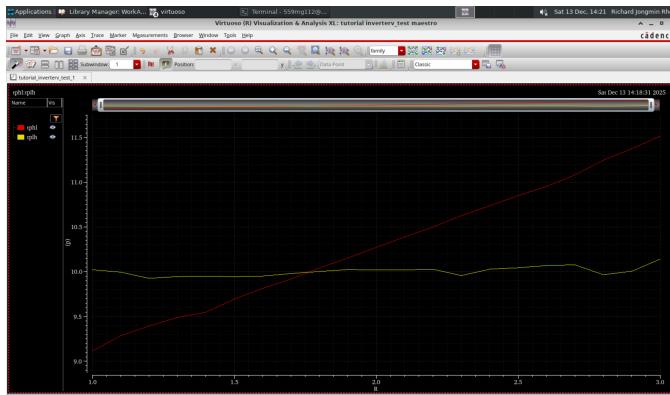


Fig. 1. Parametric analysis of PMOS-to-NMOS width ratios

Using this ratio, the PMOS width is set to $1.76 * 260 = 460$ nm, and NMOS to 260 nm. The delay and energy measurements of this inverter can be found in [Fig. 2].

III. NOT-AND (NAND) GATE

The NAND gate is built in the CMOS architecture using 2 PMOS and 2 NMOS transistors. To optimize the NAND

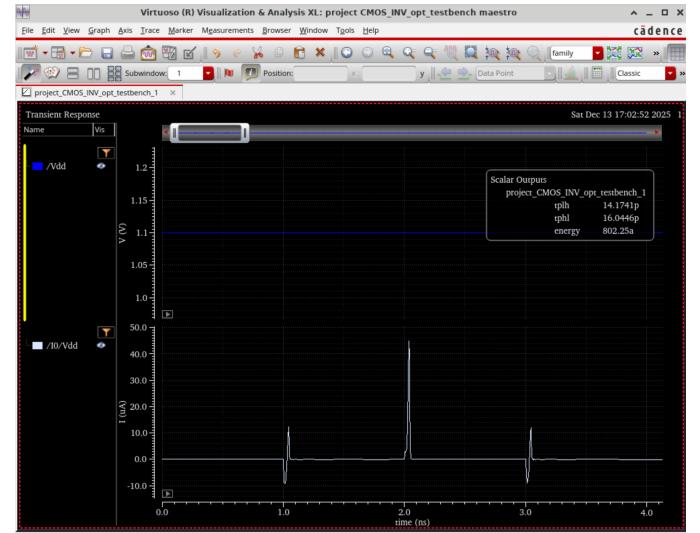


Fig. 2. Propagation delay and energy consumption of CMOS inverter

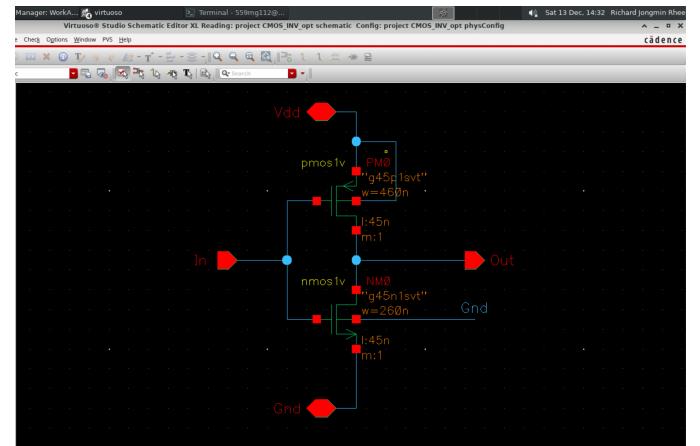


Fig. 3. Schematic of CMOS inverter

gate for worst-case propagation delay, the reference inverter is used with the transistor sizing method for PMOS width of $1.76 * 260 = 460$ nm, and NMOS width of $2 * 260 = 520$ nm [Fig. 5].

IV. EXCLUSIVE-OR (XOR) GATE

The XOR gate is built in the CMOS architecture, incorporating the previously discussed inverter to produce the complementary input signals. Using the worst-case transistor

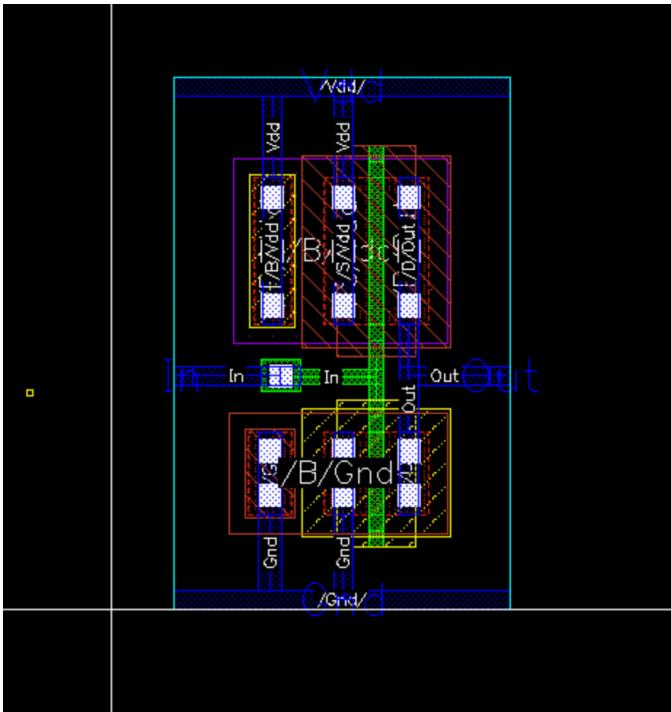


Fig. 4. Layout of CMOS inverter

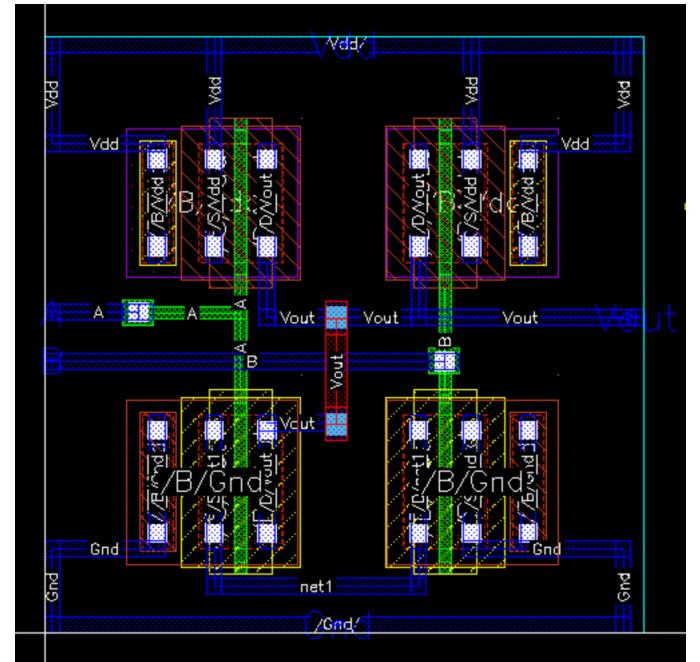


Fig. 6. Layout of CMOS NAND gate

sizing method, the width of the PMOS is set to 460 nm, and the NMOS to 260 nm [Fig. 7].

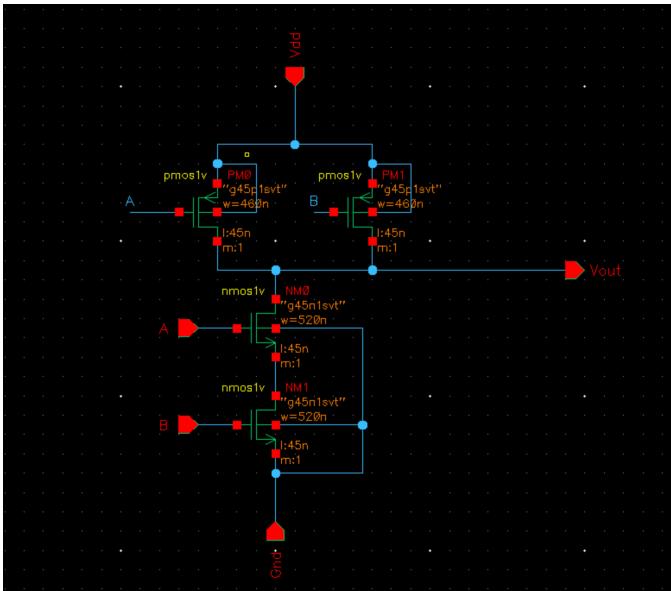


Fig. 5. Schematic of CMOS NAND gate

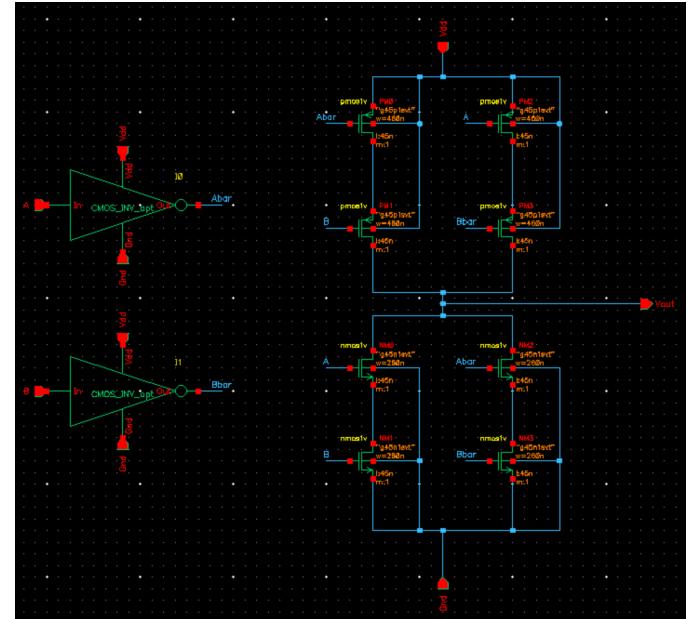


Fig. 7. Schematic of CMOS XOR gate

V. FULL ADDER AND 4-BIT RIPPLE-CARRY ADDER (RCA)

The full adder is a 1-bit adder which adds two 1-bit inputs with a carry-in bit. This implementation uses two XOR gates which produce the sum bit, and three NAND gates which produce the carry-out bit [Fig. 9].

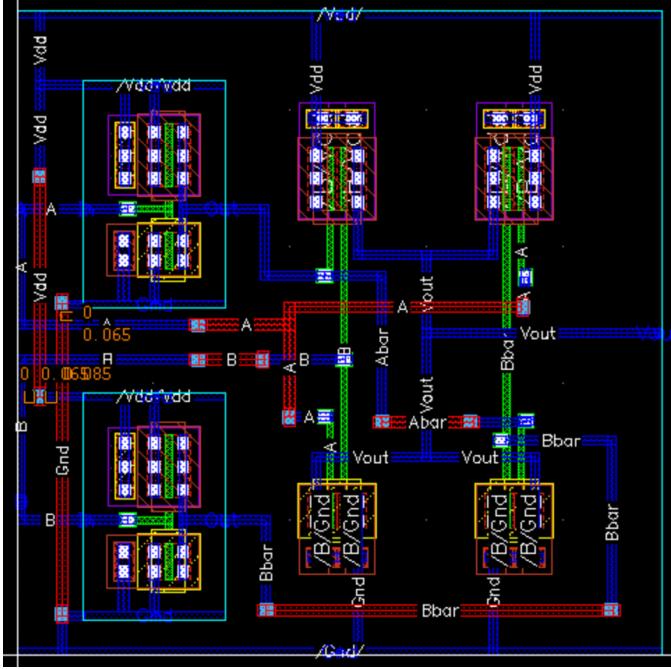


Fig. 8. Layout of CMOS XOR gate

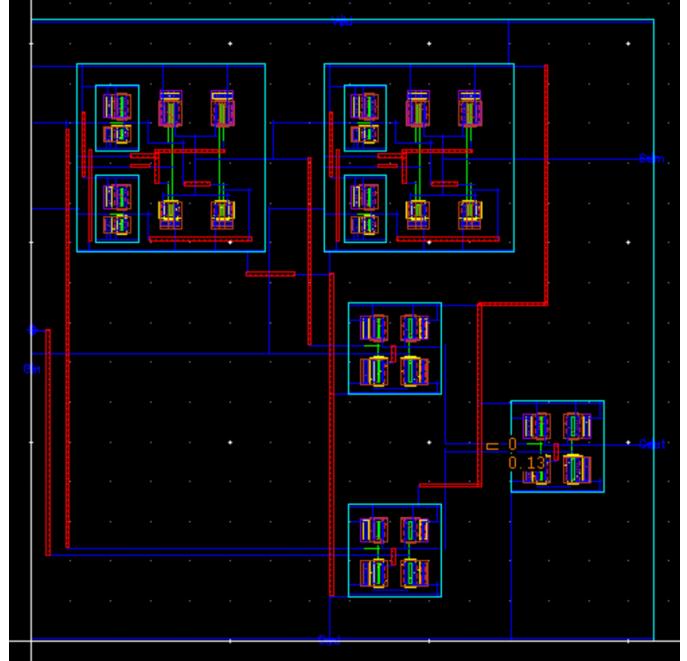


Fig. 10. Layout of full adder

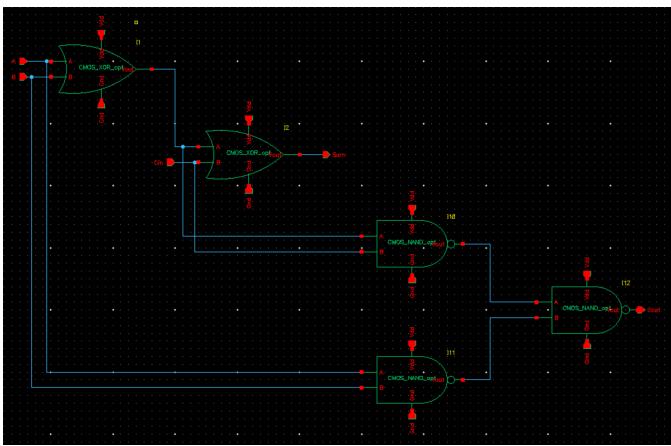


Fig. 9. Schematic of full adder

The 4-bit RCA uses four full adders chained together, where the carry-out of the previous bit carries into the next significant bit [Fig. 11].

VI. 2x1 MULTIPLEXER (MUX) AND 8x4+CARRY-OUT MUX

The 2x1 MUX is implemented using four NAND gates [Fig. 13], and the 8x4 with carry-out MUX uses five 2x1 MUXes to produce 4 out-bits and a carry-out bit chosen by a select bit [Fig. 14].

VII. 4-BIT AND 16-BIT CARRY-SELECT ADDER (CSA)

The 4-bit CSA calculates the two possible sum results in 4-bit chunks, reducing the propagation delay over the traditional RCA. In this design, the 4-bit CSA is built using

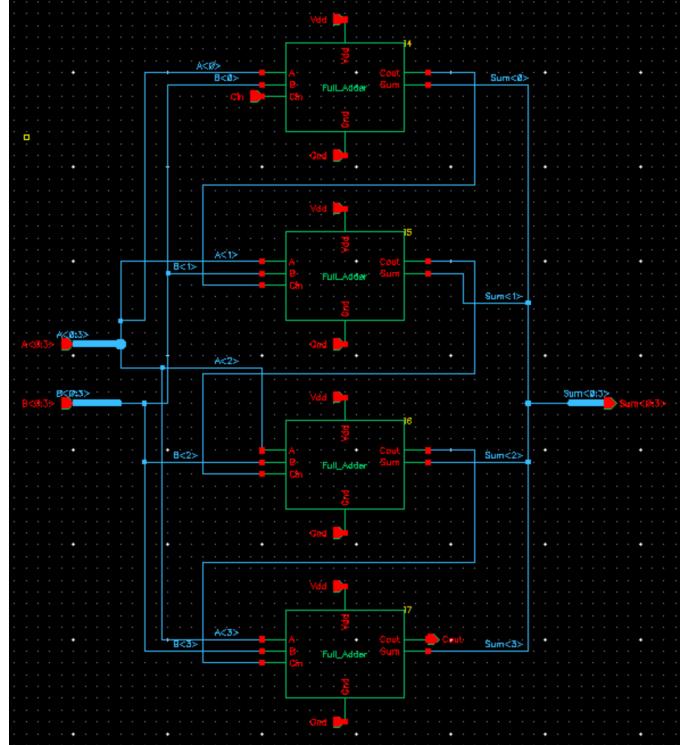


Fig. 11. Schematic of 4-bit RCA

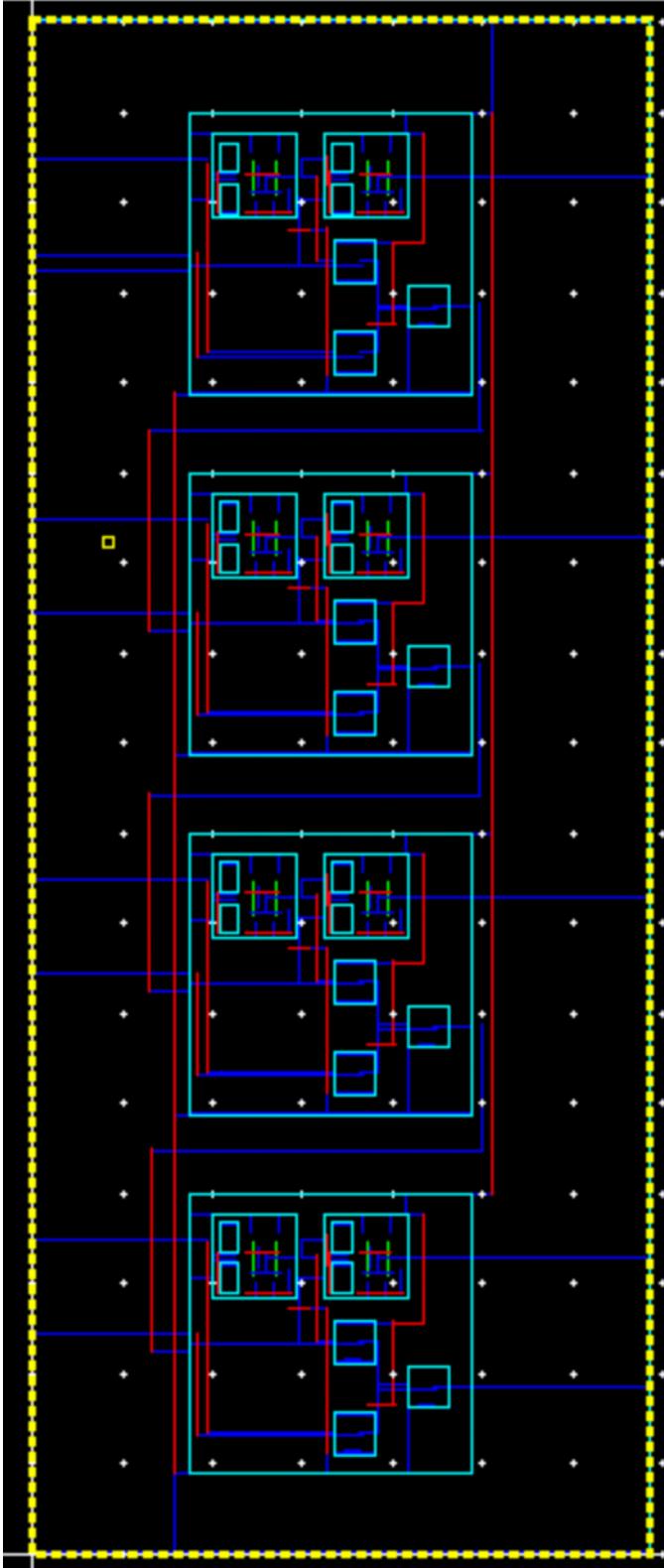


Fig. 12. Layout of 4-bit RCA

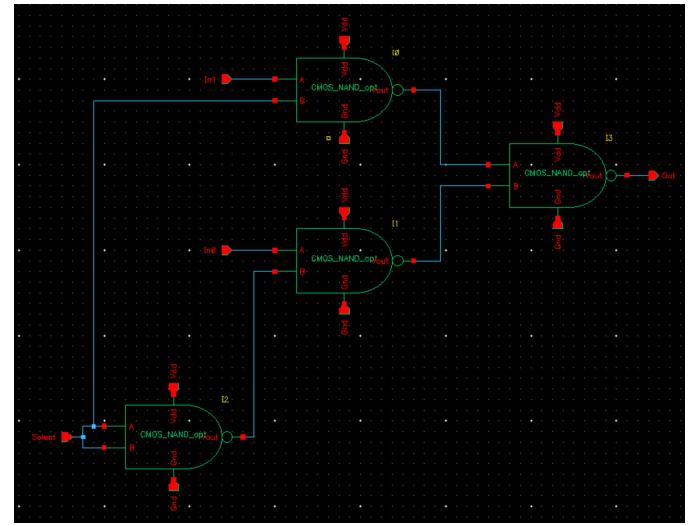


Fig. 13. Schematic of 2x1 MUX

two RCAs which take the same inputs from A and B, but with 0 and 1 as the carry-in. The 4-bit sum is calculated, then when the carry-out of the previous 4 bits is calculated the MUX selects the correct output to be sent to the next 4 bits [Fig. 15].

This implementation of the 16-bit CSA utilizes three 4-bit CSAs for the 12 most significant bits, and a 4-bit RCA for the last four least significant bits.

The worst-case propagation delay occurs when the carry-in bit switches, which results in the carry-out bit of the 16-bit CSA switching. With input A being 16 bits of 1's and B being 16 bits of 0's, the carry-in modulates from 0 to 1 and back to 0. Using the layout extracted parasitics of the design and 50 ns rise and fall times of signals, the device had a low-to-high propagation delay of 1.22 ns and high-to-low 1.42 ns, with energy consumption of 455 fJ [Fig. 22, 23].

VIII. TOPOLOGY SELECTION, DELAY AND AREA ANALYSIS

All primitive components of the design (inverter, NAND, XOR) were built using the CMOS topology. The CMOS logic family allows for low static power consumption, high noise immunity, and ability to precisely tune the PMOS and NMOS widths, providing low delay. As discussed above, the worst case propagation delay is around 1.2 to 1.4 ns, well below 2 ns. Best case, the propagation delay shows to be around 370 ps [Fig. 24]. This can be compared to a 16-bit RCA by taking the propagation delay of the full adder [Fig. 25] and multiplying it by 16 to get a theoretical delay, which results in $153 * 16 = 2448$ ps, or about 2.5 ns. The worst-case propagation delay of the 16-bit CSA is about a full ns faster than the RCA implementation. The final layout of the 16-bit CSA had dimensions of 185.77 um by 442.185 um, [Fig. 16] resulting in a total area of 82,144.70745 um^2 .

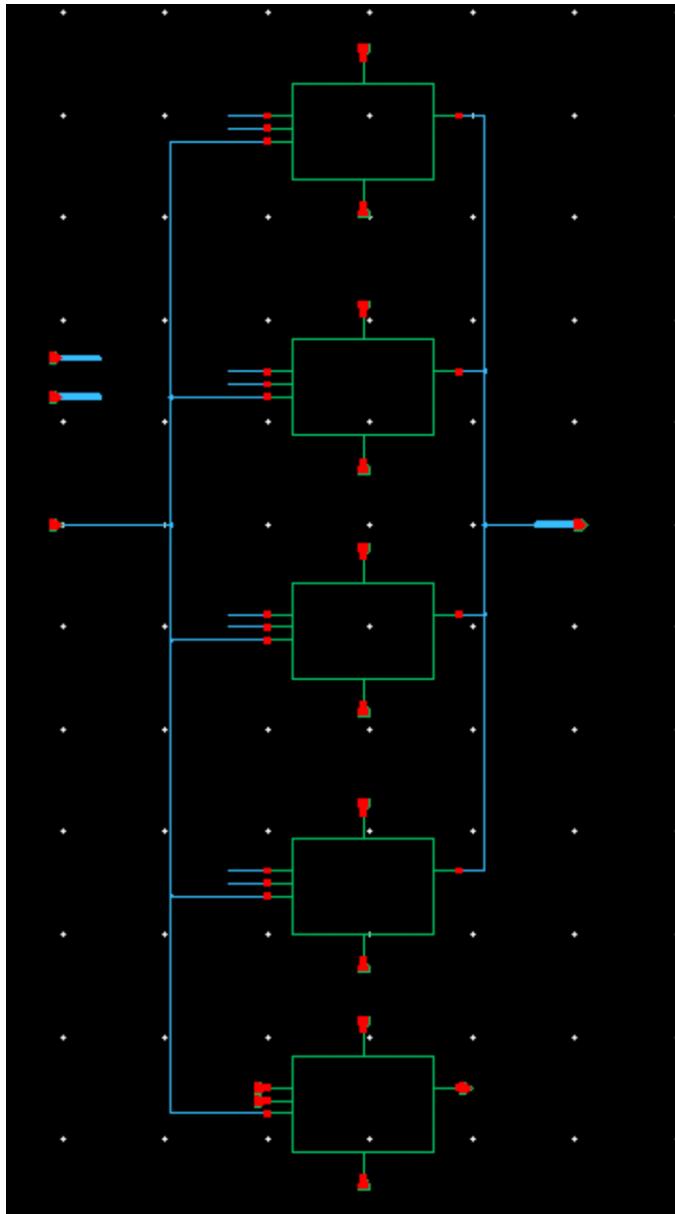


Fig. 14. Schematic of 8x4+Cout MUX

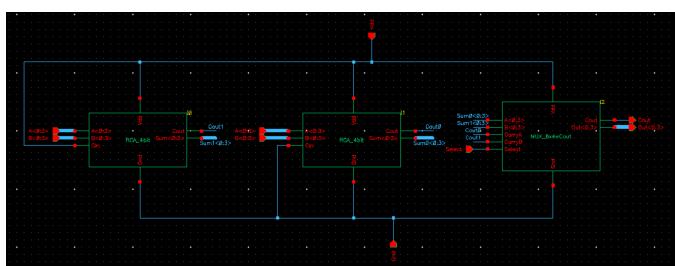


Fig. 15. Schematic of 4-bit CSA

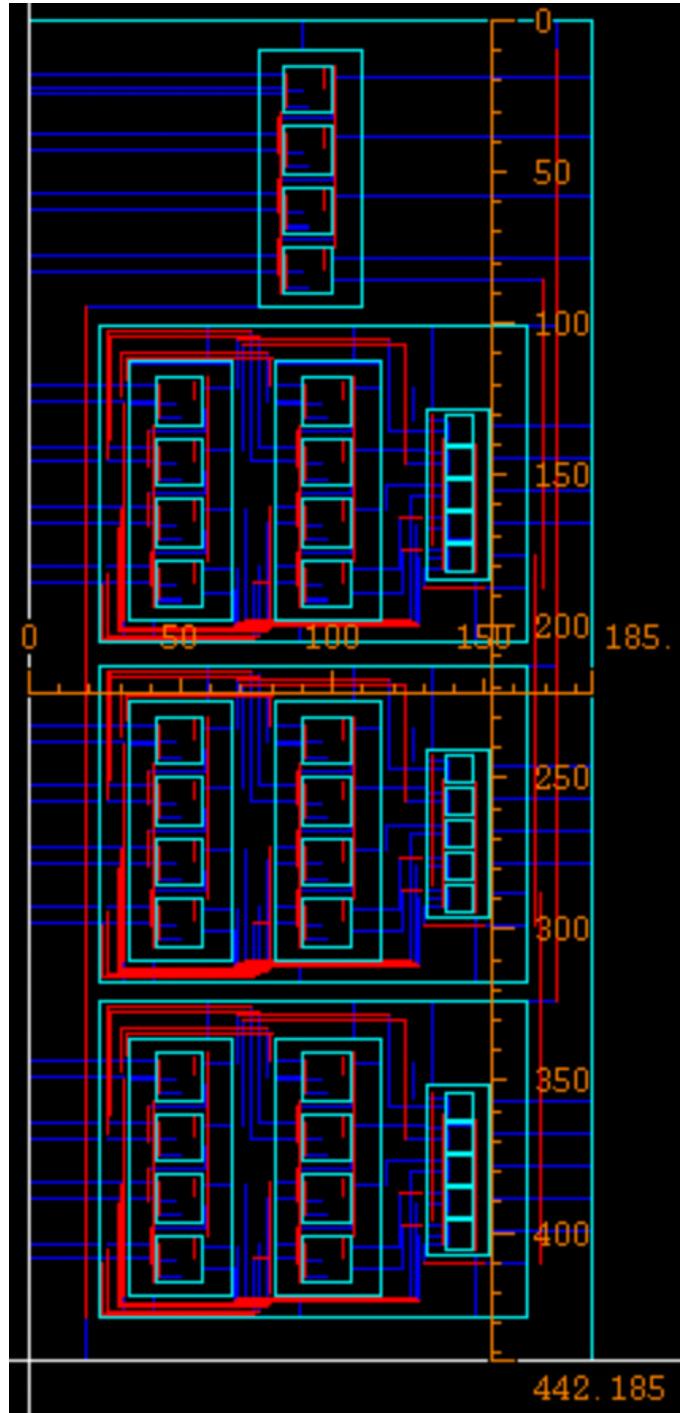


Fig. 16. Schematic of 16-bit CSA

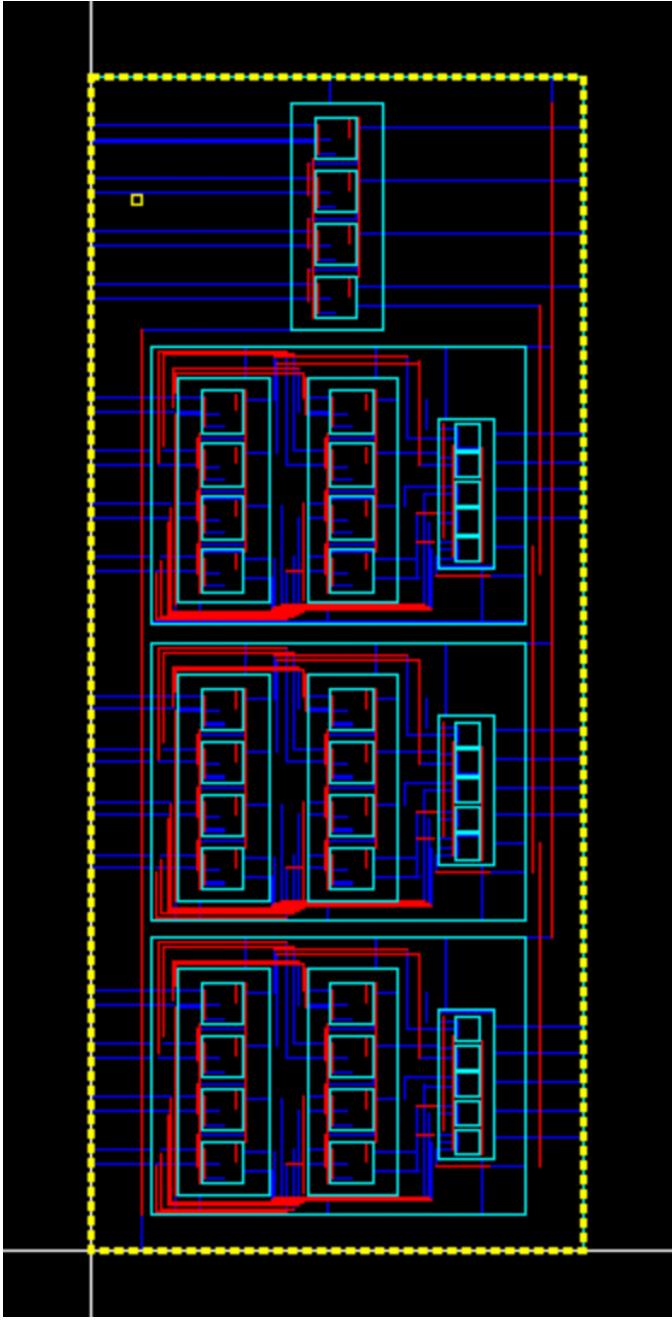


Fig. 17. Layout of 16-bit CSA

IX. CONCLUSION

The carry-select adder is an implementation of the ripple-carry adder which drastically reduces the propagation delay by calculating sums of smaller chunks ahead of time, then selecting the correct sum by using a carry-in bit. As addition is a crucial, if not the most important arithmetic function in computing devices, this optimization can play a huge role in creating faster devices.

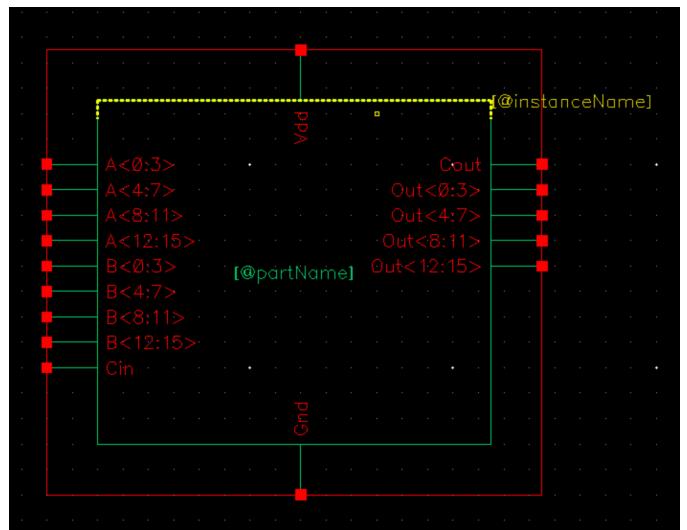


Fig. 18. Symbol of 16-bit CSA

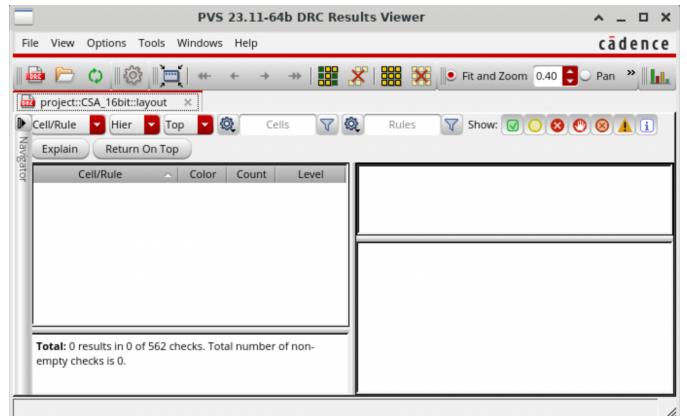


Fig. 19. 16-bit CSA DRC clean result

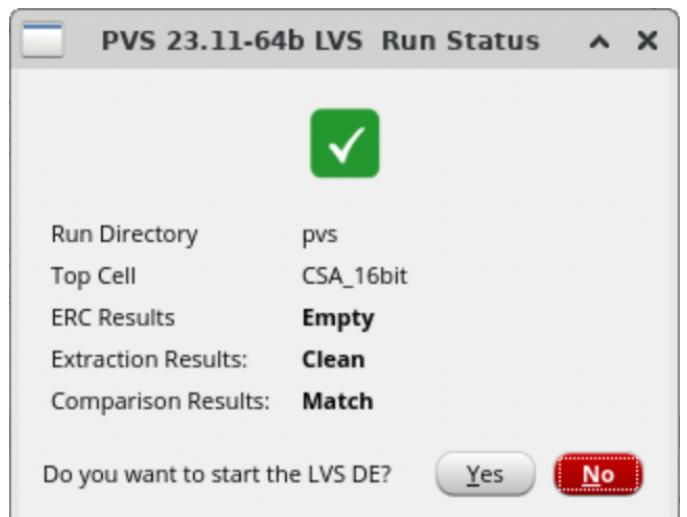


Fig. 20. 16-bit CSA LVS clean result

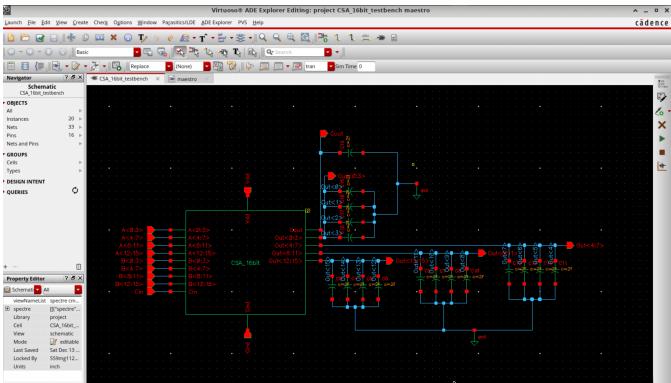


Fig. 21. Testbench schematic of the 16-bit CSA

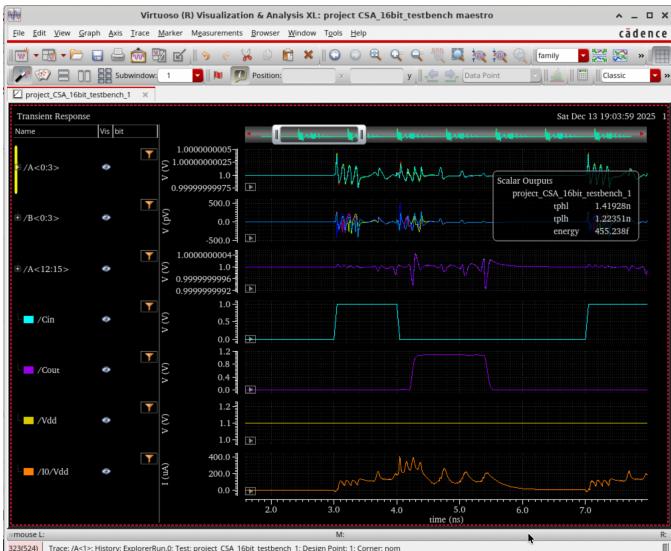


Fig. 22. 16-bit CSA worst-case propagation delay waveform

| | | | |
|--------|------|--|--------|
| tphl | expr | delay(?wf1 VT("Cin") ?value1 0.5 ?edge1 "falling") | 1.419n |
| tphh | expr | delay(?wf1 VT("Cin") ?value1 0.5 ?edge1 "rising") | 1.224n |
| energy | expr | integ((IT("I0/Vdd") * VT("Vdd")) 2.92e-09 6.2...) | 455.2f |

Fig. 23. 16-bit CSA worst-case propagation delay table

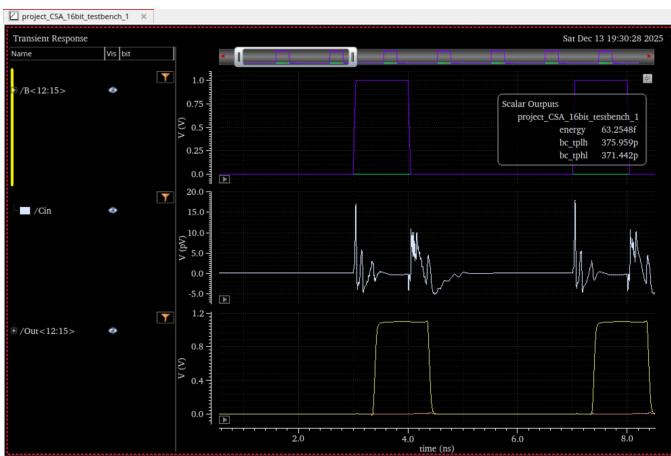


Fig. 24. 16-bit CSA best-case propagation delay waveform

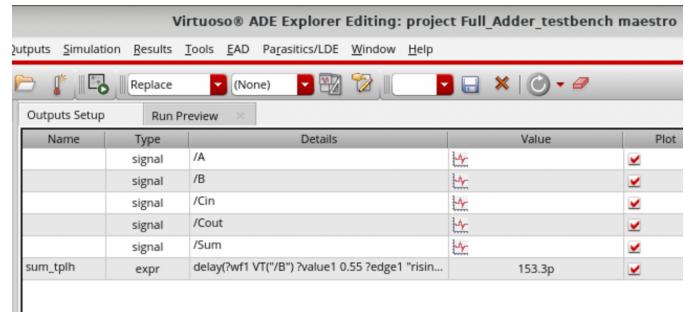


Fig. 25. Full adder propagation delay

REFERENCES

- [1] GeeksforGeeks, "Full Adder in Digital Logic," GeeksforGeeks, Mar. 21, 2017. <https://www.geeksforgeeks.org/digital-logic/full-adder-in-digital-logic/>
- [2] M. Gupta and S. Sharma, "Design and Implementation of 16-bit Adder using Carry Select and Carry Save Mode." Available: <https://www.internationaljournalsrg.org/uploads/specialissuepdf/ICT-2020/2020/OTHERS/P109.pdf>