

Optimizing 32-bit Adders in Cadence Virtuoso

Aden Briano, Aidan Lopez

I. INTRODUCTION

The purpose of this paper is to explore various Full Adder, Multiplexer, and 32-bit Adder architectures to determine the design that yields both the highest speed and/or the lowest Energy Delay Product. All designs are implemented in Cadence Virtuoso, a software tool that allows for layout and simulation of logic devices. The adders that will be primarily covered are carry bypass adders, carry select adders, and a Kogge-Stone Tree Adder.

The project makes use of the standard cell library provided by our professor, Dr. Kaiyuan Yang, for a 45 nm process.

II. FULL ADDER DESIGNS

We explored three different full adder topologies and measured their delay and EDP, in order to determine which would be used in our 32 bit adder. The topologies we explored were the standard CMOS Full Adder, the CMOS Mirror Full Adder, and the CPTL (Complementary Pass Transistor Logic) Full Adder.

A. CMOS Full Adder

The CMOS full adder was designed using standard CMOS logic. Out of the three adder topologies we tested, the CMOS full adder had the largest delay, averaging around 80 ps, and it also consumed about $8.015 \mu\text{W}$. If we set the average delay to 80 ps, we can compute the PDP and EDP of this cell. We found that the PDP was equal to $(80 \cdot 10^{-12})(8.015 \cdot 10^{-6})$, which is $6.412 \cdot 10^{-16} \text{ J}$. We can then multiply the PDP by the delay again, and we determine that the EDP of the CMOS Full Adder cell is:

$$(6.412 \cdot 10^{-16}) \cdot (80 \cdot 10^{-12}) = 5.1296 \cdot 10^{-26}$$

We determined the size of the transistors by determining the worst path and then changing the ratios of the transistors until we had the ideal effective width ratio of 2:1 (PMOS to NMOS). In the block that computed the sum, we used NMOS transistors with a length of 50 nM and a width of 270 nM, and PMOS transistors with a length of 50 nM and a width of 540 nM. In the block that computed the carry out, we used NMOS transistors with a length of 50 nM and a width of 180 nM, and PMOS transistors with a length of 50 nM and a width of 360 nM.

Images of our schematic, testbench, and simulation results are shown below. We ultimately decided to not use the traditional CMOS Full Adder when creating our 32 bit adder, since it would end up having the highest average delay and an average EDP.

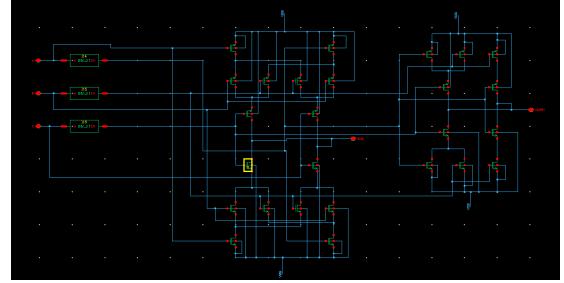


Fig. 1: Traditional CMOS Logic Implementation of Full Adder module.

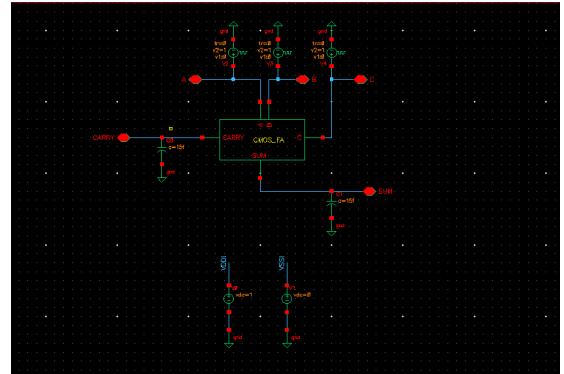


Fig. 2: Traditional CMOS Full Adder Testbench Schematic

B. Mirror Full Adder

The Mirror Full adder was designed using mirrored CMOS logic. The mirror full adder's delay was much faster than the traditional CMOS full adder, averaging around 55 ps, and it also consumed about $7.067 \mu\text{W}$. If we set the average delay to 55 ps, we can compute the PDP and EDP of this cell. We found that the PDP was equal to $(55 \cdot 10^{-12})(7.067 \cdot 10^{-6})$, which is $3.886 \cdot 10^{-16} \text{ J}$. We can then multiply the PDP by the delay again, and we determine that the EDP of the CMOS

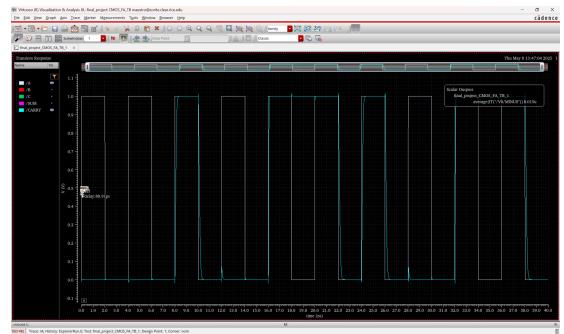


Fig. 3: Traditional CMOS delay from A to SUM.

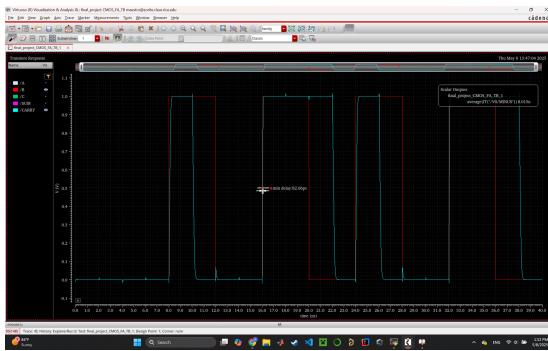


Fig. 4: Traditional CMOS delay from B to SUM.

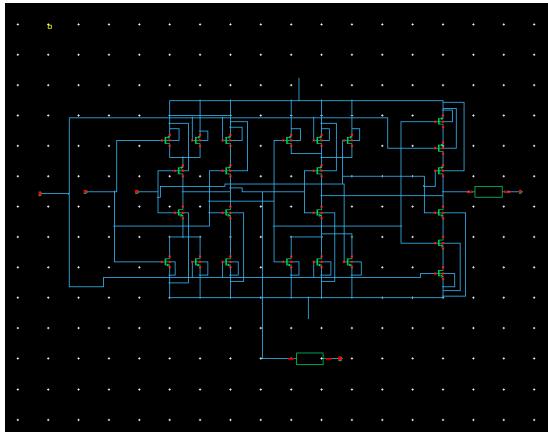


Fig. 5: Mirror Adder Schematic

Full Adder cell is:

$$(3.886 \cdot 10^{-16}) \cdot (55 \cdot 10^{-12}) = 2.138 \cdot 10^{-26}$$

We determined the size of the transistors in this design through a combination of the lecture notes that covered various adder topologies and calculating what would be the best width for the pull up and pull down networks. Images of our schematic, testbench, and simulation results is shown below. Due to the mirror adders low EDP, we left it in consideration for our 32 bit adder design.

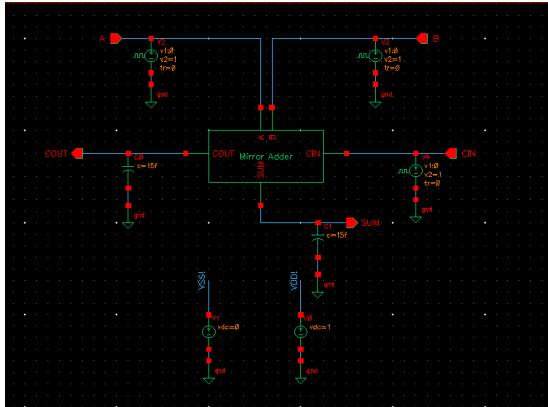


Fig. 6: Mirror Adder Testbench

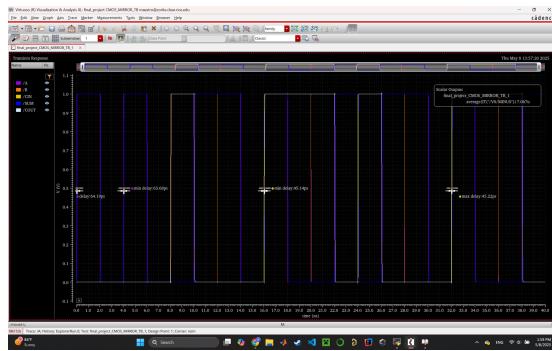


Fig. 7: Mirror Adder Delays from Inputs to SUM Output

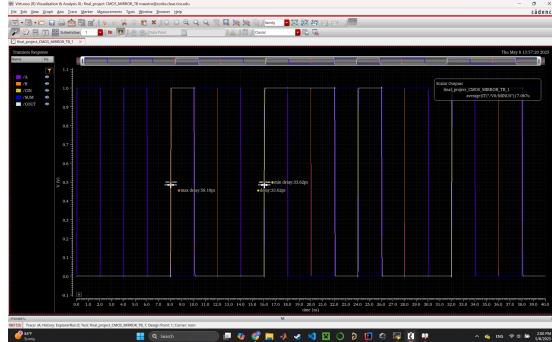


Fig. 8: Mirror Adder Delays from Inputs to C_Out

C. CPL (Complementary Pass Transistor Logic) Full Adder

The CPL Full Adder is a Full Adder design using a Complementary Pass Transistor Logic architecture. Because of the CPL Full Adder's primary use of NMOS transistors, it has an advantage over the conventional CMOS Full Adder in power dissipation, area, and overall propagation delay. The CPL Full Adder can be seen in Fig. 9.

To determine the optimal sizing of the transistors in our Complementary Pass Transistor Logic, we referenced *High performance Complementary Pass transistor Logic full adder* authored by Lixin Gao and published by IEEE. The reference image can be seen in Fig. 10.

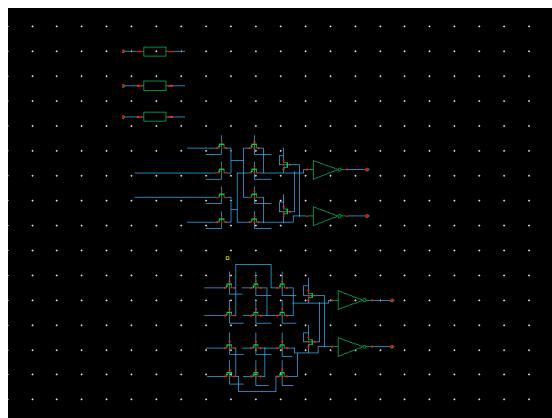


Fig. 9: Complementary Pass Transistor Logic architecture implementation of Full Adder module.

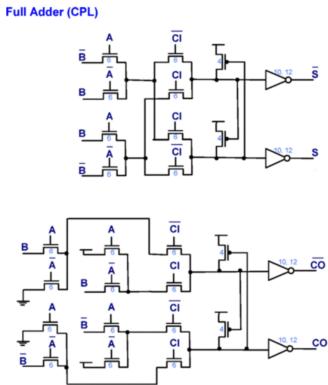


Fig. 10: CPTL Sizing Reference

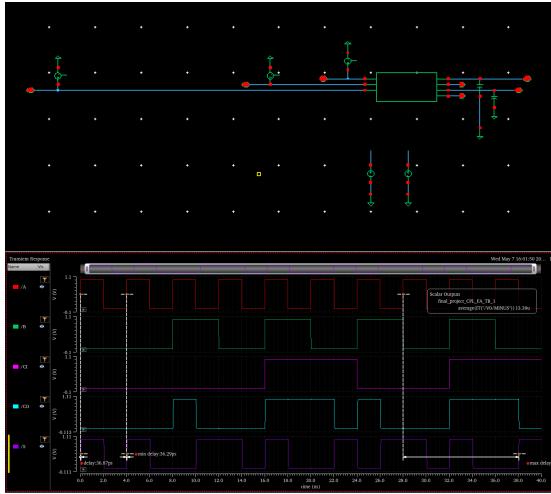


Fig. 11: CPL Testbench and simulation waveforms

The testbench used to determine the delay and average current of the CPTL Full Adder can be viewed in Fig. 11 , alongside the simulation waveforms with the measured delay and average current.

Because the supply voltage is 1V, we can easily convert our measured average current draw to average power consumption.

$$P = IV = (13.39 \cdot 10^{-6} A)(1V) \\ = 13.39 \mu W$$

The corresponding EDP is thus

$$EDP = P \cdot t_p^2 = (0.01339\text{mW})(0.03629)^2 \\ = 1.76 \cdot 10^{-5}\text{mW}\cdot\text{ns}^2$$

D. Section Summary

We evaluated three different full adder topologies and determined that the mirror full adder and the CPTL Full Adder best fit our intended use cases. This was primarily due to the fact that the Mirror Full Adder has a low power consumption, and the CPTL Full Adders have a high speed. Thus, we will explore and utilize both types of full adders when creating our 32-bit adder.

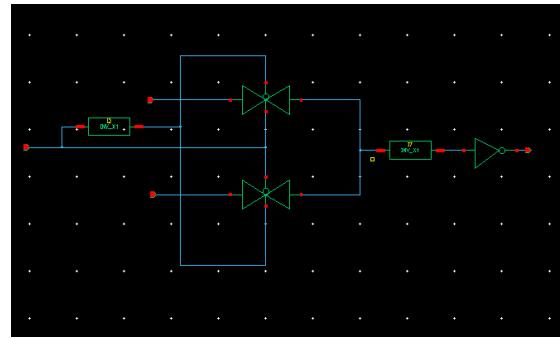


Fig. 12: Transmission gate multiplexer.

III. MUX DESIGNS

After concluding the development and analysis of various Full Adder architectures, we moved onto iterating on the second module most prevalent in our desired 32-bit Adder designs: the Multiplexer.

A. Transmission Gate Multiplexer

The first design we made was a transmission gate multiplexer, which can be seen in Fig. 12. Our primary reason for choosing this design was because of the small number of transistors. We were also interested in the fact that this design would be able to efficiently pass through both ones and zeros, and that there would likely be symmetric rise/fall times. For this design, we eventually choose to have a 2:1 ratio between the PMOS and the NMOS in the transmission gate, since this would allow for faster rise times. We also added a buffer/inverter chain to the output to ensure that we had full rail-to-rail voltage swings from the input to the output.

B. PTL (Pass Transistor Logic) Multiplexer

The second multiplexer we had in mind was the PTL, or Pass Transistor Logic, multiplexer. This multiplexer design would theoretically provide us with faster switching and smaller area, at the expense of increased power dissipation due to static power dissipation. The design can be seen in Fig. 14.

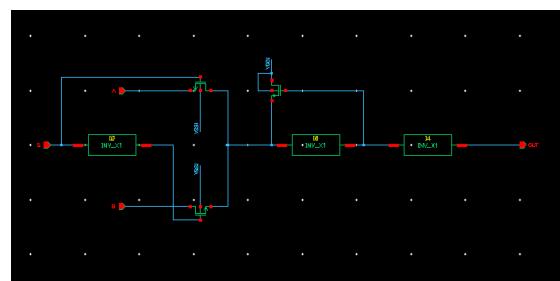


Fig. 13: Pass transistor logic multiplexer.

To determine the optimal sizing of the transistors in our PTL multiplexer, we resorted to a more brute-force method. We knew that we wanted our NMOS devices to be fairly wide to decrease resistance and increase switching speeds, and that our inverter to produce S bar can be minimum sized since

it's only driving a single NMOS device. We also wanted our PMOS to be strong enough to fix any degraded logic HIGHs, but not so strong that it overpowers our NMOS devices. The testbench we developed for the PTL Multiplexer can be seen in Fig. 14, as well as the measured delay and average current.

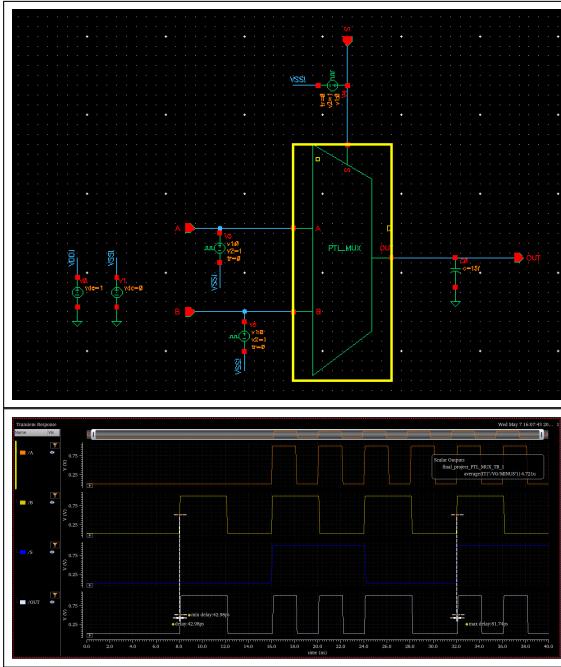


Fig. 14: PTL Mux testbench and measured delay + average current.

The corresponding EDP is thus

$$\begin{aligned} EDP &= P \cdot t_p^2 = (0.004721\text{mW})(0.04298)^2 \\ &= 8.721 \cdot 10^{-6}\text{mW}\cdot\text{ns}^2 \end{aligned}$$

C. Section Summary

We successfully developed and tested two different Multiplexer topologies. In the end, it wasn't quite clear which of the two would best suite our needs, so we decided to implement both of them into separate 32-bit adder designs. This will be the subject of our next section.

IV. EXPLORED 32 BIT ADDERS

A. Carry Square Root Select Adder (CSA)

When developing the Carry Square Root Select Adder, we went through multiple iterations. Pictured below is our CSA with our PTL multiplexers and CPTL Full Adders. They both proved to be slower, as well as more power hungry, than our final design, which utilizes the transmission gate multiplexers and Mirror Adders.

Our first iteration utilized the CPTL Full Adders and transmission gate adders. This version has a solid critical path delay of about 480 ps, however it consumed almost 380 μW . This led us to explore methods of reducing the total amount of power used, which caused us to switch our design to utilize the mirror adder cells. We also tried experimenting with the

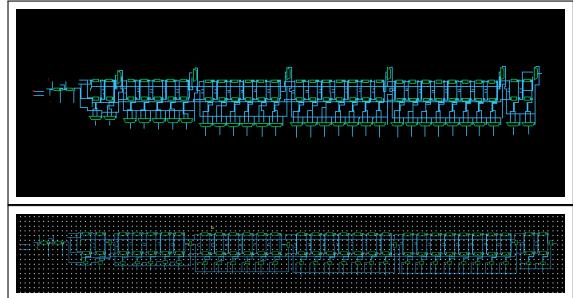


Fig. 15: CSA 32-Bit Adder Iterations

PTL multiplexers, however this caused an increase to delay and power consumption. We eventually decided to create larger transmission gate multiplexers that were placed in between the stages, since they had to drive multiple outputs. We also used the smaller transmission gate multiplexers for when the sum was being computed.

Our final design, which can be seen in Fig. 16, was our fastest. The average delay of our CSA 32-bit adder was

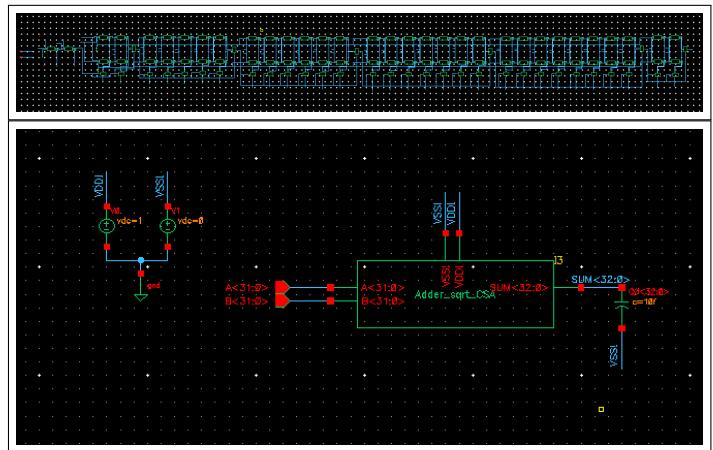


Fig. 16: CSA with TGates and Mirror Full Adders

464.1ps, corresponding to a clock speed of 2.15 GHz. The calculated average current draw was $159\mu\text{A}$. This corresponds to an EDP of $0.03424\text{ mW}\cdot\text{ns}^2$. Refer to Fig. 17. Our design behaved as expected as well. Refer to Fig. 18.

After adding all the NMOS and PMOS widths in our cell, we estimate our cell to be about 525,105 nM.

B. Kogge-Stone Tree Adder

After we were satisfied with our CSA 32-bit adder, we wanted to try our hand on a 32-bit Kogge-Stone tree adder. Unfortunately, we did not quite get it to work, and though it was a grueling process, we learned a lot and are more prepared for our next encounter with the infamous adder. We referenced Github user salzhang's schematic of a 32-bit Kogge-Stone tree adder.

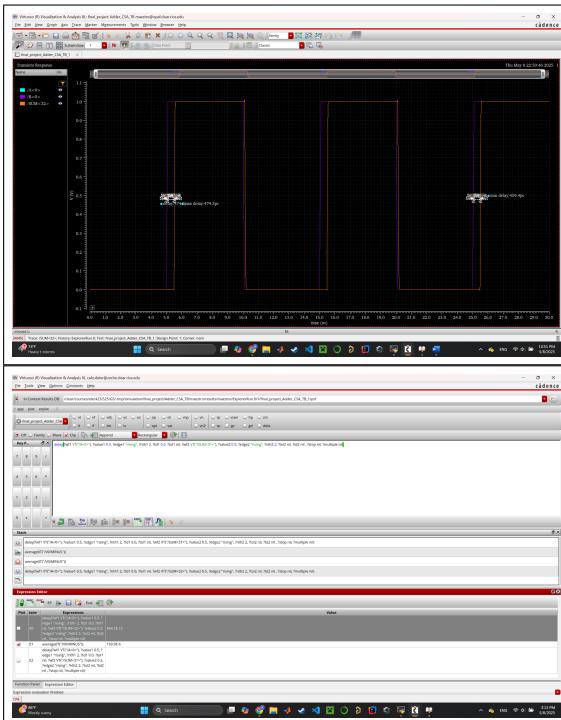


Fig. 17: Calculated delay and current draw of 32-bit Mirror CSA

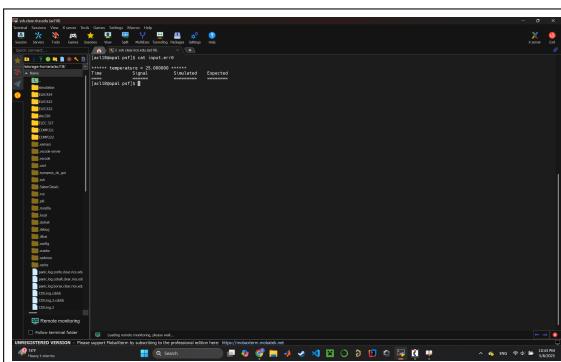


Fig. 18: No errors after testing Mirror CSA.

V. CITATIONS

- 1) L. Gao, "High performance Complementary Pass transistor Logic full adder," Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, Harbin, China, 2011, pp. 4306-4309
 - 2) Salzhang. (n.d.). Salzhang/Koggestone-Adder: A 32-bit Kogge-Stone Adder is implemented in this design. GitHub. <https://github.com/salzhang/KoggeStone-Adder>

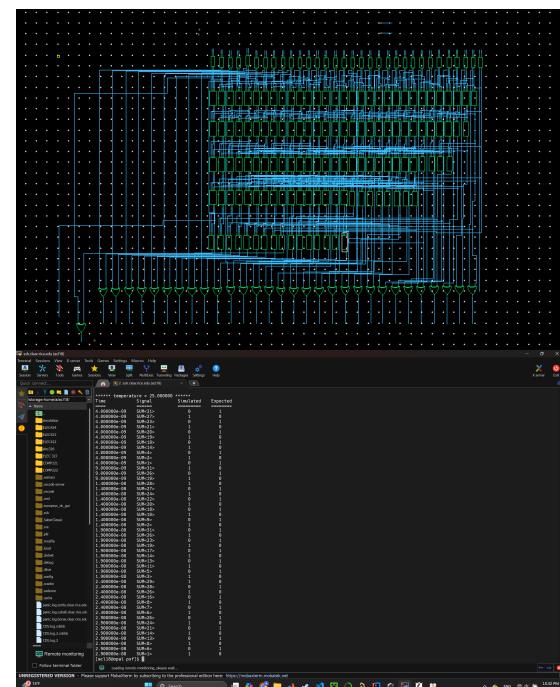


Fig. 19: Kogge-Stone Tree Adder