Indian Journal of Natural Sciences www.tnsroindia.org.in ©IJONS

Vol.14 / Issue 78 / June / 2023 *International Bimonthly (Print) – Open Access* *ISSN: 0976 – 0997*

RESEARCH ARTICLE

Implementation of Energy Efficient Hybrid Adder for Approximate Computing Applications

Telugu Maddileti[1], M.Jagadeesh Chandra Prasad[2], Aaradhana[3]* and K.Maheswari Devi[4]

[1]Associate Professor, Department of ECE, Malla Reddy Engineering College, Secunderabad, Hyderabad, Telangana 500100, India.
[2]Professor, Department of ECE, Malla Reddy Engineering College, Secunderabad, Hyderabad, Telangana 500100, India.
[3]PG Scholar, Department of ECE, Malla Reddy Engineering College, Secunderabad, Hyderabad, Telangana 500100, India.
[4]Assistant Professor, Department of ECE, Malla Reddy Engineering College, Secunderabad, Hyderabad, Telangana 500100, India.

Received: 20 Feb 2023 Revised: 20 Apr 2023 Accepted: 30 May 2023

***Address for Correspondence**
Aaradhana
PG Scholar,
Department of ECE,
Malla Reddy Engineering College,
Secunderabad, Hyderabad,
Telangana 500100, India.
E.Mail: aaradhana543@gmail.com

ABSTRACT

As the dimensions of transistors have shrunk to sub-micron scales, new challenges, including power efficiency and undetected radiation hazards, have emerged as important areas of concern in the realm of extremely large-scale integration. Because they are key areas of worry, these new problems have recently come to the forefront as important areas of concern. In addition, complementary metal oxide semiconductors, which are more often referred to as CMOS transistors, have a greater consumption of space, power, and delay than the various other kinds of transistors. As a consequence of this, a brand-new 10 Transistor (10T) Multiplexer Logic based Full Adder (MLFA) has been created and is now being simulated. Utilizing both kinds of transistors was necessary in order to achieve this goal, which is considered one of the possible answers to the problems caused by standard CMOS. Additionally, the MLFA that is being presented functions on the basis of a multiplexer selection logic, which has the capability of effectively reducing the route delays as well as the number of transistors. In addition, employing suggested MLFAs allowed for the development of a 4-bit ripple carry adder (RCA). In the end, the simulations are performed with the help of the Tanner-EDA simulation programme. The results

of the simulation indicated that the recommended approach resulted in superior performance when compared to other approaches that are presently regarded to be state of the art in the field.

Keywords: Full adder, Multiplexer, simulation, CMOS, RCA, Tanner EDA etc.

INTRODUCTION

Conventionally, computing systems are built to function with the highest feasible degree of accuracy. However, this tendency is met with significant obstacles in terms of technology, such as high performance and power consumption, as well as dependability of circuits. Computer systems have been gradually improving in both their performance and their power consumption for well over half a century, with scaling technologies being the primary factor in this improvement [1]. It is anticipated that this pattern will maintain its prevalence far into the foreseeable future. In accordance with Dennard's scaling, the size of a transistor has been noticeably smaller during the course of its development, and the supply voltage has likewise become lower. As a direct consequence of this development, electronic circuits can now function at higher frequencies while nearly preserving the same amount of power dissipation. However, because Den nard's scaling is drawing ever-nearer to its end, it is becoming increasingly difficult to achieve additional performance improvements while adhering to the same 10.1109/JPROC.2020.2975695 power limits. This is because Den nard's scaling is getting closer and closer to its conclusion. Consumption of electricity has been a major source of concern for a very long time, and in recent years, it has developed into a problem that impacts the whole industry as a whole and is of critical importance. This is because it is difficult to monitor and avoid parameter fluctuations and defects at advanced nanoscales, which is the reason why this is the case [2]. As a consequence of this, there will be a significant increase in the costs associated with the manufacturing and verification processes in order to ensure the success of the complete accuracy of the signals, logic values, devices, and interconnects [3].

When it comes to the creation of digital processing devices, with an emphasis on portable systems, the key goals are to lower the amount of power that is needed while concurrently improving the processing speed[4]. In most cases, increasing the speed of the precise processing unit results in an increase in the amount of power that is required to run [5]. The accuracy of the calculation is compromised as one of the strategies for increasing processing power without sacrificing speed. It is possible to utilise this method, which is known as approximate computing, for the applications in which it is acceptable to have some inaccuracies [6]. Among these are the applications in which digital signal processing, often known as DSP, is carried out on the signals connected to human senses. The capabilities of human perception are restricted, thus most of the time; approximation computing is used for processing these signals in bespoke DSP blocks. This is done since it is more accurate than human perception [7]. The realisation of this component utilising the technique of approximation computing was motivated by these facts, which can be found in the previous sentence. Previous studies on approximation adders have focused on either reducing the error weight or the error probability, both of which are considered to be more broad techniques. This is because, in most cases, the majority of the operations take place in the LSB section. The second method makes use of structures that are pure approximation adders. The primary design goals for these adders are to cut down on the amount of power they use and the amount of delay they incur, as well as the error probability associated with the summing. They could also be accompanied with an error correcting unit, which incurs overhead costs in terms of both time and power consumption [8].

The ECG signals coming from a patient's body are continually monitored by a device based on WBSN. A continuous monitoring result in the production of an excessive amount of data, which, in turn, calls for a greater quantity of storage space, results in higher expenses associated with transmission, and raises the required amount of power consumption. In addition, in order for the Processing Subsystem of the WBSN device to be able to deal with this enormous volume of data, it needs additional processing time to be carried out, which results in an extraordinarily

high level of power consumption. As a result, data compression methods are routinely applied in order to accomplish sparse encoding of this data. Sparse encoding has the effect of lowering the needs for processing, which in turn leads to a reduction in the amount of power that is lost by the Processing subsystem[9]. It is possible to reduce the amount of data being transferred by cutting down on the number of bits being sent (which are what make up the packets) [10]. This is due to the fact that the reconstructed signals are not exactly the same as the original signals.

Literature Survey

Betzel *et al*. [11] highlighted the potential advantages of approximation computing for the reduction of communication by conducting a review of three possible strategies for approximate communication. These techniques were compression, relaxed synchronization, and value prediction. A comparison of the methods is made using an assessment methodology that takes into consideration factors such as the reduction of communication costs, performance, energy reduction, applicability, overheads, and output deterioration. In conclusion, this essay offers a number of recommendations for further research on approximation communication methods.

Mahmud and colleagues presented a thorough assessment with connection to the processing and analysis of large data of the procedures and techniques of data partitioning and sampling[12]. The first thing that students do when they begin this course is get an overview of the most widely used big data frameworks that can be operated on Hadoop clusters. Following that, we will go through the fundamental approaches to data partitioning, which will include a discussion of the three most frequent horizontal partitioning schemes: range, hash, and random partitioning. Following that, we shall go to the subsequent part of this discussion. The topic of data partitioning on Hadoop clusters is also discussed, along with an overview of novel methodologies for the partitioning of large amounts of data, such as the recently developed Random Sample Partition (RSP) distributed model. In addition, the topic of data partitioning on Hadoop clusters is also discussed. In addition, the topic of data partitioning on Hadoop clusters is covered.

Wei *et al*. [13] suggested a novel method of using static analysis for the purpose of performing security checks on Android applications, as well as a generic framework known as Amandroid. In an Android app component, Amandroid will do data flow and data dependency analysis on the component in addition to determining points-to information for every objects inside the component in a flow- and context-sensitive manner (which is configured by the user). Additionally, Amandroid monitors the activities of inter-component communication. In order to undertake intra-app or inter-app analysis, it is able to stitch together information at the component level and information at the app level.

Stanley-Marbell *et al*. [14] made the observation that it could be useful to provide an overview of the research results on computer systems that only produce as many mistakes as their end-to-end applications can tolerate. This action was taken as a direct reaction to a suggestion made by the previously stated organisation. [There must be other citations for this] The results include a broad spectrum of academic disciplines, such as information theory, computer-aided circuit design, digital system design, computer architecture, programming languages, and operating systems, to mention just a few of them. Instead of over-provisioning the resources that are controlled by each of these layers of abstraction to prevent errors, it is more efficient to take advantage of the masking of errors that are occurring at one layer in order to stop those errors from propagating to a higher layer. This is because errors that occur at one layer can be masked by errors that occur at higher layers. This is due to the fact that faults that take place at one layer might be concealed by errors that take place at a higher layer. This is because errors that occur at one layer can be masked by errors that occur at higher layers. This is achieved by eliminating the possibility of mistakes arising at a more fundamental level in the first place.

An edge-computing-based lightweight block chain framework (ECLB) was suggested for mobile devices by Liu *et al*. [15]. In order to achieve greater performance, this work presents a unique set of ledger structures and implements a transaction consensus procedure. In addition, taking into consideration the permission block chain environment, we

make explicit use of various cryptographic approaches in order to create a pluggable transaction regulation module. In conclusion, the results of our security analysis and performance evaluation demonstrate that ECLB is capable of maintaining the same level of security as block chains similar to Bit coin while achieving superior performance in terms of the cost of ledger storage in mobile devices, the cost of computing blocks, the throughput of transactions, the confirmation latency of transactions, and the transaction regulation cost.

zhou, *et al*. [16] created a Java annotation-based offloading framework for android mobile devices and gave it the name MCAF. This framework was created with the intention of easing the process of developing android apps that have the potential to offload their processing to a remote server. The only thing that the developers need to do is import the SDK library of our MCAF, and then they need to annotate the methods that need a lot of calculation. MCAF has the capability to automatically build the code that will be executed in the cloud as well as extract the annotated source code. In addition, the programmes that make the judgements about offloading are automatically integrated into the code that was originally written. Additionally, they carried out the actual trials in order to demonstrate that our MCAF is applicable.

The Wireless Body Sensor Nodes was a concept that was suggested by Gosh *et al*. [17]. (WBSN). These wireless body sensor networks (WBSNs) initially comprise of bio-sensors that collect signals from a patient's body and wireless transmitters that transfer the acquired data to a server that is located in either a private or public cloud. The collected data may then be analysed. These WBSNs are equipped with the appropriate hardware to process signals before sending them to the cloud in order to be stored there. In energy-constrained WBSNs, the simultaneous occurrence of all of these actions results in a large level of power consumption, which, in turn, shortens the operational lifetime of these networks. The vast majority of these data that are being sent to the servers are, by their very nature, duplicated; as a consequence, the therapeutic value that they have is negligible. This is because error-resilience is an inherent property of signal processing algorithms. In conclusion, the findings of the experiments reveal a 96% increase in system-level energy efficiency with just a 2% reduction in the effect on signal quality.

Liu, *et al*. [18] suggested an examination of approximation computing from both its past and its potential future. The size of a transistor has been greatly reduced from the beginning in line with Den nard's scaling, and the supply voltage has been decreased during the course of the years. As a direct consequence of this development, electronic circuits can now function at higher frequencies while nearly preserving the same amount of power dissipation. Not only does an increase in power consumption occur when the feature size of complementary metal-oxide-semiconductor (CMOS) technology is reduced to 7 nanometres, but an improvement in reliability also occurs at this point. This is due to the fact that it is more difficult to regulate and steer clear of parameter changes and flaws at advanced nanoscales.

It was thought that a reverse carry propagate adder (RCPA)should be used, as stated by Pashaeifar *et al*. [19]. Even if there are variations in the latency, the overall system will be more stable if you use this sort of carry propagation at the beginning of the process. We describe three distinct implementations of the reverse carry propagate full-adder (RCPFA) cell; each of these implementations differs from the others in terms of the amount of delay, power, and energy it consumes, as well as the accuracy it provides. It is possible to produce hybrid adders with varied degrees of accuracy by combining the structure that has been described with an accurate carry adder that works in reverse. Using a variety of different permutations of these structures allowed for the generation of these adders. The discrete cosine transforms (DCT) block of the JPEG compression and the finite-impulse response (FIR) filter applications are analysed in order to determine whether or not the recommended RCPAs are successful. This is the last but not the least of the topics covered.

Marchisio *et al*. [20] presented their idea of using deep learning for edge computing. In the beginning, as DNNs increase in complexity, the linked issue of their high energy consumption becomes a difficult one to deal with. This task is made much more difficult by edge computing, in which the computing devices have restricted access to resources and must function within a limited energy budget. As a direct result of this, it is necessary to carry out

specific optimizations for deep learning on both the software and the hardware levels. In this work, we undertake a complete analysis of the current trends of such optimizations, and we address key outstanding research concerns both in the medium term and in the long term.

Proposed Method

Because full adders are the fundamental components of any integrated circuit, their design and development must be approached with extreme caution in order to get optimal results. Conventional CMOS full adders have much greater power, delay, and energy consumption figures than their digital counterparts. As a result, the alternative technologies of FinFET and GnrFET are the technologies that may give reduced resource utilisation in lieu of the fundamental CMOS technology. In this part, an in-depth discussion of the creation of a full adder via the use of FinFET and GnrFET technology, together with 10T modelling, is presented. The MLFA is shown in the block diagram seen in figure 1. It demonstrates the operation of adding one bit by making use of a number of different components, such as XOR gates, NOT gates, and a Multiplexer 2 to 1 (MUX21), in that order. The functions of the MLFA are mostly focused on the switching operations for MUX21. Because of its performance, which is characterised by the rapid triggering of data in comparison to that of other building blocks, the computational complexity of the MUX21 is very low in this instance. In addition to that, the logic of FA is satisfied by this activating procedure. In this setup, each individual component is crafted by the use of the FinFET and GnrFET technologies, in that order. In addition, both the process of sum generation and the process of carry out generation are shown in Table 1. The fact that the lines in Figures 1 as well as Table 1 all have different colours indicates that they represent the same procedure.

In the beginning, inputs A and B are placed into the XOR gate, which ultimately produces the half adder sum output. This is the outcome of the first step. The output of this generation is sent as the data input to the MUX21 once it has been processed (input 0). Additionally, the XOR gate is used on the NOT gate, which leads in the output being referred to as XNOR. This is because of the way in which the XOR gate is utilised on the NOT gate. In addition to this, the result of the XNOR operation is fed into the MUX21 as the data input (input 1), and the MUX21 utilises C in as its selection input in the appropriate manner. In addition, depending on the value of C in, MUX21 will choose the data inputs using either the XOR or XNOR outputs. This is shown by the equations (2) and (3) that are presented below, in that order. The total adder sum, which is represented by the symbol S, is found in equation (1), which can be found below.

$S = A \oplus B \oplus C_{in}$ (1)

$C_{in} = 0 \rightarrow S = A \oplus B \oplus 0 \rightarrow A \oplus B$ (2)

$C_{in} = 1 \rightarrow S = A \oplus B \oplus 1 \rightarrow A \odot B$ (3)

In a similar fashion, the result of the XNOR operation is fed into the second MUX21 as the selection input, with C in serving as "data input 0" and B being the "data input 1." This, in turn, produces the carry out (C out).In addition, Equation 4 demonstrates the fundamental process of MLFA carry out, and it is altered in accordance with the XOR-XNOR logic for implementations that are based on multiplexers.

$C_{out} = AB + C_{in}(A \oplus B) = C_{in}(A \oplus B) + \bar{B}(A \odot B)$ (4)

Additionally, equations (5) and (6) illustrate the method of carry out generation in the following manner:

$A \odot B = 0 \rightarrow C_{out} = C_{in}$ (5)

$A \odot B = 1 \rightarrow C_{out} = B$ (6)

The varied colours in Figure 2 and Table 1, respectively, reflect how these equations are used in practise. This can be seen in both of these figures.

Figure 2 illustrates the FinFET modelling of the proposed MLFA that has been developed. The proposed MLFA is constructed from 5 numbers of FinFET PMOS transistors and 5 numbers of FinFET NMOS transistors, each in their own separate 5 number configuration. Due to the fact that the NMOS and PMOS transistors used in this particular implementation are carbon copies of one another, it is essential to include an equilibrium state into the MLFA. This is done in order to ensure that the unbiased electron-hole pair is both regulated and synchronised. In addition to this, the condition of balance also helps to maintain the appropriate level of power consumption while simultaneously improving energy efficiency. The combination of P1 and N1 FinFET transistors performs the function of an inverter, and it gives the value B minus as the output for any value of B that is input. Therefore, the combination of P2 and N2 FinFET transistors functions as an XOR gate, taking the values B and A as its inputs and producing the values A and B as its outputs.

In addition, as an inverter, the combination of P3 and N3 FinFET transistors causes them to produce the XNOR result AB in response to the XOR input AB. AB is the output of the XNOR operation. This is because the result of performing the XOR operation on AB is AB. In addition, the functionality of a MUX21 is accomplished by combining P4 and N4 FinFET transistors in a single device. The letter C in represents the selection input, data input-0 represents AB, data input-1 also represents AB, and the letter S represents the sum output. This specific arrangement of transistors is referred to as a MUX21 and has been given that name. The MUX21 that is generated by the combination of P5 and N5 FinFET transistors operates in the following manner: A and B are used as the selection input, data input-0 is used as C in, data input-1 is used as B, and the carry output is created as C out, respectively. This brings us to our final point, which is that the combination of these transistors creates a carry output. The implementation of a MLFA using GnrFETs makes use of the same transistor level circuit as before, and the operation remains the same.

RESULTS AND DISCUSSIONS

For the creation of each and every MLFA design, The Xilinx ISE software was the one that was used. This piece of software has the capacity to generate two separate kinds of outputs, namely simulation and synthesis. These outputs are both possible. The results of the simulation make it possible to conduct an in-depth analysis of the MLFA architecture with relation to the many permutations of input and output byte levels. A simple decoding technique may be approximated by applying a large number of different combinations of inputs and watching a wide variety of outputs while doing a simulation study of accurate encoding. As a consequence of the conclusions of the synthesis, the use of space in proportion to the number of transistors will be carried out. In addition, a time summary will be obtained with reference to the various route delays, and a power summary will be prepared making use of the static and dynamic power consumption. Both of these summaries will be gathered. Both of these summaries will be done. The results of the simulation of the existing MLFA are shown in Figure 3. Here, the outputs are wrong. Further, the results of the simulation of the proposed MLFA are shown in Figure 4.Table 2 presents a comparison and contrast of the results of performance assessments conducted on a number of different MLFA controllers. In this instance, the proposed MLFA resulted in superior (reduced) performance in terms of transistors, time-delay, and power consumption when compared to conventional approaches such EFA [9], 16T-FA [17], 12T -FA [16], and 11T-FA [13]. This was the case because the MLFA used fewer transistors than the conventional approaches. This was the case because the MLFA was able to reduce the time-delay without sacrificing the number of transistors.

CONCLUSION

This academic article focuses mostly on the design and manufacturing of a MLFA as well as a 4-bit RCA by making use of FinFET and GnrFET technologies. This is the primary topic of the study. It is engaged between the XOR results and the XNOR outcomes so that the entire output is created. In addition, a second multiplexer enabled the data inputs and produced the carry output, both of which contributed to the successful reduction of the route delays. The suggested MLFA is constructed on the basis of the logic used by the multiplexer selection, and as a result, it is activated between those two possibilities in order to provide the sum output. In addition to this, the technique that

was proposed was able to reduce the amount of power that was used by activities that were quickly activated. These libraries were used to create the final product. The results of the simulation make it possible to reach the conclusion that the design that was supplied is an excellent choice for a broad variety of addition-based applications. This conclusion can be reached because the findings of the simulation enable it. In addition to this, this strategy is extensible, meaning that it is used to build the complete the subtractor and the multiplication operations in the order given.

REFERENCES

1. Chiluveru, Samba Raju, *et al*. "Memory efficient architecture for lifting-based discrete wavelet packet transform." IEEE Transactions on Circuits and Systems II: Express Briefs 68.4 (2020): 1373-1377.
2. Hasan, Md Mehedi, and Khan A. Wahid. "Low-cost lifting architecture and lossless implementation of Daubechies-8 wavelets." IEEE Transactions on Circuits and Systems I: Regular Papers 65.8 (2018): 2515-2523.
3. Basiri, M. Mohamed Asan. "Efficient VLSI architectures of lifting based 3D discrete wavelet transform." IET Computers & Digital Techniques 14.6 (2020): 247-255.
4. Lone, Rafi, and Najeeb-ud-Din Hakim. "Multiplier-less architecture for 4-tap Daubechies wavelet filters using algebraic integers." International Conference on Intelligent Computing and Smart Communication 2019. Springer, Singapore, 2020.
5. Singh, Gyanendra, *et al*. "Novel Architecture for Lifting Discrete Wavelet Packet Transform With Arbitrary Tree Structure." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 29.7 (2021): 1490-1494.
6. M. T. Khan and R. A. Shaik, "Optimal Complexity Architectures for Pipelined Distributed Arithmetic-Based LMS Adaptive Filter," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 66, no. 2, pp. 630-642, Feb. 2019, doi: 10.1109/TCSI.2018.2867291.
7. M. T. Khan, J. Kumar, S. R. Ahamed and J. Faridi, "Partial-LUT Designs for Low-Complexity Realization of DA-Based BLMS Adaptive Filter," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no. 4, pp. 1188-1192, April 2021, doi: 10.1109/TCSII.2020.3035693.
8. Di Meo, Gennaro, *et al*. "A Novel Low-Power High-Precision Implementation for Sign–Magnitude DLMS Adaptive Filters." Electronics 11.7 (2022): 1007.
9. Mula, Subrahmanyam, Vinay ChakravarthiGogineni, and Anindya Sundar Dhar. "Algorithm and VLSI architecture design of proportionate-type LMS adaptive filters for sparse system identification." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 26.9 (2018): 1750-1762.
10. Narendran, S., and B. T. Geetha. "Performance Analysis of Parallel FIR Digital Filter Based on Even Symmetric Fast FIR Algorithm using Different Adders." 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE, 2021.
11. Betzel, Filipe, *et al*. "Approximate communication: Techniques for reducing communication bottlenecks in large-scale parallel systems." ACM Computing Surveys (CSUR) 51.1 (2018): 1-32.
12. Mahmud, Mohammad Sultan, *et al*. "A survey of data partitioning and sampling methods to support big data analysis." Big Data Mining and Analytics 3.2 (2020): 85-101.
13. Wei, Fengguo, Sankardas Roy, and XinmingOu. "Amandroid: A precise and general inter-component data flow analysis framework for security vetting of android apps." ACM Transactions on Privacy and Security (TOPS) 21.3 (2018): 1-32.
14. Stanley-Marbell, Phillip, *et al*. "Exploiting errors for efficiency: A survey from circuits to applications." ACM Computing Surveys (CSUR) 53.3 (2020): 1-39.
15. M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Computation offloading and content caching in wireless blockchain networks with mobile edge computing," IEEE Transactions on Vehicular Technology, vol. 67, no. 11, pp. 11008–11021, 2018.
16. Zhou, Yilian, *et al*. "MCAF: Developing an Annotation-Based Offloading Framework for Mobile Cloud Computing." Scientific Programming 2020 (2020).

17. Ghosh, Avrajit, Arnab Raha, and Amitava Mukherjee. "Energy-efficient IoT-health monitoring system using approximate computing." Internet of Things 9 (2020): 100166.
18. Liu, Weiqiang, Fabrizio Lombardi, and Michael Shulte. "A retrospective and prospective view of approximate computing [point of view." Proceedings of the IEEE 108.3 (2020): 394-399.
19. Pashaeifar, Masoud, *et al*. "Approximate reverse carry propagate adder for energy-efficient DSP applications." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 26.11 (2018): 2530-2541.
20. Marchisio *et al*., "Deep Learning for Edge Computing: Current Trends, Cross-Layer Optimizations, and Open Research Challenges," 2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2019, pp. 553-559, doi: 10.1109/ISVLSI.2019.00105.

Table 1: Entire Proposed Adder Operating Table.

Inputs			Sum Output	Comments for sum	Temporary outcome	Carry out Outcome	Comments for Carry out
C_{in}	A	B	S		XNOR(A,B)	C_{out}	
0	0	0	0	Eq.(2) is Indicated by blue colour	1	0	Eq.(5) is Indicated by REDcolour
0	0	1	1		0	0	
0	1	0	1		0	0	
0	1	1	0		1	1	
1	0	0	1	Eq.(3) is indicated by green colour	1	0	Eq.(6) is indicated by violet colour
1	0	1	0		0	1	
1	1	0	0		0	1	
1	1	1	1		1	1	

Table 2. Comparison Table

METHOD	Transistor count	Average power	Max power	Delay
EFA [9]	16	15.2uw	1.11mw	20.05ns
16T-FA [17]	15	13.8uw	1.17mw	60.16ns
12T -FA [16]	12	5.477uw	0.963mw	20.21ns
11T-FA [13]	11	4.90uw	0.510mw	40.28ns
proposed MLFA	10	2.90uw	0.421mw	12.34ns

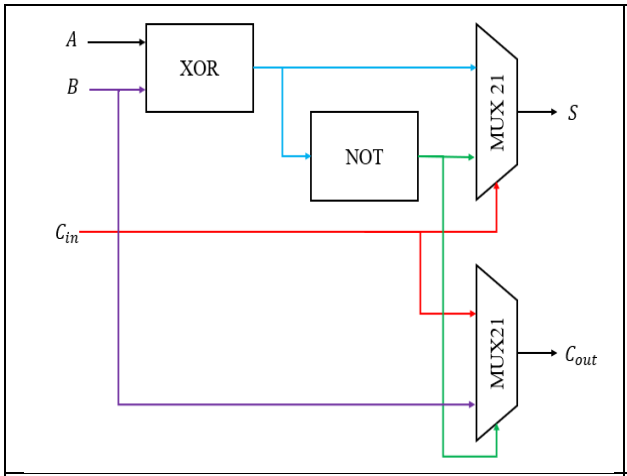

Figure 1: Proposed MLFA block diagram.

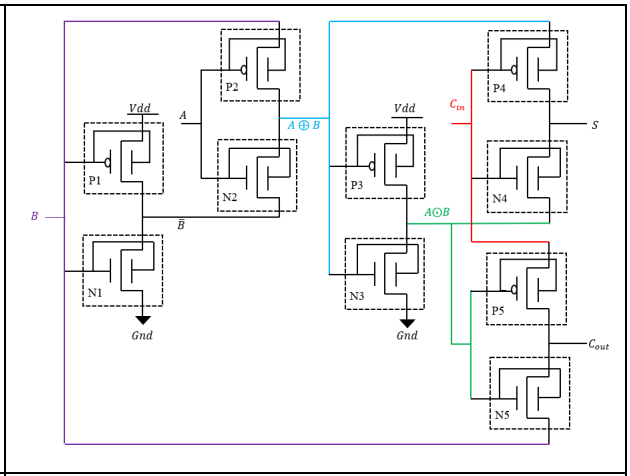

Figure 2: FinFET simulation of the suggested MLFA.

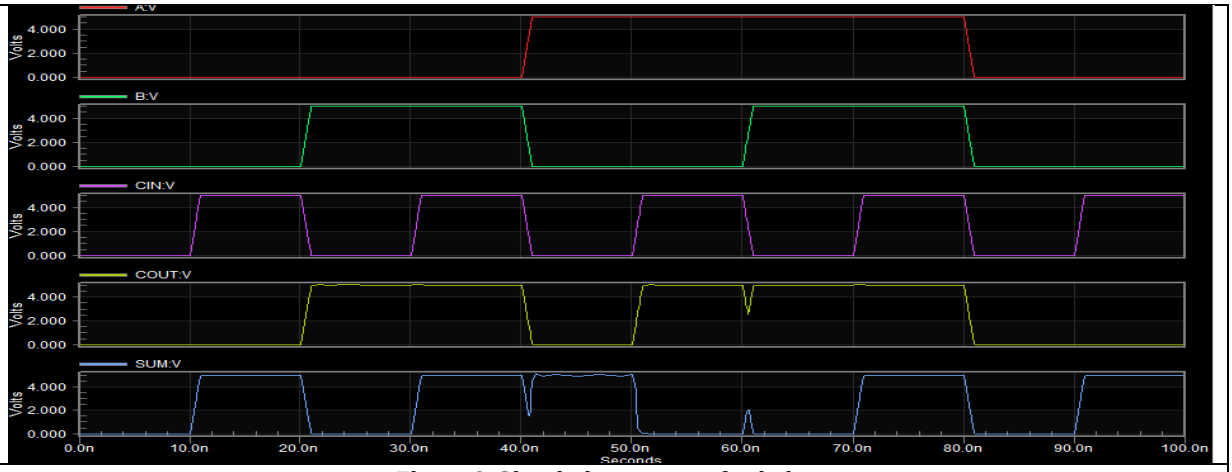

Figure 3. Simulation output of existing

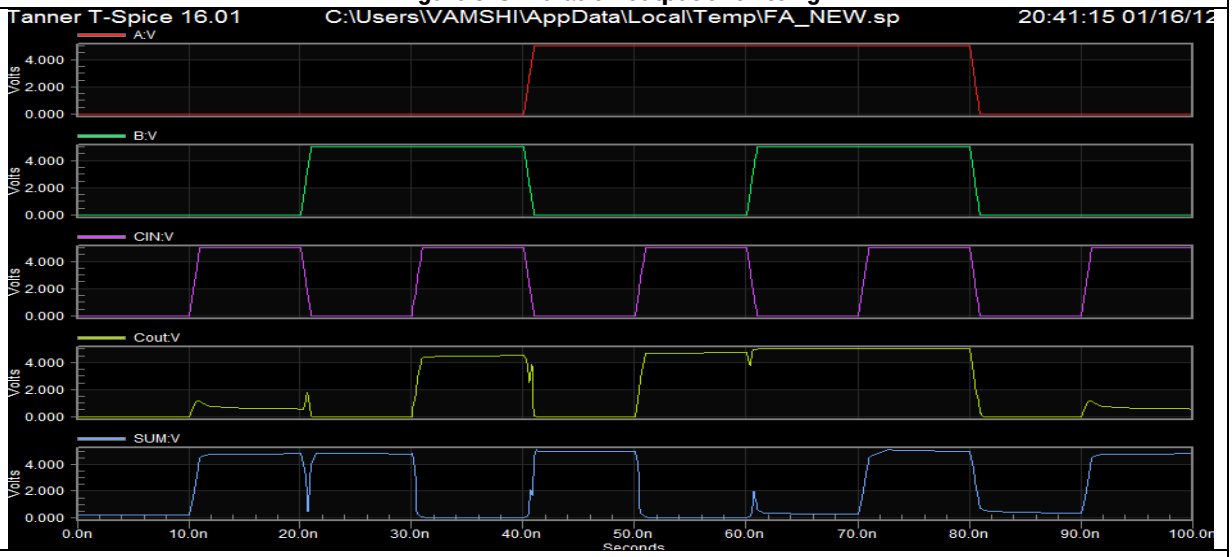

Figure 4. Simulation output of proposed MLFA