

# Augmented Efficiency of CLA Logic through Multiple CMOS Configurations

Naga Spandana Muppaneni

Electrical Engineering, Idaho State University  
Pocatello, USA 83209  
muppnaga@isu.edu

Steve C. Chiu, PhD, IEEE SM

Electrical Engineering, Idaho State University  
Pocatello, USA 83209  
chiustev@isu.edu

**Abstract**—Adders are the basic building blocks in the digital systems. Addition is an indispensable operation in digital, analog and control systems. Performance optimization of a digital adder relies on the parameters such as power, speed and area. Much research has been going on in optimizing the delay and power dissipation of adders. Carry-lookahead adder (CLA) is considered one of the fastest digital adders. It emerges from the concept of computing all incoming carriers in parallel. This paper introduces various design implementations using CMOS transistors, producing unique logic of CLA adder. Each design implementation is analyzed by assessing the power dissipation and the delay at every possible state by transistor sizing. Simulations have been performed on Tanner EDA tools based on 250nm technology at 2.5V supply voltage. Previous work on CLA has been examined, and the 8-bit design of CLA and its delay and power dissipation has been evaluated.

**Keywords**—Carry-lookahead adder (CLA); performance optimization; simulations; transistor sizing; power dissipation; Tanner EDA tools

## I. INTRODUCTION

Binary addition is one of the most primitive operations of the digital systems. In today's world, we are influenced by digital systems every day. They are utilized in many devices to perform arithmetical and logical operations. The core of all these systems is a Central Processing Unit (CPU), and the heart of the CPU is the Arithmetic Logic Unit (ALU). One of the critical and common capabilities of the ALU is its ability to perform addition. Any increase in the CPU's performance will increase the speed of the machine over all. Hence, optimizing the performance of the digital adder is always an interesting area of research [1].

Single-bit addition is readily accomplished by using an XOR gate to perform the summing as well as an AND gate to perform the carry. However, as machines progress to 64-bit technology and beyond it is easy to see that this solution quickly becomes unfeasible. The alternative option in speeding up the CPU is to create a simple two-level logic adder. In order to create this form of adder, a logical truth table for all combinations of inputs as well as outputs is formed. The logic that satisfies the generated table is created by implementing a level of AND gates that feeds a subsequent level of OR gates. Although this solution is fast, it grows exponentially with the number of input bits.

An additional solution is the ripple-carry adder [3]. This adder uses a 5-gate full adder block for each bit and processes

carry bits by cascading a carry-out and carry-in between each block. The configuration of this type of adder greatly reduces the number of transistors required to perform 64-bit calculations. Yet, the rippling of the carry bit can cause large propagation delays within the systems.

As a final compromise between the speed of the two-level adder and the size of the ripple carry, we looked at the carry-lookahead adder (CLA) [5]. Propagation delays within the ripple-carry adder occur because the carry bit is computed alongside the sum bit, meaning each bit must wait until the previous carry bit is computed before its own result and carry can be calculated. This is not the case with the CLA as it generates one or more carry bits before computing the sum bit, resulting in faster operation [4]. Additionally this means that the CLS contains three signals within its design a sum (S), a carry (G), and a propagate (P). At the single bit level, input signals are said to generate if both A and B are true and propagation of either A or B is true. In logic terms,  $G = A \cdot B$  and  $P = A + B$ . This sets up a system of binary addition, where each adder block can input a C that depends on the G and P signals of every bit before it. S can then be generated without having to wait for the rippling S and C signals of each previous block [8]. Thus, C is generated by the logic equation.

$$C_{i+1} = G_i + (P_i \cdot C_i) \quad (1)$$

Where, "i" is the binary digit of the adder block [3].

This paper will focus on the carry signal generated by an 8-bit carry block. It is always said that "the lower the supply voltage, the better" [2]. This paper elucidates how lowering the overall supply voltage reduces the power dissipation of the circuits, without compromising in its frequency and this paper proposes an 8-bit Carry-Look ahead adder circuit design by expanding and re-doing the work of [1] using Tanner S-edit tool by reducing the supply voltage to 2.5v and it also shows how the reduction in the sizing of the transistor geometries paves the way to provide optimized results [7].

The rest of the paper is organized as follows. Section II describes the past work of CLA configurations. Section III presents the design, schematics and simulation results of the 4-bit and 8-bit CLA. Section IV analyzes the transition time of all the critical paths with their waveforms of transient analysis and current consumption. Sections V, VI and VII discuss the transistor sizing, future considerations and conclusion of the work.

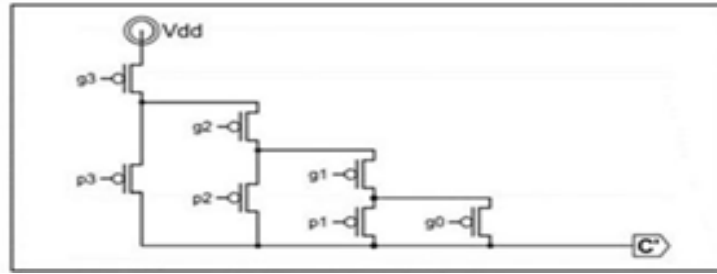


Fig. 1. PMOS block, type A.

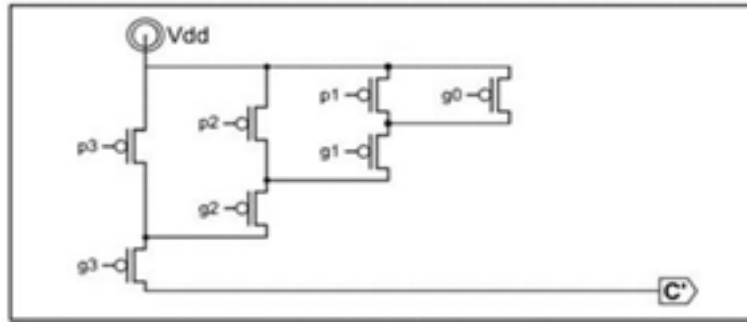


Fig. 2. PMOS block, type V.

## II. LITERATURE REVIEW

The two major constraints, for a digital adder are speed and the power dissipation [12]. Normally, though the dynamic power of a CMOS circuit is to a large extent dependent on the supply voltage, parasitic capacitance and the frequency of operation, the overall supply voltage has the largest effect [3]. In paper [2], it is discussed that reduction of power dissipation and short power leakage of CLA could be done by sizing the transistors.

**Block Inversion:** The work presented in [1] states mathematically that there is a unique solution to the carry-lookahead equation. The logic of the equation can be implemented in many fashions using CMOS technology [9]. From previous works, for example, we can discover that PMOS blocks may be inverted to arrive at a logically equivalent circuit that is physically different.

This is illustrated in Fig. 1 and 2 [1]. Due to the nature of CMOS technology, any number of equivalent circuits that are physically different can be constructed [10]. Yet because of physical constraints within the technology itself, we can surmise that one of these solutions is faster than the others in terms of propagation delay. This is called block inversion [1].

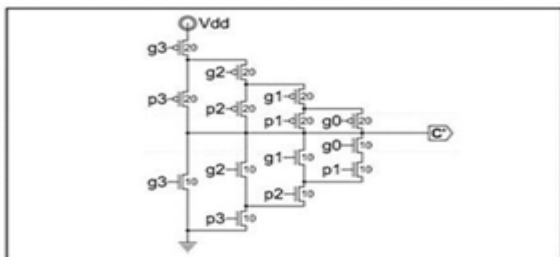


Fig. 3. Carry for 4-bit CLA, method  $A_P V_N$ .

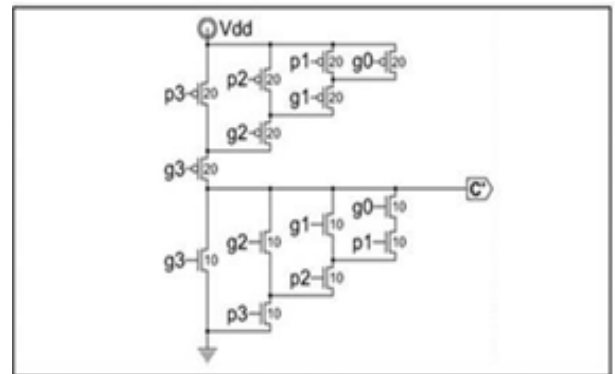


Fig. 4. Carry for 4-bit CLA, method  $V_P V_N$ .

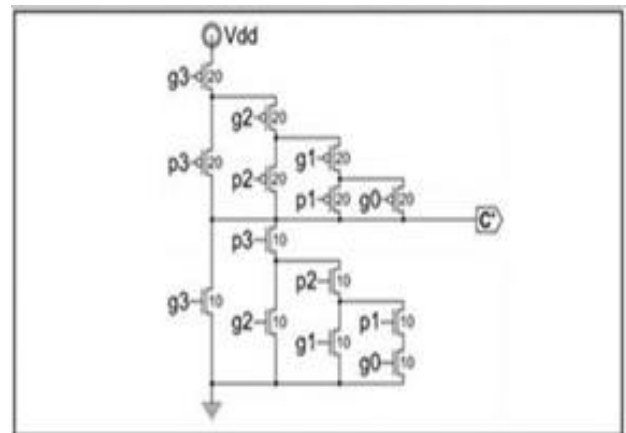


Fig. 5. Carry for 4-bit CLA, method  $A_P A_N$ .

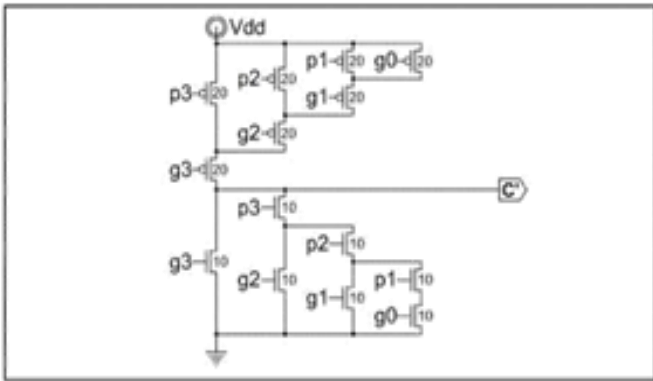


Fig. 6. Carry for 4-bit CLA, method VpAn.

**CMOS Configurations:** Fig. 3 to 6 [1] depicts four possible solutions when the PMOS block is combined with an NMOS block. The propagation delays of each of the four solutions are analyzed using Electric VLSI and LTSpice tools. All the designs discussed in the previous section only account for a 4-bit carry-lookahead. These designs are only applicable for standalone 4-bit adders with no carry in.

If an application demands an adder larger than 4 bits, or requires a carry in, then some other CLA designs must be used. The 4-bit layouts discussed in the previous works [1] can be expanded to 8-bit along with the analyzation of their propagation delays.

### III. DESIGN IMPLEMENTATIONS OF 4-BIT AND 8-BIT CLA

Taking the past implementations from [1], as a reference, and using Tanner S-edit as a design tool, with the 0.25um, CMOS model parameters, the 4-bit designs have been recreated for all the four configurations with transistor sizing. Fig. 7 to 10 represents the 4-bit CLA configurations, whereas Fig. 11 represents the schematic of an 8-bit CLA. It has been designed by cascading two 4-bit CLA's, back to back. Their implementations are shown as follows in Fig. 7 to 10.

**Simulations:** Simulations of all the four configurations are performed in Tanner T-edit. Firstly, T-spice code is set up to test the four bit adders listed in the figures above. Moreover, during the creation of the code certain methods were put into place in order to ensure that the algorithm would, at the very least, be scalable to an eight bit adder. The four bit adders represented above requires seven input signals, which results to  $2^7$  states for the circuit to be in at any given time [11]. For a perfect outcome, a stable period between the states is necessary, in order to avoid any transient behavior within the transistors.

As well, as we have a large number of state variations within the circuit, a method to automate the recording of delay between the state variations would be required. This technique was found in both Tanner EDA tools.

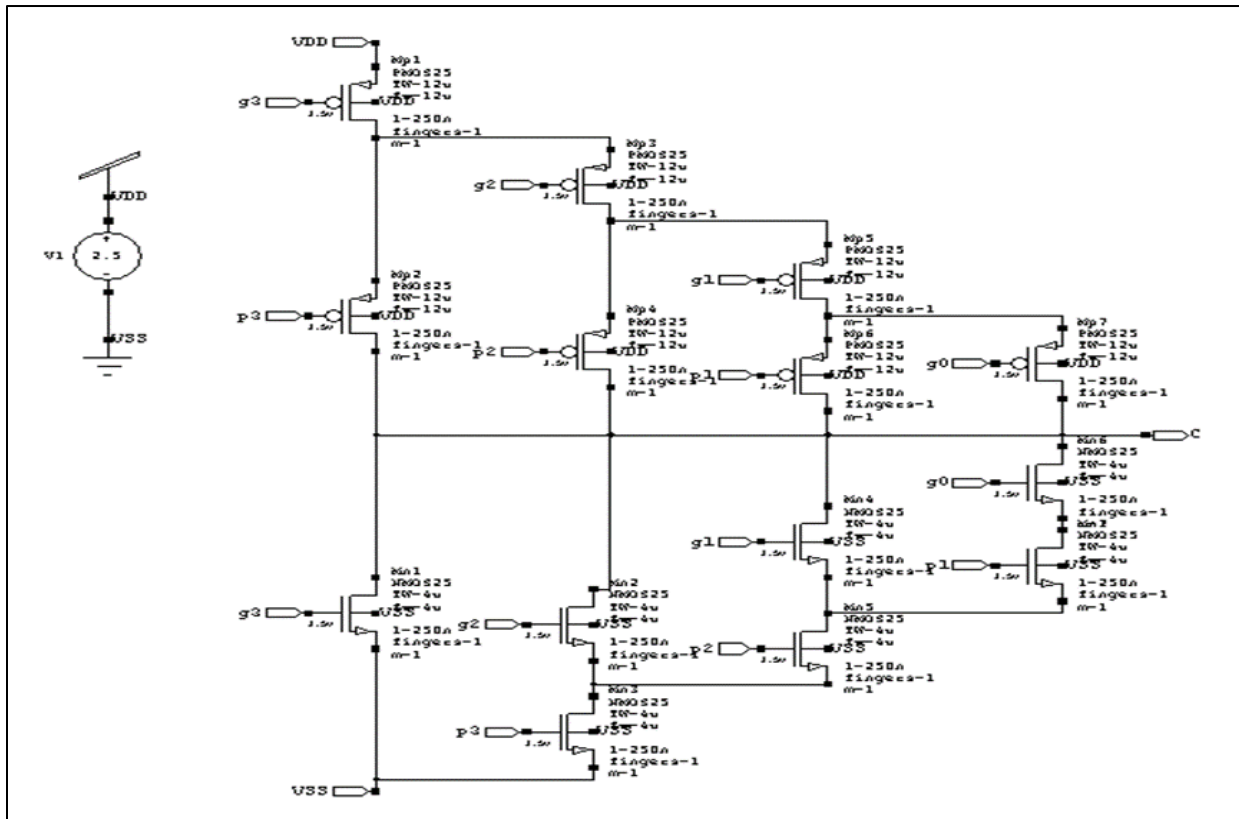


Fig. 7. Carry for 4-bit CLA, method ApVn.

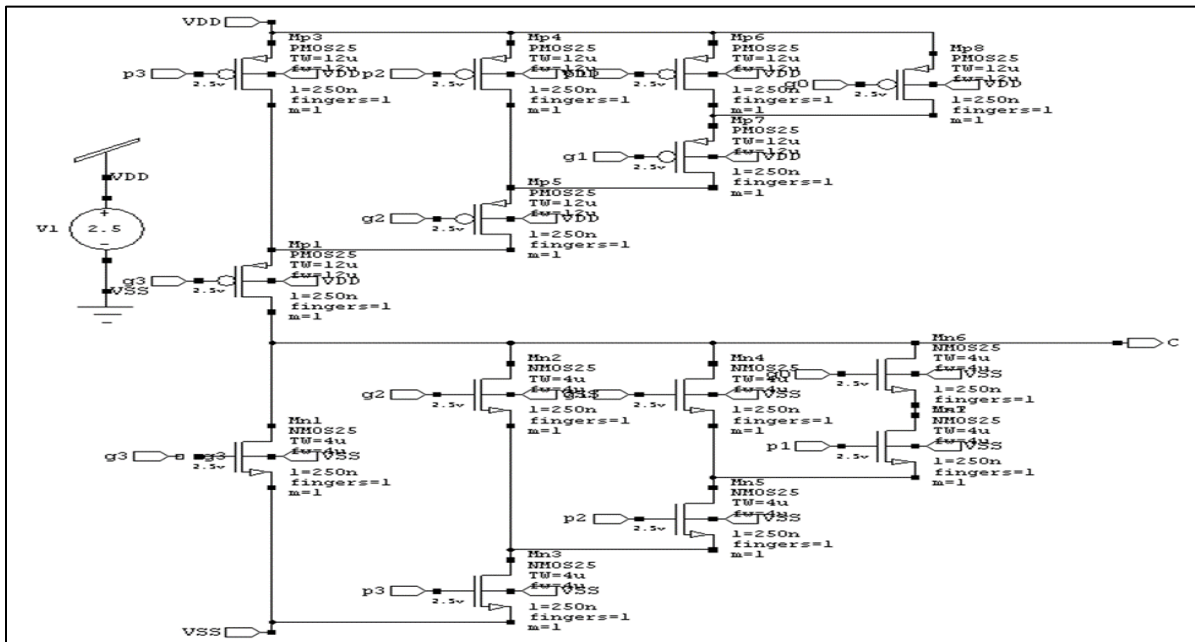


Fig. 8. Carry for 4-bit CLA, method VpVn.

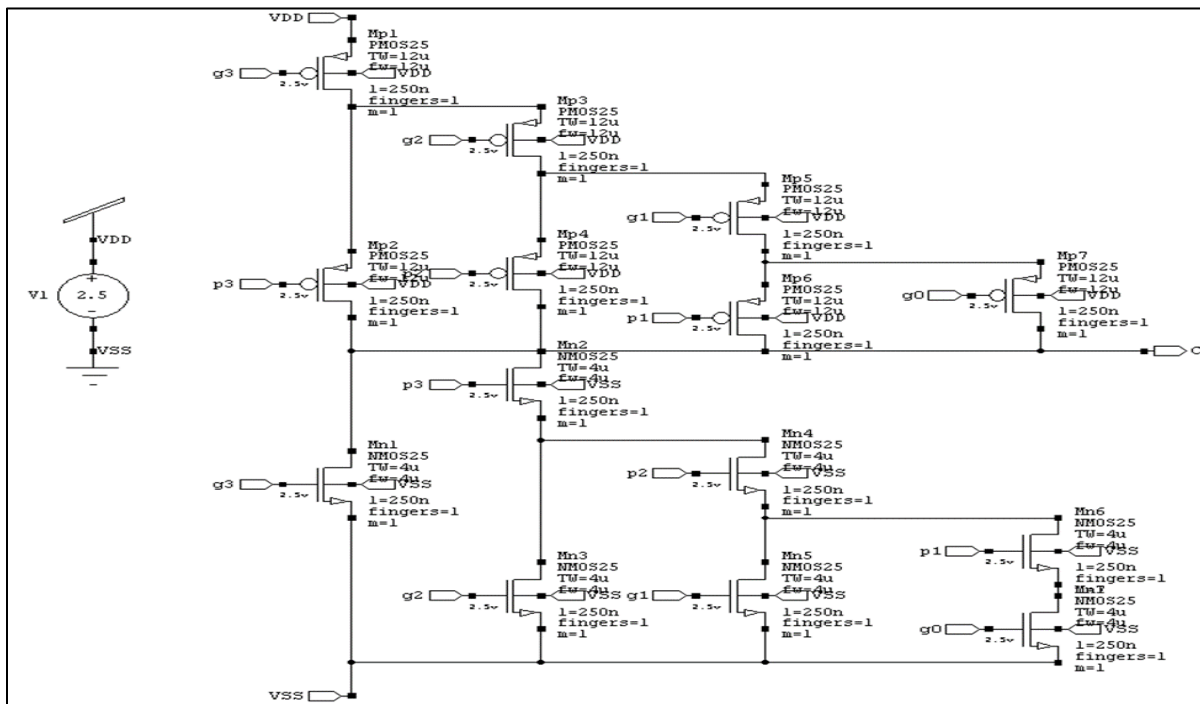


Fig. 9. Carry for 4-bit CLA, method ApAn.

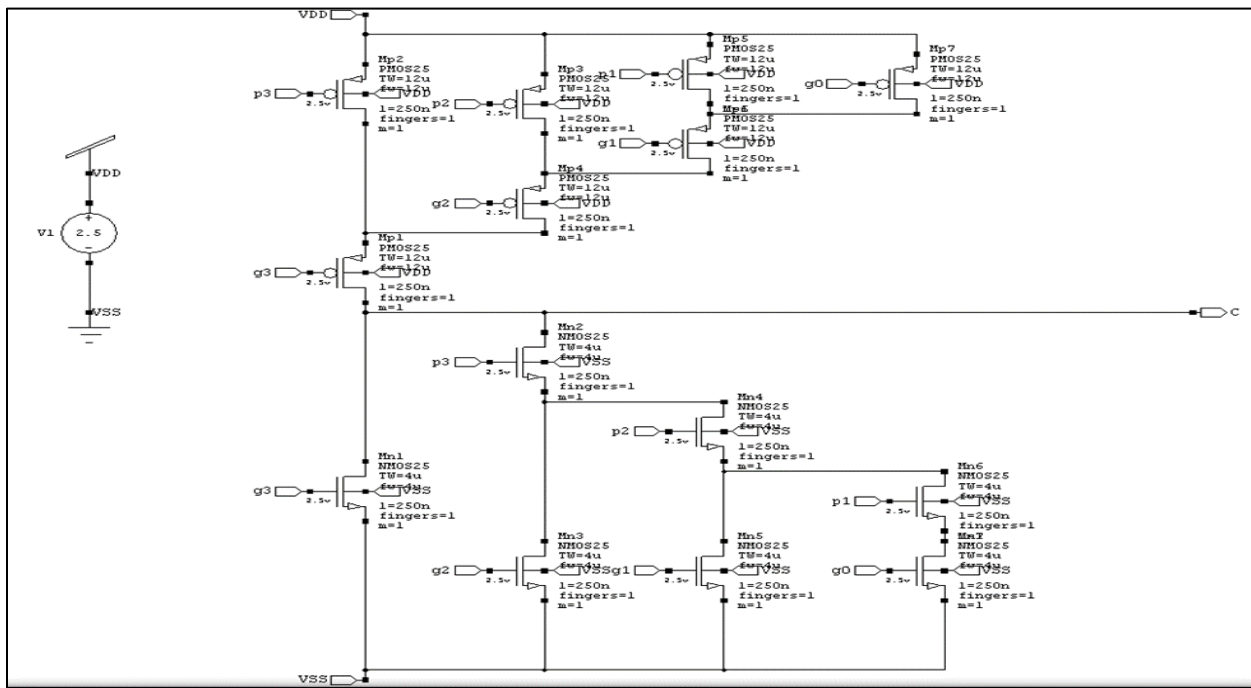


Fig. 10. Carry for 4-bit CLA, method VpAn.

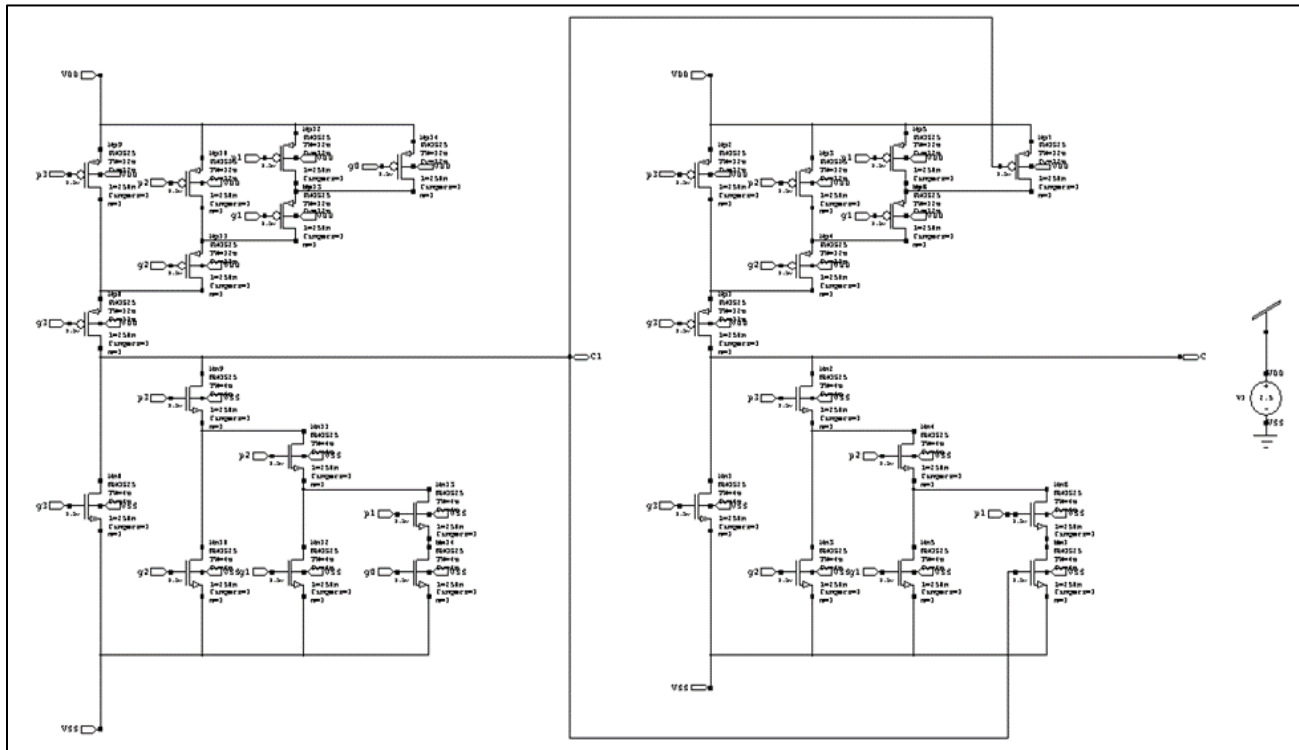


Fig. 11. Carry for 8-bit CLA, method VpAn.

Fig. 12 to 16 represents transient analysis response waveforms of 4-bit and 8-bit CLA configurations represented in Fig. 7 to 11 respectively. The rise and fall times at each and

every transmitting path are observed and hence, the delay check at every possible state is permitted by this technique.

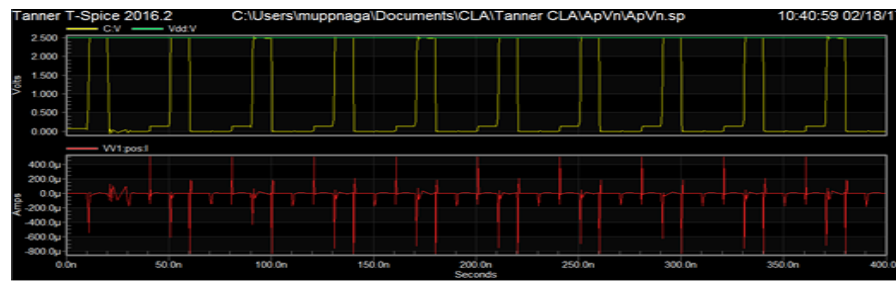


Fig. 12. Transient Analysis of 4-bit ApVn.

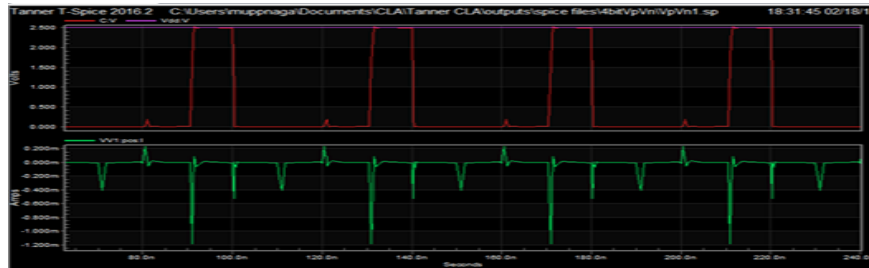


Fig. 13. Transient Analysis of 4-bit VpVn.

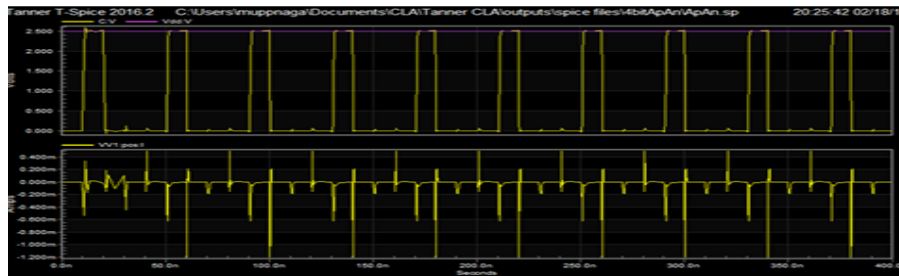


Fig. 14. Transient Analysis of 4-bit ApAn.

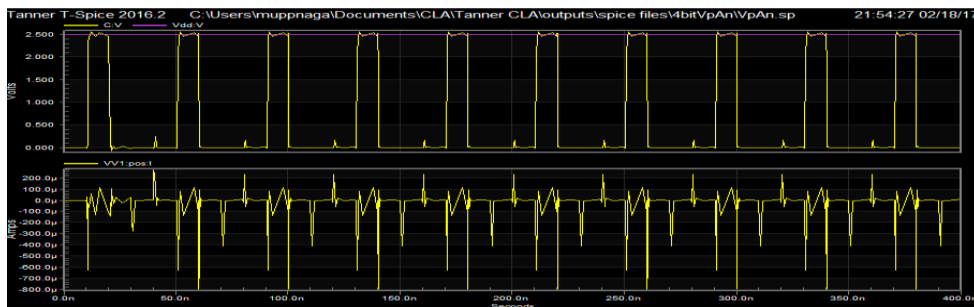


Fig. 15. Transient Analysis of 4-bit VpAn.

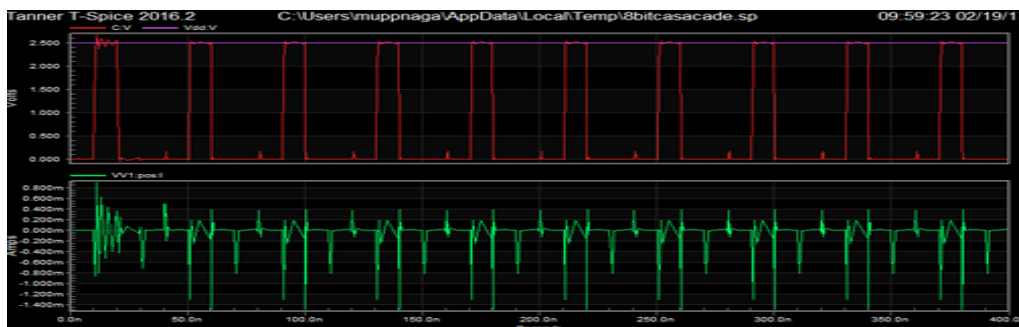


Fig. 16. Transient Analysis of 8-bit VpAn.

#### IV. ANALYSIS

An analysis was conducted, concentrating on the critical paths, the paths which require only certain combination of transistors of the circuit to operate, as stated in the previous sections of the paper, by running the transient analysis of those critical paths and recording their rise and fall times as done in [1]. Measurements were taken at the moment when the input signals finished changing (Vdd or 0V), and ending when the output came within  $V_{dd} * 10\%$  of the result (0.25V or 2.25V respectively).

Transistors were sized consistently for this portion of the experiment, all PMOS transistors had a width of 12u and a length of 0.25u. All NMOS transistors had a width of 4u and a length of 0.25u. The width of PMOS transistors is maintained thrice of the NMOS. The justification for the uniform sizing was to maintain that the only variable under examination for the delay study was the physical arrangement of the transistors themselves. Other factors affecting delay are not considered. Only the individual step delays and overall propagation delay of a discreet CLA unit are discussed.

When the delay values of all the 4-bit configurations are compared with each other, in this case NMOS orientation and PMOS data of changing paths play a major role. Unlike, the results in [1], in this case, VpAn is considered the best configuration with the better delay of 0.28ns.

It is apparent from Table 1 that the propagation delay of the 8-bit VpAn CLA is lower for all the changing paths, though the voltage driven is low. This has been possible by the optimized transistor sizing. The power dissipation of the 8-bit CLA has been observed to be very low as seen in Fig. 16

The 8-bit VpAn delay values are tabulated as follows:

TABLE I. SIMULATION DATA, 8-BIT VPAN CLA

Changing Signals		Delay (ns)	
		Falling C'	Rising C'
NMOS	G3	<b>0.28</b>	0.44
	G2/P3	0.39	0.39
	G1/P2/P3	0.58	<b>0.06</b>
	G0/P1/P2/P3	0.72	0.59
PMOS	G3/P3	<b>0.13</b>	0.34
	G3/G2/P2	0.84	0.45
	G3/G2/G1/P1	0.85	0.5
	G3/G2/G1/G0	0.80	<b>0.3</b>

It is apparent from Table 1 that the propagation delay of the 8-bit VpAn CLA is lower for all the changing paths, though the voltage driven is low. This has been possible by the optimized transistor sizing. The power dissipation of the 8-bit CLA has been observed to be very low as seen in Fig. 16.

#### V. TRANSISTOR SIZING

Transistor sizing was the main parameter that played a vital role in reducing the propagation delay and power dissipation. Initially the transistor dimensions from [1], i.e. PMOS with the width of 20u, length 2u and NMOS with the width of 10u, Length 2u were taken and then they were down sized step by step to PMOS with the width of 12u, length of 0.25u and then the NMOS width to 4u and length to 0.25u for enhanced yield. Majorly, diminishing the length of the transistor condensed the propagation delay to a large extent.

#### VI. FUTURE CONSIDERATIONS

Continuing buffer insertion for making 8-bit CLA adder, faster further reducing the transition time. Then, we would try to implement the CLA configuration by using multiplexers and look up tables. The goal is to extend the CLA configurations to 64-bit input circuits with high performance by cascading. Further, the layout of this circuit would be developed using Tanner L-edit tool.

The long-term plan is to extend this project with a CMOS sensor and then to fabricate the prototype of this design by submitting it to MOSIS. MOSIS is for "Metal Oxide Semiconductor Implementation Service," an integrated circuit foundry service, which acts as a media between the customer and fabrication foundries [6].

#### VII. CONCLUSION

In this paper, an 8-bit CLA adder using CMOS transistors is implemented, the transition time of all the critical paths of the circuit is evaluated, and also the current consumed across each path is observed and then the propagation delay and power dissipation of the circuit to a large extent are reduced, thereby increasing the performance of the CLA adder. It is expected that the performance gain of the CLA adder circuit for higher inputs would be sustained.

#### REFERENCES

- [1] Binggeli, M.; Denton, S.; Muppaneni, N.S.; Chiu, S, "Optimizing carry-lookahead logic through a comparison of PMOS and NMOS block inversions," 2015 IEEE International Conference on Electro/Information Technology (EIT), Dekalb, IL,2015, pp. 641 – 646.
- [2] Amuthavalli G.; Gunasundari R; "Analysis of 16-bit Carry Look Ahead Adder - A Subthreshold Leakage Power Perspective" www.arpn journals.com, Vol. 10, No. 6, 2015.
- [3] Amita.; Mrs. Nitin Sachdeva; "Design and Analysis of Carry Look Ahead Adder using CMOS Technique," IOSR journal of Electronics & Communication Engineering. ISSN: 2278-2834, Vol-9, issue 2[Mar-APR], 2014, pp.92-95.
- [4] P.L.Chen; "A Low-cost Carry look-Ahead Adder for Flying-Adder Frequency Synthesizer," 2016, IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Nantou, 2016, pp.1-2. Doi:10.1109/ICCE-TW.2016.7521070.
- [5] Jasbir Kaur.; Lalit Sood.; "Comparison Between Various Types of Adder Topologies "International Journal on Computer Science and Technology (IJCSST), Vol.6, Issue 1, Jan-Mar-2015.ISSN:2229-4333.
- [6] MOSIS.; "https://www.mosis.com/pages/about/working-with-mosis".
- [7] O. Coudert, Synopsis Inc., Mountain View, CA, USA; "Gate sizing for constrained delay/power/area optimization," IEEE Transactions on Very Large Scale Integration (VLSI) system, vol: 5, Issue4, pp.465-472.
- [8] M. Lehman and N. Burla," A Note on the Simultaneous – Carry Generation System for High-Speed Adders," Electronic Computers, IRE Transactions on, vol. EC-9, no 4, pp.510-510, December 1960.

- [9] G.N. Tobias, "Carry-Save Architectures for High-Speed Digital Signal Processing , " Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology , vol. 3, no. 1- 2,pp.121- 140, June 1991, doi: 10.1007/BF00927839.
- [10] J.F. Wakerley, Digital Design: Principles and Practices, 4<sup>th</sup> Ed. Upper Saddle River, New Jersey: Pearson Education, Inc., 2006.
- [11] F. Vahid, Digital Design: with RTL Design, and Verilog. 2<sup>nd</sup> Ed.Hoboken, New Jersey: John Wiley & Sons, Inc., 2011.
- [12] C. Nagendra, M. J. Irwin, and R. M. Owens, "Area-time-power tradeoffs in parallel adders, "Circuits and Systems II: Analog & Digital Signal Processing, IEEE Transaction on, Vol.43, No: 10, October 1996.