

Investigations of Carry Look Ahead Adder at $V_{DD}=1.0V$ at 65nm CMOS Technology for High Speed and Low Power Applications

D. Rajesh¹, S. Sowmya¹, M. Subashini¹, P. Suryaprabha¹, G. Naveen Balaji², A. Gautami³

¹UG Student, Department of ECE, SNS College of Technology, Coimbatore, India

^{2,3}Assistant Professor, Department of ECE, SNS College of Technology, Coimbatore, India

Abstract: - Carry look-ahead adder (CLA) is a fast adder. This paper is to accelerate the 4-bit CLA circuit. In this circuit, power been analysed for various size. Moreover, logic gates have less fan-in and fan-out and signal through one less MOS transistor in critical path. Both the static and dynamic design are being compared with power and size. Carry look ahead adder is designed in various size like 2um, 250nm, 125nm, 65nm in Tanner Tool.

Key words: Carry Look Ahead Logic, CMOS, Tanner EDA, Digital Circuits, logic gates.

I. INTRODUCTION

Addition is the basic mathematical operation and an essential operating in Algorithm Logic Units (ALU) [1]. Currently, addition have many logic designs, such as ripple adder, carry select adder and so on [2]. In these circuits, the carry-sum of higher bits depend on the carry-out of lower significant bits and this causes long propagation delay. Carry look-ahead adder (CLA) solves this problem. This method makes the carry-sum of each bit to be calculated by inputs directly. Some articles implement the logic gate in different technologies, such as BiCMOS [3], in order to speed up the CLA operation. And some fabricate the logic gate only in NMOS transistors [4]. Circuits in [5]-[7] are accomplished in CMOS logic gate. They are constructed by XOR AND and OR gates. All of them have the same formulas. However, the design [8] uses NAND gate to replace the AND and NOT gates. This can efficiently decrease the cost and increase the speed. Therefore [8] is the fastest CLA design at present, and it has lower cost in area and power.

Our work is to accelerate the CLA circuit established by dynamic CMOS logic gates. This simplification scheme realizes this acceleration and reduces power consumption.

In this paper, section I introduces the background about CLA. In section II, the principle of 4-bit CLA circuit is described in part A. Part B gives the new derived functions and analysis.

Comparisons of the simulation results are presented in section III. At last, a conclusion is given.

II. IMPLEMENTATION OF CARRY LOOK-AHEAD ADDER

A. The principle of 4-bit CLA

Function of the CLA is to calculate the carry-in of each bit from inputs directly. For addition with two inputs, the external carry-in is denoted as C_0 . Both inputs have 4 bits and they are represented as $A_3A_2A_1A_0$ and $B_3B_2B_1B_0$ respectively. Then sum of this addition $S_3S_2S_1S_0$ can be calculated by Full adders. Thus $S_3S_2S_1S_0 = A_3A_2A_1A_0 \oplus B_3B_2B_1B_0 \oplus C_3C_2C_1C_0$, where C_3 C_2 and C_1 are the carry-outs of previous bits.

The logic relationships between the carry-outs and inputs are expressed by equations (1)-(3). Among them, signals

Carry-Propagation (abbreviated P_i) and Carry-Generate (abbreviated G_i) are obtained $P_i = A_i \oplus B_i$ and $G_i = A_i \& B_i = A_i B_i$, in which is the value 0 1 2 or 3. The symbol '&' represents logic AND and it can be omitted in function sometimes. Besides, the symbol '+' denotes the logic OR.

$$P_i = A_i \oplus B_i$$

$$G_i = A_i B_i$$

The sum output and carry output can be expressed as

$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + P_i C_i$$

Where G_i is a carry generate which produces the carry when both A_i , B_i are one regardless of the input carry. P_i is a carry propagate and it is associate with the propagation of carry from C_i to C_{i+1} .

The carry output Boolean function of each stage in a 4 stage carry-Look ahead adder can be expressed as

$$C_1 = G_0 + P_0 C_{in}$$

$$C_2 = G_1 + P_1 C_1$$

$$= G_1 + P_1 G_0 + P_1 P_0 C_{in}$$

$$C_3 = G_2 + P_2 C_2$$

$$= G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{in}$$

$$C_4 = G_3 + P_3 C_3$$

$$= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{in}$$

Therefore, a 4-bit parallel adder can be implemented with the carry-Lookahead scheme to increase the speed of binary addition as shown in below figure. In this, two Ex-OR gates are required by each sum output. The first Ex-OR gate generates P_i variable output and the AND gate generates G_i variable.

Hence, in two gates levels all these P's and G's are generated. The carry-Lookahead generators allows all these P and G signals to propagate after they settle into their steady state values and produces the output carriers at a delay of two levels of gates. Therefore, the sum outputs S_2 to S_4 have equal propagation delay times. It is also possible to construct 16 bit and 32 bit parallel adders by cascading the number of 4 bit adders with carry logic. A 16 bit carry-Lookahead adder is constructed by cascading the four 4 bit adders with two more gate delays, whereas the 32 bit carry-Lookahead adder is formed by cascading of two 16 bit adders. In a 16 bit carry-Lookahead adder, 5 and 8 gate delays are required to get C_{16} and S_{15} respectively, which are less as compared to the 9 and 10 gate delay for C_{16} and S_{15} respectively in cascaded four bit carry-Lookahead adder blocks. Similarly, in 32-bit adder, 7 and 10 gate delays are required by C_{32} and S_{31} which are less compared to 18 and 17 gate delays for the same outputs if the 32-bit adder is implemented by eight 4 bit adders.

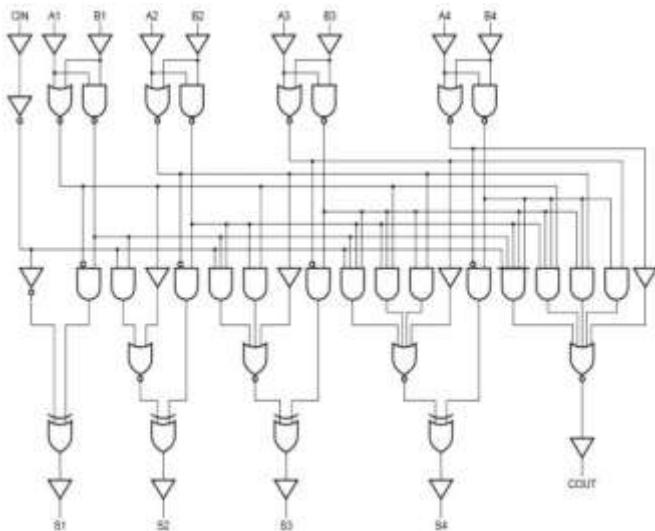


Fig.1 Carry Look Ahead Adder

III. STATIC LOGIC

The most widely used logic is complementary CMOS logic due to advantages associated with it like Low sensitivity to noise, Low power consumption with no static power dissipation, Good performance and Robustness. These properties lead to implementation of large fan in logic circuits using same devices. Static MOS circuits design includes complementary CMOS, ratio logic and pass transistor logic.[11]

IV. COMPLEMENTRY CMOS

Static CMOS gates are implemented by using combination of two networks, the pull up network (PUN) and pull down network (PDN). Static CMOS is characterized by very good current driving capabilities and high noise margins. In Static CMOS design, at every point in time, each gate output is connected to either V_{dd} or V_{ss} via a low-resistance path. Also, the outputs of the gate assume at all times the value of the Boolean function implemented by the circuit. A Static CMOS gate is a combination of two networks, the pull up network (PUN) and the pull down network (PDN). The function of the PDN is to provide a connection between the output and V_{dd} when the output of the logic gate is supposed to be 1. Similarly, the PDN connects the output to V_{ss} when the output is expected to be 0. The PUN and PDN networks are constructed in a mutually exclusive manner such that one and only one of the networks are conducting in steady state. The Static CMOS gates have rail-to-rail swing, no static power dissipation. The speed of the static CMOS circuit depends on the transistor sizing and the various parasitic that are involved with it. The problem with this type of implementation is that for N fan-in gate 2N number of transistors are required, i.e., more area required to implement logic. This has an impact on the capacitance and thus the speed of the gate.[13]

V. DYNAMIC LOGIC

In integrated circuit design, dynamic logic (or sometimes clocked logic) is a design methodology in combinatory logic circuits, particularly those implemented in MOS technology. It is distinguished from the so-called static logic by exploiting temporary storage of information in stray and gate capacitances. It was popular in the 1970s and has seen a recent resurgence in the design of high speed digital electronics, particularly computer CPUs. Dynamic logic circuits are usually faster than static counterparts, and require less surface area, but are more difficult to design. Dynamic logic has a higher toggle rate than static logic [2] but the capacitance loads being toggled are smaller[3] so the overall power consumption of dynamic logic may be higher or lower depending on various trade-offs. When referring to a

particular logic family, the dynamic adjective usually suffices to distinguish the design methodology, e.g. dynamic CMOS or dynamic SOI design. Dynamic logic is distinguished from so-called static logic in that dynamic logic uses a clock signal in its implementation of combinational logic circuits. The usual use of a clock signal is to synchronize transitions in sequential logic circuits. For most implementations of combinational logic, a clock signal is not even needed.

The static/dynamic terminology used to refer to combinatorial circuits should not be confused with how the same adjectives are used to distinguish memory devices, e.g. static RAM from dynamic RAM.

VI. STATIC VERSUS DYNAMIC LOGIC

The largest difference between static and dynamic logic is that in dynamic logic, a clock signal is used to evaluate combinational logic. However, to truly comprehend the importance of this distinction, the reader will need some background on static logic. In most types of logic design, termed static logic, there is at all times some mechanism to drive the output either high or low. In many of the popular logic styles, such as TTL and traditional CMOS, this principle can be rephrased as a statement that there is always a low-impedance DC path between the output and either the supply voltage or the ground. As a sidenote, there is of course an exception in this definition in the case of high impedance outputs, such as a tri-state buffer; however, even in these cases, the circuit is intended to be used within a larger system where some mechanism will drive the output, and they do not qualify as distinct from static logic. In contrast, in dynamic logic, there is not always a mechanism driving the output high or low. In the most common version of this concept, the output is driven high or low during distinct parts of the clock cycle. During the time intervals when the output is not being actively driven, its impedance causes it to maintain a level within some tolerance range of the driven level.

Dynamic logic requires a minimum clock rate fast enough that the output state of each dynamic gate is used or refreshed before the charge in the output capacitance leaks out enough to cause the digital state of the output to change, during the part of the clock cycle that the output is not being actively driven. Static logic has no minimum clock rate—the clock can

be paused indefinitely. Being able to pause a system at any time for any duration can also be used to synchronize two asynchronous events. (While there are other mechanisms to do this, such as interrupts, polling loops, processor idling input pins [like RDY on the 6502], or processor bus cycle extension mechanisms such as WAIT inputs, using hardware to gate the clock to a static-core CPU is simpler, is more temporally precise, uses no program code memory, and uses almost no power in the CPU while it is waiting. In a basic design, to start waiting, the CPU would write to a register to set a binary latch bit which would be ANDed or ORed with the processor clock, stopping the processor. A signal from a peripheral device would reset this latch, resuming CPU operation.[12]

In particular, although many popular CPUs use dynamic logic[citation needed], only static cores—CPUs designed with fully static technology—are usable in space satellites owing to their higher radiation hardness. Dynamic logic, when properly designed, can be over twice as fast as static logic. It uses only the faster N transistors, which improve transistor sizing optimizations. Static logic is slower because it has twice the capacitive loading, higher thresholds, and uses slow P transistors for logic.[22]

Dynamic logic can be harder to work with, but it may be the only choice when increased processing speed is needed. Most electronics running at over 2 GHz these days require the use of dynamic, although some manufacturers such as Intel have completely switched to static logic to reduce power consumption. Note that reducing power use not only extends the running time with limited power sources such as batteries or solar arrays (as in spacecraft), but it also reduces the thermal design requirements, reducing the size of needed heatsinks, fans, etc., which in turn reduces system weight and cost. In general, dynamic logic greatly increases the number of transistors that are switching at any given time, which increases power consumption over static CMOS. There are several power saving techniques that can be implemented in a dynamic logic based system. In addition, each rail can convey an arbitrary number of bits, and there are no power-wasting glitches. Power-saving clock gating and asynchronous techniques are much more natural in dynamic logic.[23][25]

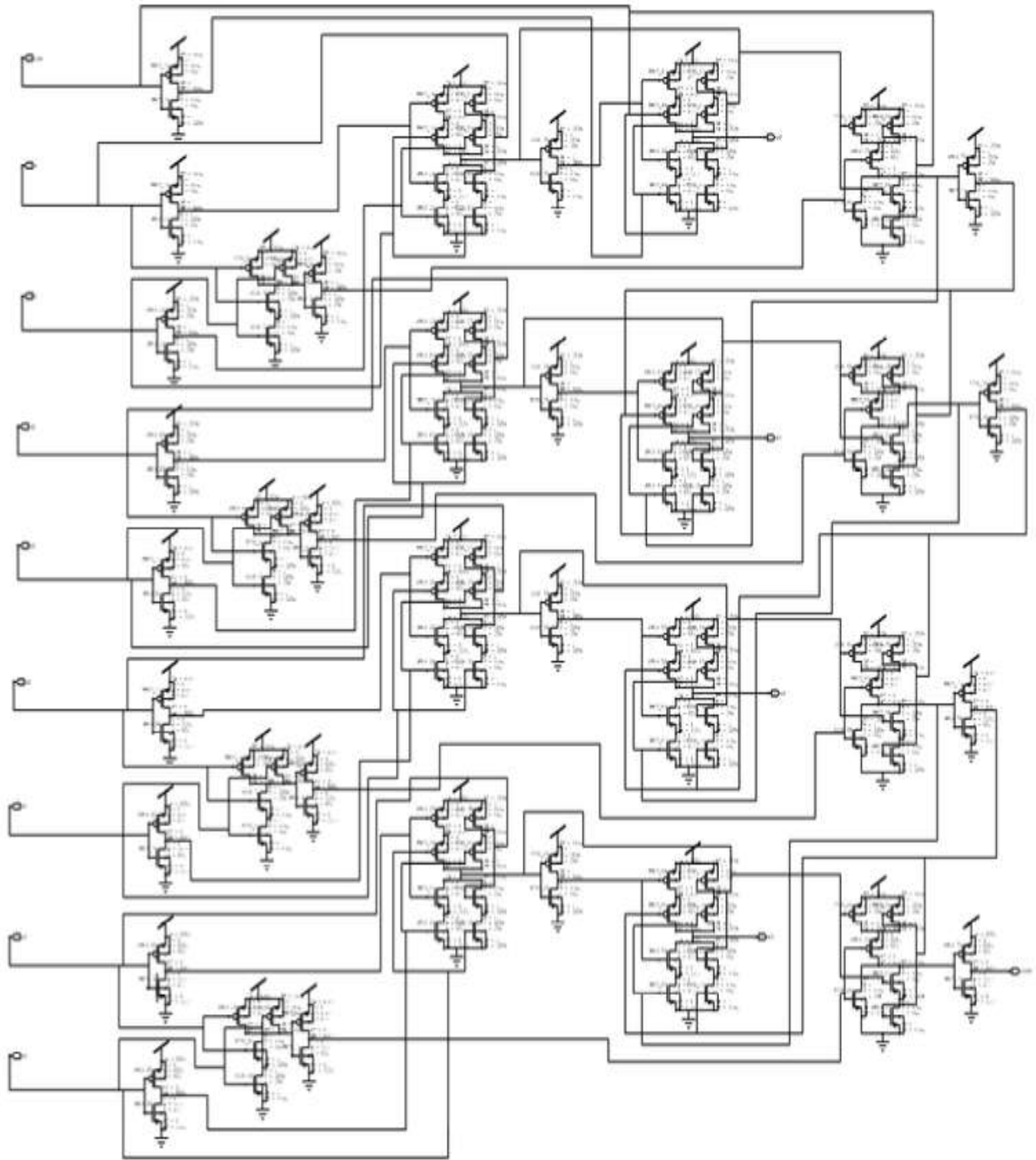


Fig.2 Carry Look Ahead Adder Static Logic

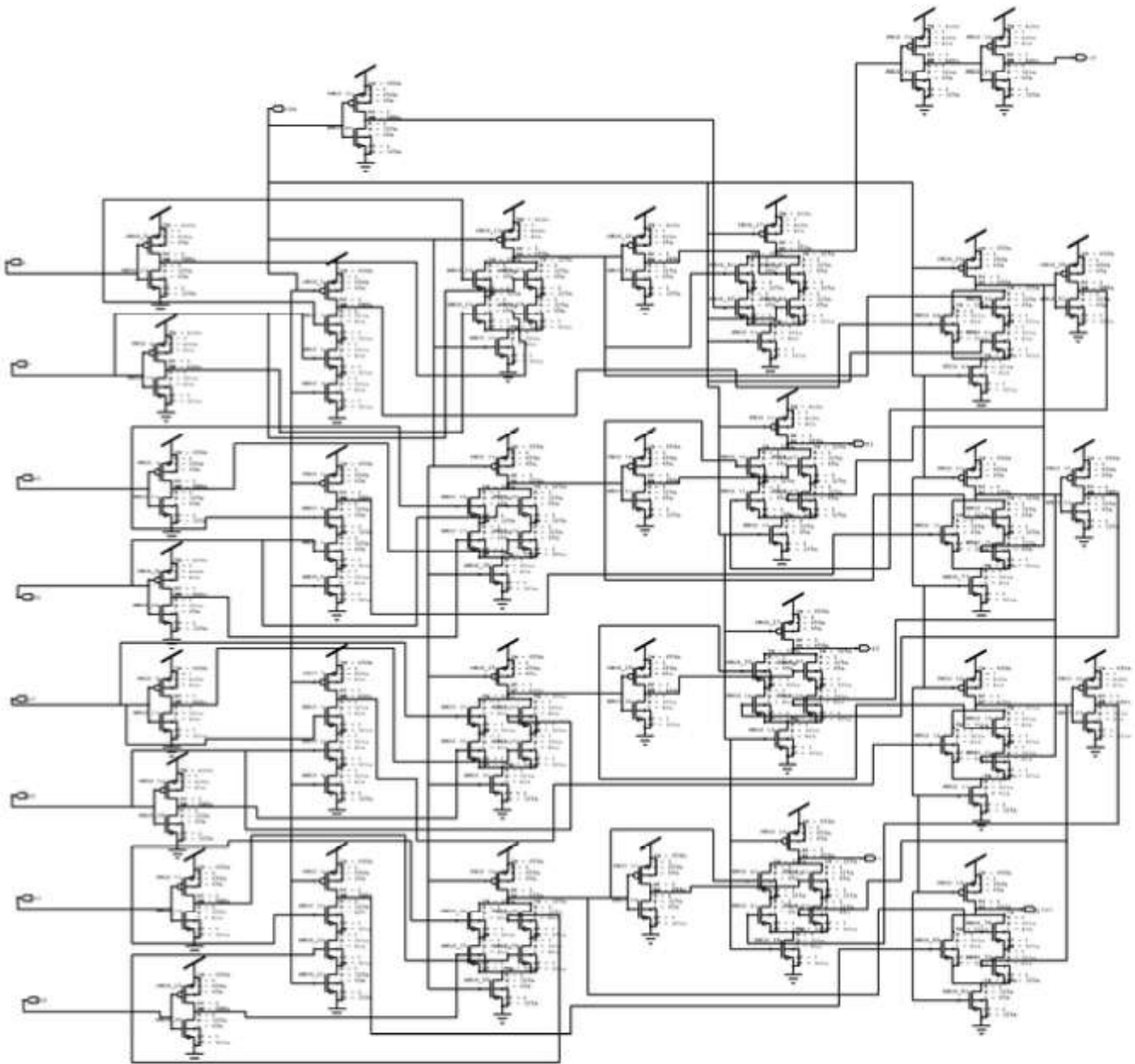


Fig.3 Carry Look Ahead Adder Dynamic Logic

TABLE I POWER COMPARISON CLAA

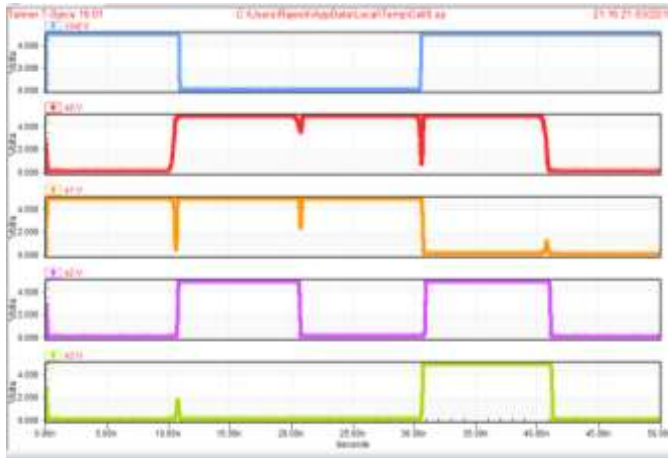
Size (nm)	Max Power (W) at 3.79×10^{-11} s	Average Power (W)
65	1.27×10^{-002} at 3.79×10^{-11} s	7.53×10^{-003}
125	2.93×10^{-001} at 1.51×10^{-008} s	9.45×10^{-003}
250	9.67×10^{-002} at 4.20×10^{-011} s	8.67×10^{-003}
2000	4.93×10^{-001} at 3.70×10^{-011} s	1.19×10^{-002}

The power is maximum and average power is compared based on the time. [19] The no. of transistors in static design of carry look ahead adder is 146 and in dynamic design 120. The result for the sum and carry is based on the inputs $A_0, A_1, A_2, A_3, B_0, B_1, B_2, B_3$ and C_{in} then the bit streams for the inputs[24]

A_0	00111	B_0	11100	
A_1	10000	B_1	11001	
A_2	11000	B_2	00011	
A_3	11111	B_3	01100	C_{in} 10101

The resultant waveform for the bit stream is shown in the Fig. 4

VII. RESULT



REFERENCES

- [1] Morrison M, Lewandowski M, Meana R, et al. Design of a novel reversible ALU using an enhanced carry look-ahead adder Nanotechnology (IEEE-NANO), 2011 11th IEEE Conference on. IEEE, 2011: 1436-1440.
- [2] Knowles S. A family of adders Computer Arithmetic, 2001. Proceedings. 15th IEEE Symposium on. IEEE, 2001: 277-281 [3]
- [3] Kuo J B, Liao H J, Chen H P. ABiCMOS dynamic carry lookahead adder circuit for VLSI implementation of high-speed arithmetic unit[J]. IEEE journal of solid-state circuits, 1993, 28(3): 375-378.
- [4] Hwang B K. Carry lookahead adder: U.S. Patent 5,951,631[P]. 1999-9-14.
- [5] Mano M M, Kime C R, Martin T. Logic and computer design fundamentals[M]. Prentice Hall, 2008.
- [6] Preethi K, Balasubramanian P. FPGA implementation of synchronous section-carry based carry look-ahead adders Devices, Circuits and Systems (ICDCS), 2014 2nd International Conference on. IEEE, 2014: 1-4.
- [7] Singh R P P, Kumar P, Singh B. Performance analysis of 32-bit array multiplier with a carry save adder and with a carry-look-ahead adder. International Journal of Recent Trends in Engineering, 2009, 2(6): 83-86.
- [8] Pai Y T, Chen Y K. The Fastest Carry LookaheadAdder, IEEE International Workshop on Electronic Design, Test and Applications. IEEE Computer Society, 2004:434.
- [9] Jan M. Rabaey Anantha Chandrakasan, Borivoje Nikolic, "Digital Integrated Circuits: A Design Perspective", Second Edition, Prentice Hall, 2003.
- [10] Jia Miao and Shuguo Li "A Novel Implementation of 4-bit Carry Look-ahead Adder"
- [11] Wenbin Xu, Sachin S. Sapatnekar, Jiang Hu, "A simple yet efficient accuracy configurable adder design", Low Power Electronics and Design (ISLPED 2017 IEEE/ACM International Symposium on, pp. 1-6, 2017.
- [12] G. Naveen Balaji, D. Rajesh "Linear Assembly in ATPG: Applications in Fast SCAN – BIST VLSI Circuits" International

- Journal of Innovations in Engineering and Technology, Vol. 6, Issue 3 (Feb 2016) pp: 101-106, ISSN: 2319-1058
- [13] R. ArunSekar, G. Naveen Balaji, A. Gautami, B. Sivasankari "High Efficient Carry Skip Adder in various Multiplier Structures" Advances in Natural and Applied Sciences (Annexure II), Vol. 10 Issue 14 (Special) (Oct 2016) pp: 193-197, ISSN: 1995-0772
- [14] M. Srinivasaperumal, K. Boopathi Raja, G. Naveen Balaji, E. Christina Dally "Concurrent Node Recovery From Failure In Wireless Sensor-Actor Networks" Advances in Natural and Applied Sciences (Annexure II), Vol. 10 Issue 17 (Dec 2016) pp: 240-246, ISSN: 1995-0772
- [15] G. Naveen Balaji, R. Prabha, E. Shanthini, J. Jayageetha, Mohand Lagha "Rapid low power Synchronous circuits using transmission gates" Advances in Natural and Applied Sciences (Annexure II), Vol. 10, Issue 17 (Dec 2016) pp: 287-291, ISSN: 1995-0772
- [16] G. Naveen Balaji, S. Chenthur Pandian, D. Rajesh "Fast Test Pattern Generator using ATALANTA M 2.0" Asian Journal of Research in Social Sciences and Humanities (Annexure I) Vol. 7 No. 2 (Feb 2017) pp. 721-729 ISSN: 2249-7315 DOI: 10.5958/2249-7315.2017.00124.1
- [17] G. Naveen Balaji, V. Aathira, K. Ambhikavathi, S. Geethiga, R. Havin "Low Power and High Speed Synchronous Circuits using Transmission Gates" Asian Journal of Research in Social Sciences and Humanities (Annexure I), Vol. 7 No. 2 (Feb 2017) pp. 713-720. ISSN: 2249-7315, DOI: 10.5958/2249-7315.2017.00123.X
- [18] G. Naveen Balaji, S. Chenthur Pandian, D. Rajesh "High Performance Triplex Adder Using CNTFET" International Journal of Trend in Scientific Research and Development, Vol.1, No. 5, pp: 368-373, ISSN 2456 - 6470
- [19] G. Naveen Balaji, S. Chenthur Pandian, S. Giridharan, S. Shobana, J. Gayathri "Dynamic and Non-Linear Charge Transfer Through Opto-Deporation by Photovoltaic Cell" International Journal of Trend in Scientific Research and Development, Vol. 1, No. 5, pp: 486-492, ISSN 2456 - 6470
- [20] G. Naveen Balaji, S. Karthikeyan, M. Merlin Asha " 0.18µm CMOS Comparator for High-Speed Applications" International Journal of Trend in Scientific Research and Development, Vol. 1, No. 5, pp: 671-674, ISSN 2456 - 6470
- [21] G. Naveen Balaji, K. Saravanan, R. Poorani, T. Vishnu Priya, R. Reka Raj "Advanced Security System using PIC through Bluetooth" International Journal of Trend in Scientific Research and Development, Vol. 1, No. 5, pp: 675-685, ISSN 2456 - 6470
- [22] G. Naveen Balaji, N.V. Hari Suriya, S. Anand Vikash, S. Arun Kumar, R. Arun, "Gasoline Purity Meter Using Peripheral Interface Controller for Automobile Applications" International Journal of Engineering and Technical Research (IJETR), Vol. 7, No.10, pp:46-55, ISSN: 2321-0869
- [23] G. Naveen Balaji, S.Chenthur Pandian, "Design for Testability of Kipbond Logic" "Perspectivas em Ciência da Informação" (Annexure - I), School of Information Science of the Federal University of Minas Gerais (UFMG), Vol. 22, No. SP.01, pp: 261-284, ISSN 1413-9936
- [24] Gokilavani.C, Naveen Balaji.G, "Analysis And Implementation Of Mean, Maximum And Adaptive Median For Removing Gaussian Noise And Salt & Pepper Noise In Images", International Journal of Innovations in Engineering and Technology (IJET), Vol. 9, No. 3, pp: 34-38, ISSN: 2319-1058
- [25] G. Naveen Balaji, S. Chenthur Pandian, "Design of test pattern generator (TPG) by an optimized low power design for testability (DFT) for scan BIST circuits using transmission gates", Cluster Computing, The Journal of Networks, Software Tools and Applications, DOI: 10.1007/s10586-018-2552-x, ISSN: 1386-7857 (Impact Factor: 2.040)