Storing and retaining divider using BDD-based adder/subtractor circuit

C.Senthilpari[1], Vishnupriya[1], S.Deivasigamani[2], Rosalind[3] G.Narmadha[4]
1 Faculty of Engineering, Multimedia University, 63100 Cyberjaya
2. Faculty of Engineering, Technology & Build Environment, UCSI University, Kuala Lumpur, Malaysia.
3. School of Mathematics Actuarial and computer sciences, Herriot-Watt University, Putrajaya, Malaysia.
4. Department of Electrical and Electronics Engineering, Sethu Institute of Technology, Virrudhunagar, India
Corresponding author:c.senthilpari@mmu.edu.my

Abstract: The BDD-based circuits are tree-structured and equally share the current/power in the cell, which gives reduced power dissipation reduced and increased speed. The proposed adder/subtractor circuits are designed and verified in this article using a Decision Diagram, which is implemented into a Retaining Array Divider (RAD) and Non- retaining Array Divider (NRAD) for 5G applications. The circuits are simulated and layouts tested using the Mentor graphics tool. The layout vs circuit schematic has been performed for the proposed adder-based RAD and NRAD and evaluated for the parameters of Chip area, propagation delay, and power dissipation. The results obtained are compared with the results of existing works by different. The proposed adder/subtractor circuits were designed using Silterra 0.13µm. The subtractor circuit is compared with the existing author circuit, which gives more than 95% improvement in Power dissipation and 17.39% improvement in propagation delay. Our proposed subtractor circuit has been designed with an inverter model, which occupies more area. The adder/subtractor circuits are further implemented in the retaining and non-retaining array divider circuits, giving better power dissipation with 36.02% than A,Arya et al. DAXD 99.79% and 99.74% than A.Arya et al. ADIV and ADIV6 divider model circuit. The propagation delay and area are improved by more than 80% in terms of delay and more than 14% in terms of area than the recent report designs.

Keywords: Binary Decision Diagram (BDD), Retaining Array Divider (RAD), Non-retaining Array Divider (NRAD), Power dissipation.

1. INTRODUCTION

Authentic two-way communication, wherein one party transmits information and the other party receives it, is the norm in the actual world. However, in communication engineering, the actuality of binary signals, binary messages, information, etc., is not readily apparent, and the intangibles must be fashioned through models. Since the dawn of the information age, the Shannon theory has been used widely. Claude Shannon describes the sender, the medium of transmission, and the receiver as basic components of a human communication system [1]. His breakthrough came when he realized telephone-switching circuits and Boolean algebra had a similar foundation. A mathematical performance of the Boolean function could be written as a sum of two subfunctions. This strategy is sometimes referred to as the "Shannon expansion." To manipulate and notate Boolean functions, many notations and approaches have been developed [2]. An acyclic graph is a type of graph in which each endpoint represents a Boolean function, and the nodes that make up the graph are either 0 or 1. The Arithmetic Logic Unit and the Floating-Point Unit (FPU) are essential microprocessor components. The above-mentioned specialized circuits perform arithmetic operations like adding, subtracting, multiplying, dividing, and calculating parity. Standard adder circuit equations [3] construct the full Adder's Sum and Carry circuits. A logic (1) connection between the source inputs guarantees a constant "ON" state. The circuit operates by the well-known carry equation [4]. A full adder has 3 binary inputs, such as A, B, and Cin, and 2 binary outputs, called a one-bit full adder (Sum, Cout). Creating and verifying a digital system can be described as a Boolean sequence. A well-performing Boolean function is a data structure that uses time and space effectively during execution. Binary decision diagrams, often known as Boolean function graphs, depict a Boolean function [2].

It is common knowledge that equivalence checking [5] ensures correct implementation. Verifying a microprocessor's arithmetic circuits has always been an important step. Symbolic simulations evolve conventional digital computational tools for simulation and verification [6]. Numerous companies provided equivalence-checking technologies for use in the design verification process.

Lee was the first to introduce binary decision diagrams as a data format for Boolean functions. Binary decision diagrams were also known as binary tree diagrams [7] and Akers [8] in 1959. Akers' suggestion to use a decision diagram to represent logical

operations is supported. These data structures were not commonly used for symbolic Boolean manipulation until Bryant [9] created a set of algorithms to work on them. The data structure known as a binary decision diagram represents various Boolean operations. The diagram is a directed acyclic network where the vertices represent the two possible outcomes of a binary decision diagram, denoted by the labels 0 and 1. Both regular and compact versions of ordered binary decision diagrams exist.

One can use ordered binary decision diagrams (OBDDs) to represent and manipulate Boolean functions across a finite domain with relative ease and generality. This is made possible by the diagrams' tree-like structure. Each input variable appears once along each path in an OBDD, making it a binary decision diagram. Each terminal node in the rooted finite directed acyclic graph representing f takes on the value 0 or 1 depending on the values of all the arguments along the corresponding path to the root, and each non-terminal node is labelled with one of the argument variables and has two successors that represent the choice between setting this variable to 0 or 1. Although the choice of variable ordering is completely arbitrary, it must be kept consistent across all functions being processed in parallel, and this has significant implications for the size of an OBDD representation.

Reduced Ordered Binary Decision Diagrams (ROBDDs) are used in logic synthesis, verification, and VLSI-CAD as a memory-efficient data format for manipulating Boolean functions. Other classes of BDDs [10] have been proposed that can take advantage of both scenarios despite the trade-offs in complexity and memory requirements per node. G.F. arithmetic circuits can be verified and debugged using cutting-edge forward rewriting-based techniques. This technique permits automatic detection and correction in G.F. circuits [11], and it is fault-tolerant, avoiding the polynomial-size explosion that plagues other systems. Boolean functions described as OBDDs [12] are amenable to many transformations, and each function can be reduced to its form with little effort.

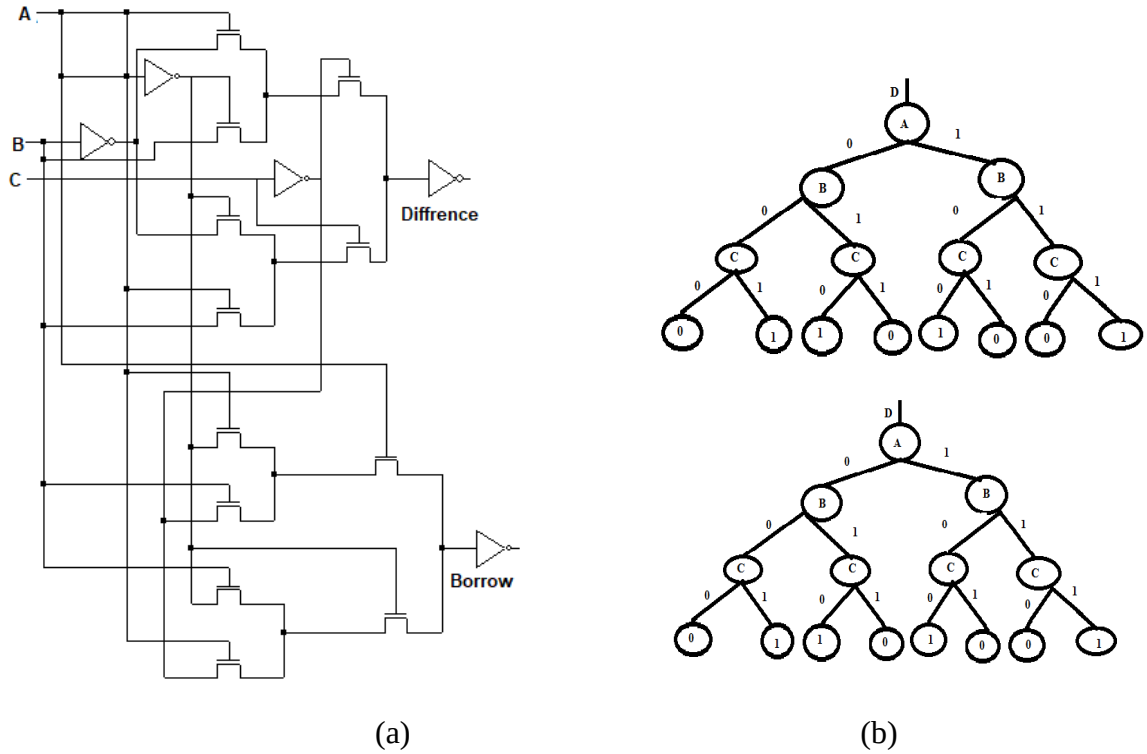
2. DESIGN OF RAD AND NRAD USING BDD

Multimedia 5G applications circuits include a retaining array divider and a non-retaining circuit. The retaining array division uses the radix-2 method, which needs to satisfy $2R_{j-1}-Y\geq 0$. The retaining circuit has been designed based on a cell structure containing a full adder and multiplexer circuit. The divider and divider inputs are fed into the corresponding inputs. In the same vein, the cell structure inspired the invention of the non-retaining array

divider circuit. The subtractor and multiplexer circuits are built within the non-retaining array divider.

2.1 FULL SUBTRACTOR

This subtractor circuit is designed using a CPL logic adder circuit in its design, shown in Figure 1 and is used to subtract three bits. It takes in A (the minuend), B (the subtrahend), and C (the subtrahend), and it gives out D (the difference) and B (the subtrahend) (borrow). According to the truth table of subtractors, the borrow logic '1' has to be considered for the design of the subtractor cell. The truth table's MSB number must only be initiated with the logic 1 values.



(a) (b)

Figure 1. (a) Circuit diagram for Full Subtractor (b) BDD diagram for subtractor circuit

Figure 2 illustrates all logic 1 values, and the CPL-based subtractor is designed in Figure 1. This subtractor circuit has been schematized employing CPL. It employs 2's complement method for the equation (A+(-B)). The subtractor design is based on the formula mentioned in equation (1). The CPL circuit has used many inverters difference so that swing restoration would be formed in the output terminal.

$$D_{iff} = A \oplus B \oplus C_{in} \ldots\ldots (1)$$

$$B = BC + \overline{A}\,(B + C)\text{---(2)}$$

The proposed circuit has been designed based on the BDD diagram shown in Figure 1 (b). This BDD diagram is associated with each node. The subtractor circuit has been designed based on 2's complement method of the adder cell, which is indicated in Figure 1 (b). The BDD-based subtractor circuit is the proposed subtractor circuit, designed based on the truth table of the subtractor. The BDD diagram changed as logic Boolean identities were simplified and schematized using the mentor graphics tool.

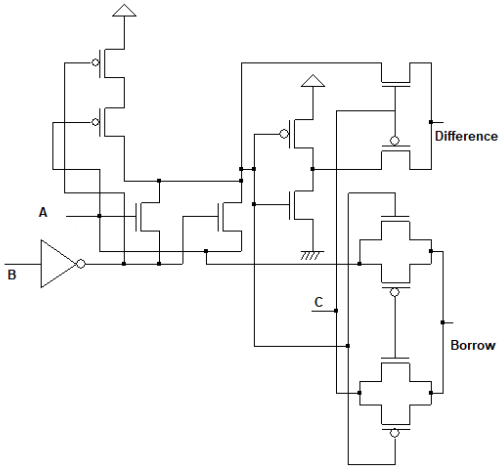
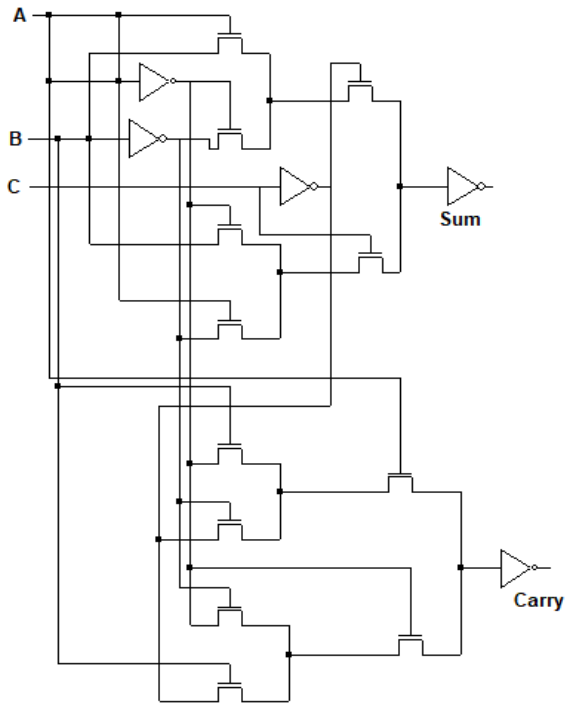


Figure 2. Proposed subtractor based on Binary decision diagram

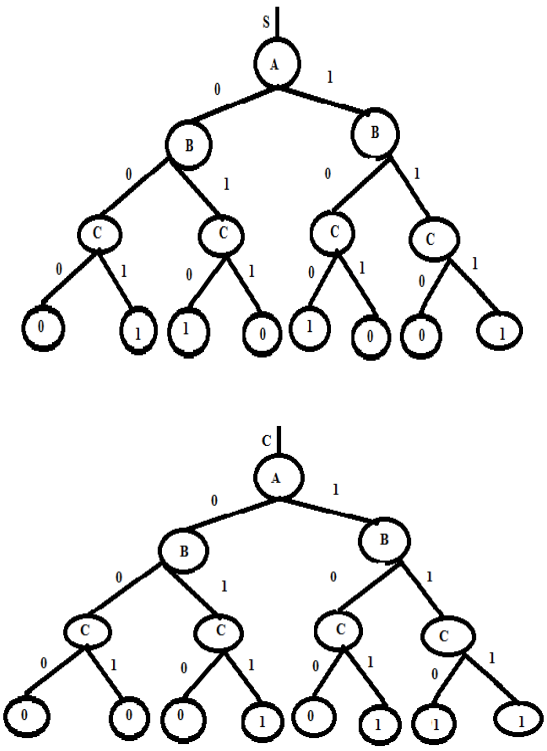
2.2 FULL ADDER USING BDD

The full-adder circuit for the CPL was built using equations 3 and 4, which used Two EXOR gates. Equation (3) shows how we came up with the multiplexing control input (MCIT) method. The modified circuit is shown in Figure 3(a), which reduced 2 transistors by the existing method. The BDD diagram for the full adder circuit is shown in Figure 3 (b). Figure 2 shows how the carry output from the above circuit is sent to the full Adder. The full Adder uses this carry output as its carry input. The full Adder's carry output is hooked up to the carry input of the next full Adder in the chain.



(a)

Figure 3 (a) Full Adder Circuit Diagram



(b)

Figure 3 (b). Binary decision diagram for Full Adder

The Boolean expression for Full Adder:

$$sum = A \oplus B \oplus C \quad (3)$$

$$Cout = C(A \oplus B) + A \cdot B \quad (4)$$

2.3 BDD-BASED PROPOSED ADDER

Figure 4 depicts the proposed adder circuit, created using a balanced layout to reduce the MOS transistors used in the original CPL adder layout (Figure 3). Further, NRAD and RAD circuit cells are built using the proposed full adder and subtractor circuits.

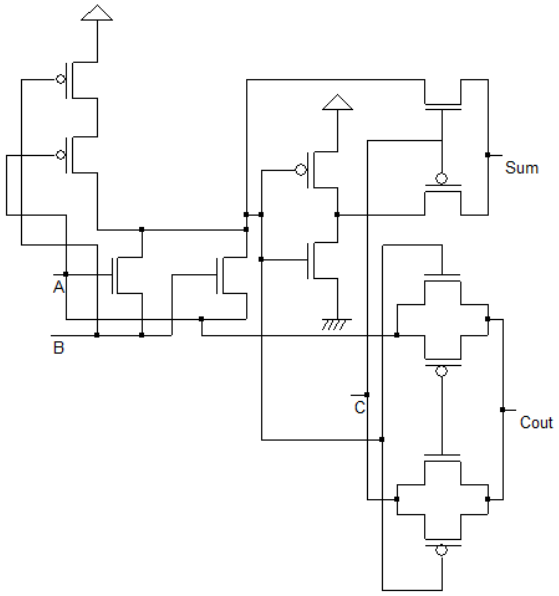


Figure 4. Proposed Adder

2.4 RETAINING ARRAY DIVIDER (RAD) ARCHITECTURE

The j^{th} bit of the quotient of the binary representation of the number Z (zj) is equal to qj when using the radix technique. When Rj′ is less than zero, the retention technique (Y) is implemented. To increase efficiency, we can use 2Rj - 1 as Rj by maintaining Rj - 1 instead of adding Y to Rj′ when Rj′ is less than 0. CPL is utilized during the manufacturing process of both the non-retaining and the retaining array divider circuits. The CPL design for the Retaining array divider is built on top of subtractor cells as its foundation. A full subtractor based on CPL is included in the RAD cell, and a multiplier with 2 inputs.

2.4.1 NRAD

A remainder correction circuit is required to get the right remainder while using a non-retaining division. If the final piece of the quotient is 1, the remainder is correct; otherwise, we need to add the divisor back to the partial remainder to get the right answer. Besides its application as an array divider, this circuit is a controlled adder. First, the registers must be shifted, then the Adder's signals must be allowed to travel, the next quotient digit must be calculated and stored, and finally, the trial difference, if necessary, must be recorded. The lengthening of the clock cycle is since later events in the same cycle are dependent on earlier ones.

2.4. 2 7X4 BIT NON-RETAINING ARRAY DIVIDER

In binary non-retaining division, the dividend and the ensuing partial product are calculated by subtracting or adding right-shifted iterations of the divisor. The carry-out from the partial remainder decides the next iteration's quotient and whether the shifted divisor is added or subtracted. A RAD is often implemented as a two-dimensional iterative array within a cell to shorten the number of division cycles. The array is built from controlled Adders, a type of basic cell.

For a complete, 7x 4 bit Non-retaining Array Divider shown in Figure 5, it has 13 inputs and 8 outputs, namely, $d_0,d_1,d_2,d_3,d_4,z_0,z_1,z_2,z_3,z_4,z_5,z_6,z_7$,cin for NRAD inputs while for outputs $q_0,q_1,q_2,q_3,r_0,r_1,r_2,r_3$. There are 4 rows consisting of 4 NRAD symbols circuits connected. Each cell of NRAD has a full adder connected to 2 input XOR circuits.

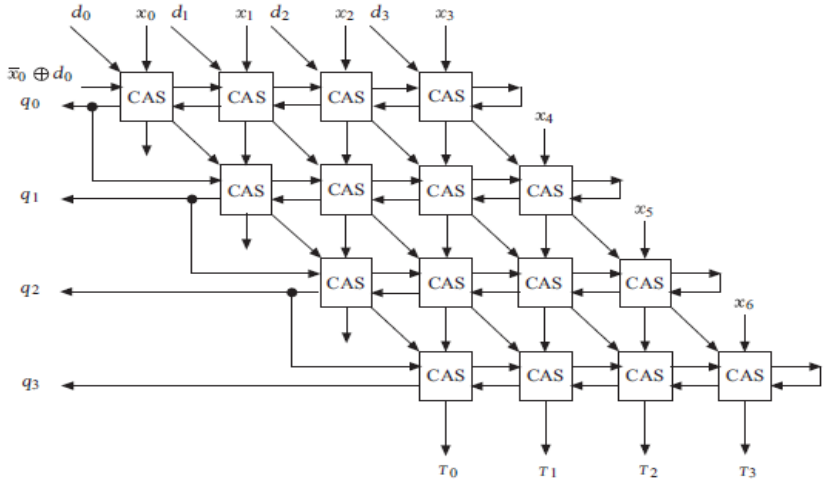


Figure 5. Non-retaining Divider architecture.

When D is a divisor, and A is the dividend, the quotient Q can be calculated using the binary division method [13] [14]. At each level of the procedure, D is either used to partition A into groups of bits or not. Bits are divided by the divisor if and only if the divisor's value must be less than input bit values. Therefore, the divider quotient can only be either logical 0 or logical 1. A full adder and an XOR gate with 2 inputs comprise a non-retaining divider cell.

The upper half of Figure 6 comprises four stages of cascaded full adders. The stages of dividend bits dictate the placement of the number of levels, while the stages are connected by full Adder seriously, which determines stages of divisor bits. Using four serially connected full adders, the seven dividend bits are divided into four stages, and the four

divisor bits are divided into four stages. The first stage of the cascaded full adders uses the four bits of the dividend considered the most significant. Subsequent stages of the cascaded full adders use one more bit of the dividend, and the three bits are carried over from the stage before it (thus the number four).

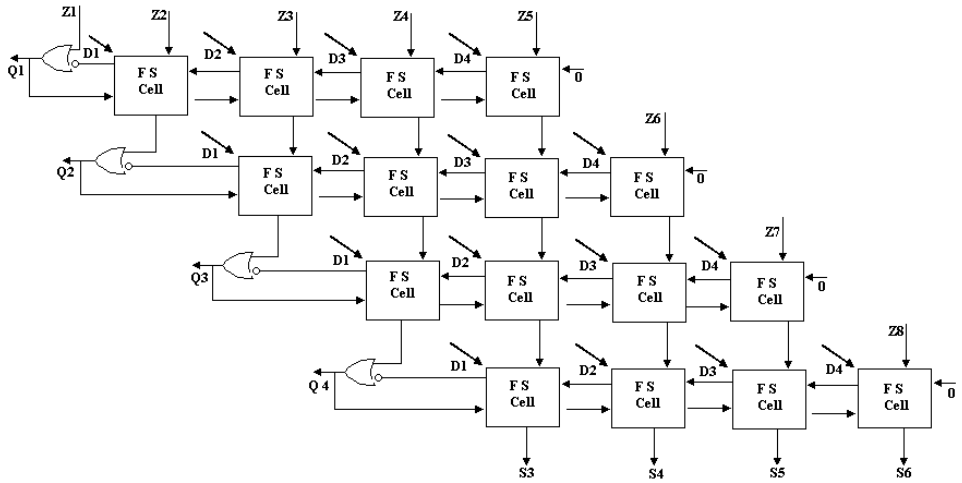


Figure 6: Non-retaining Array Divider

3. RESULTS AND DISCUSSION

The Mentor Graphics design tool simulates the proposed subtractor and adder circuits [15]. The structure of the suggested subtractor is that of a balanced tree. As a result, the inputs are in equilibrium, and the timing diagram (Figure 7) shows no losses in delivering the outputs. As a result of the well-designed -tree structure, the output of the subtractor reveals that both the difference and the borrow produce flawless outcomes. The logic function is moved up to the transistor level, and the logic 1 input is distributed evenly between the electron and themselves. The truth tables, therefore, provide the outputs.

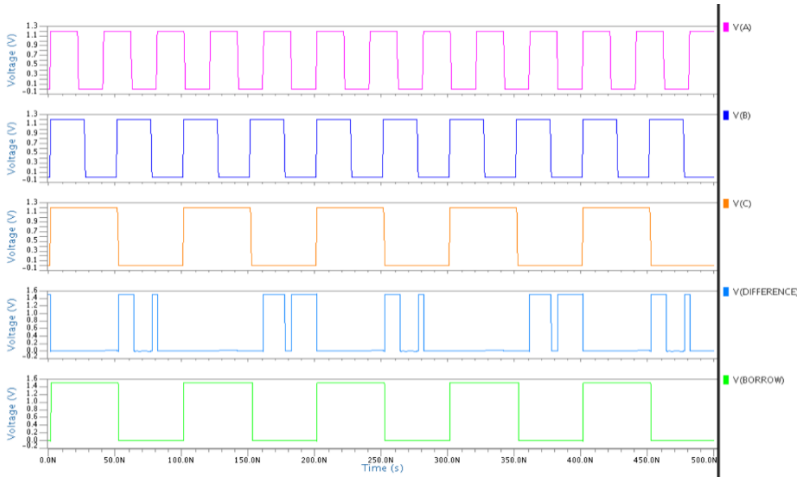


Figure 7. Full subtractor timing diagram

The full adder circuit is designed using equations (2) and (3). The full adder logic transition electron charges sharing with the transistor is good. When A=1, B=0 and Cin =0, The logic transition has a skew problem, which can be corrected by adding the capacitor in the output node, indicated in the timing diagram. (Figure 8).

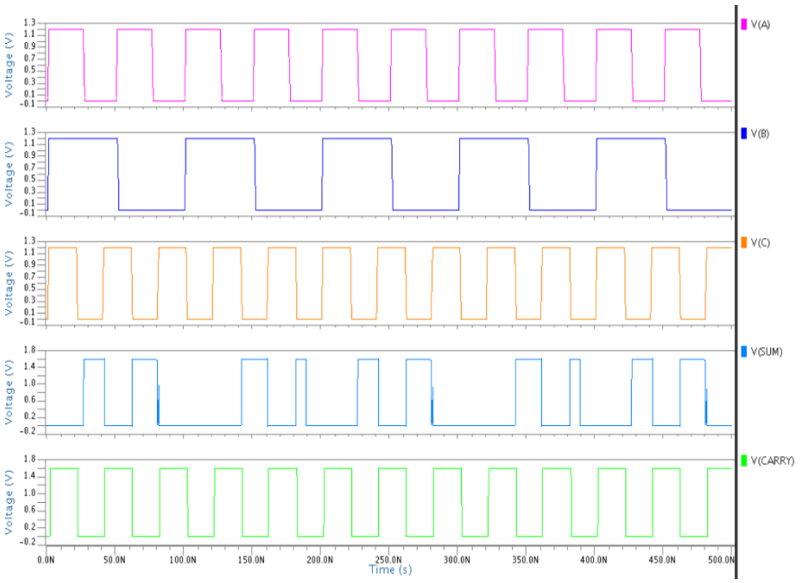


Figure 8. Full Adder – Timing diagram

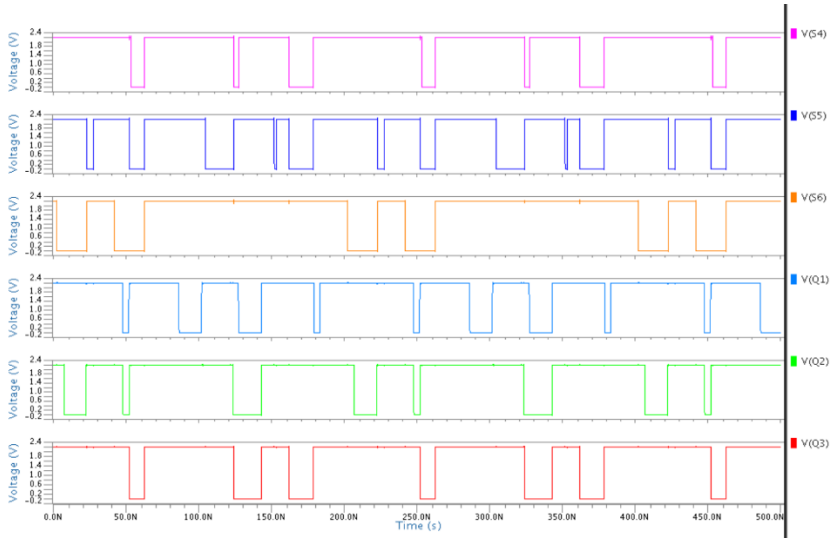


Figure 9. RAD – Timing diagram

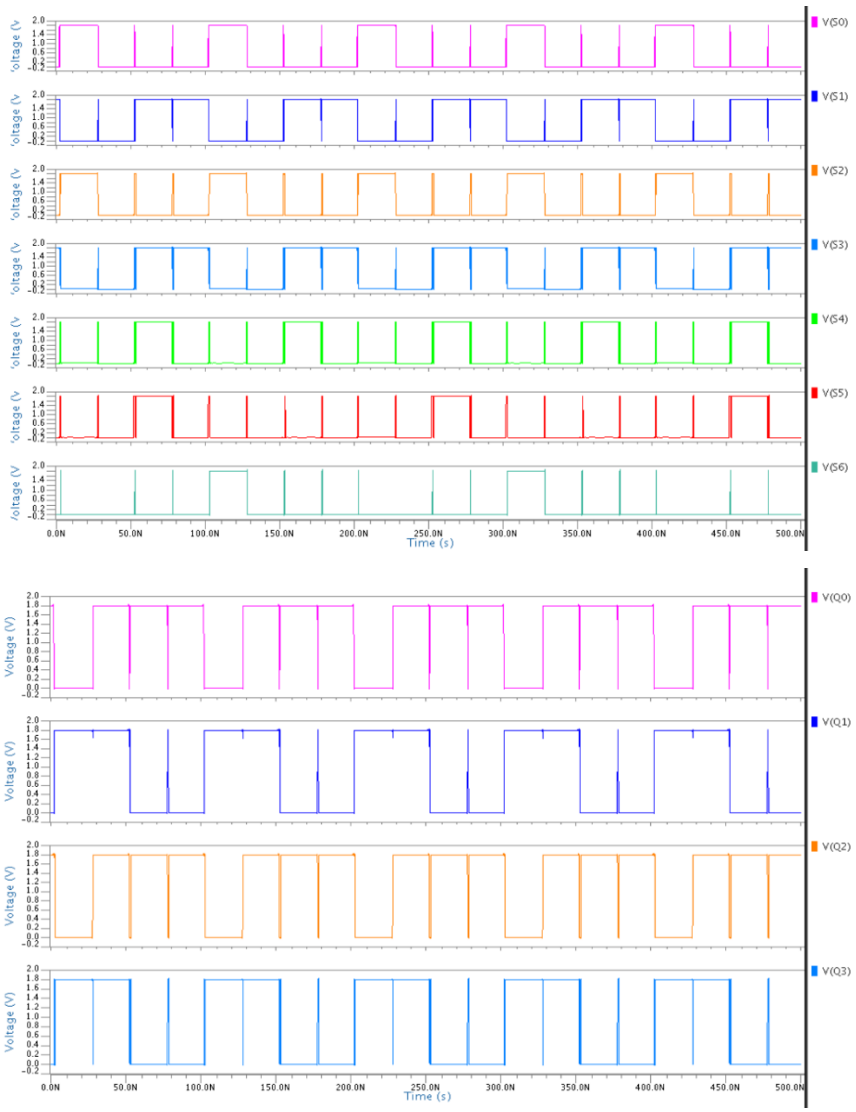


Figure 10. NRAD – Timing diagram

3.1 PERFORMANCE ANALYSIS

Mentor Graphics is used to simulate the suggested design, and the results are evaluated regarding power dissipation, area, and propagation delay[15]. Table 1 contains the Area and Power Dissipation (ADP) product and Power and Delay (PDP) production values for retaining and non-retaining array dividers with various feature sizes. There 12 transistors were utilized in the design of the full Adder, and full subtractor circuits were proposed. The RAD circuit with an adder requires 312 transistors, whereas the NRAD circuit implemented with a subtractor requires 512 transistors. The adder and subtractor cells utilized in the design each have a voltage of 1.5V, while the RAD and NRAD cells each have a voltage of 2V. The cells that have been suggested and designed will operate at a frequency of 2.3 GHz.

Table 1: Simulation results of the adder, subtractor, RAD and NRAD circuits

Circuit	Power consumption (nW)	Delay (ns)	ADP (μm^2-ns)	PDP (fJ)	Area (μm^2)	Latency ns	Through put (Mbps)	EPI ($x10^{-15}$)
Full adder	32.11	0.14	9.45	4.49	67.5	2.65	377.35	18.629
subtractor	29.1	0.19	13.25	5.52	69.75	2.69	371.74	20.260
RAD	57.8	0.42	528.7	24.27	1258.9	2.92	342.46	46.004
NRAD	96.6	0.48	967.1	46.36	2014.8	2.98	333.57	30.115

The BDD-based Adder subtractor circuits are compared with N.Arya ESC and CFSC subtractor circuits [16] [24]. The proposed subtractor circuit performs better than the N.Arya ESC and CFSC subtractor circuit due to the reduced design tree and perfect balance tree design, reducing the critical path and equal power distribution. The performance of the suggested subtractor circuit is superior to that of the ESC and SFSC subtractor circuits by more than 90 percentage points in terms of power dissipation and by 33.33 percentage points in terms of propagation latency. However, the available space has had to be reduced to accommodate the large number of transistors required for the 1-bit subtractor cell. Our proposed divider circuit's performance is superior compared to the divider circuit designed by Nagaswarareddy et al.[17] [18]. The proposed divider cell uses an adder and a subtractor to divide. The Nageswarareddy et al. divider circuit compared with our proposed model in terms of power dissipation and propagation delay, which our proposed circuit gives

83.69% and 99.04% improvement in terms of power dissipation and propagation delay Nagaswarareddy divider circuit. A comparison is made between the proposed adder and subtractor circuit and the N.Arya DAXD, ADIV, and ADIV6 divider circuits. The adder and subtractor circuit we described performs significantly better than the one developed by N.Arya and colleagues. BDD has been used as the foundation for the design of the suggested Adder and subtractor circuit. Therefore, the circuit tree structure is balanced and normally distributes electrons. Therefore, there will be no dissipation during charge sharing, and the RAD and NRAD circuits will have a reduced critical path.

Table 2: Comparison with existing authors

reference	**Power nW**	**% of Improvement**	**Delay**	**% of Improvement**	**Area**	**% of Improvement**
Proposed Adder	32.11	----	0.14ns	----	67.5 μm^2	----
Proposed subtractor	29.1	----	0.19ns	----	69.75 μm^2	----
Proposed NRAD	96.6	----	0.48ns	----	2014.8 μm^2	----
Proposed RAD	57.8	----	0.42ns	----	1258.9 μm^2	----
Nageswarareddy	592.3	83.69	50.14 ns	99.04		
N. Arya et al. ESC Subtractor	1.5µW	97.85	0.21	33.33	30 μm^2	-56.98
N. Arya et al. CFSC Subtractor	1.84 µW	98.41	0.23	17.39	34 μm^2	-51.25
N. Arya et al. DAXD divider 14 bit	90.35	36.02	3.77	88.85	2354	14.40
N. Arya et al. ADIV divider 6	46.06	99.79	5.92	91.89	2446	17.62
ADIV-6 Neelam Arya(et.al)	37.52	99.74	5.92	91.89	1,262,817	99.99

3.3 MONTE -CARLO METHOD ANALYSIS FOR NRAD/RAD

A series of numbers can be broken down into component frequencies using the digital Fourier Transform (DFT) [18]. The sequence of infinite instructions for the probability distribution format can be roughly categorized as either time decimation or frequency decimation. The Monto Carlo method of the Mentor graphics software provided the FFT, which will provide an analogue simulation of the RAD and NRAD circuit's inputs and outputs. The Monte Carlo for RAD and NRAD is shown in Figures 14 and 15, respectively. Figure 14 shows the relationship between frequency (GHz) and Voltage (mV) for node 0. The highest value of frequency is 39 MHZ at 188.25 mV. A Binary Decision Diagram can

also be implemented in approximate arithmetic, which can be used for image processing applications [19].

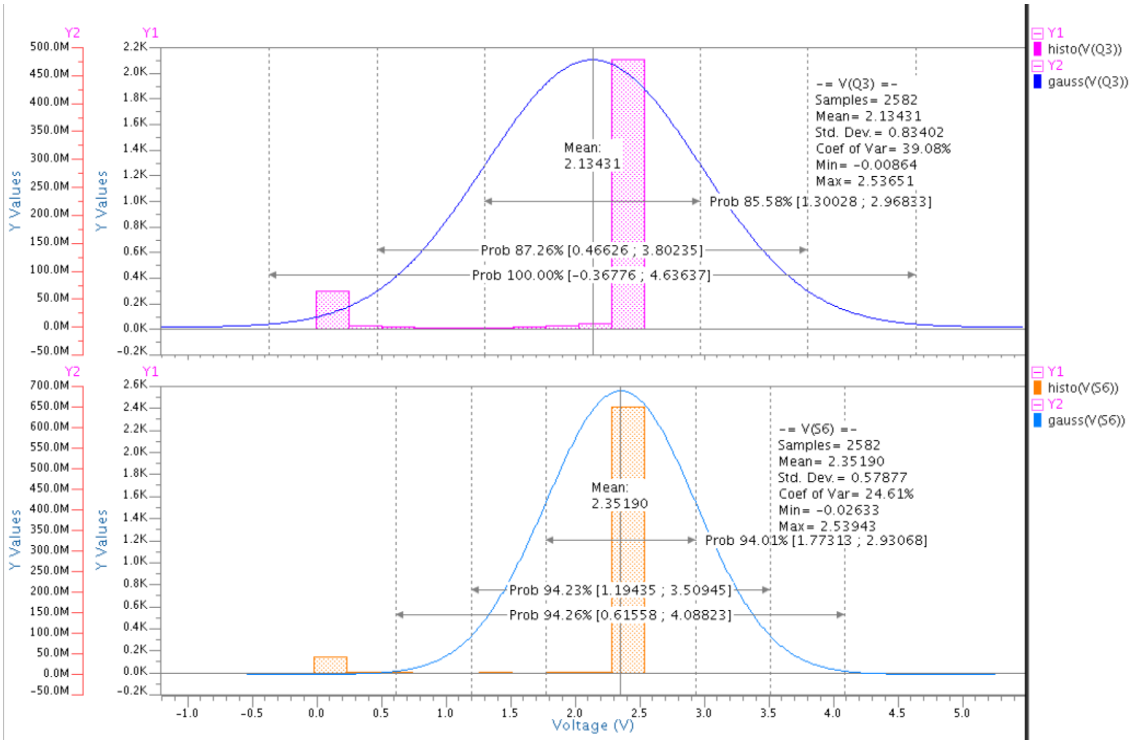


Figure 14. Monte Carlo simulation for RAD

A binary decision diagram adder-based RAD circuit has analyzed the Monte Carlo simulation. The monte carlo simulation is based on histogram and Gaussian methods[20]. The histogram method analyses a rough sense of density, which is the underlying distribution of data. The value of the distribution measures the probability of density. The means V(Q3) sample quality is 2582, and the standard deviation is 0.83402V. The pdf coefficient of variation is 39.08%.

The below-mentioned formula has measured the probability distribution format

$$P_{r[a\leq x\leq b]} = \int_a^b f_X\,(x)dx$$

The F.X. is the cumulative distribution function of X.

The Gaussian distribution of the function has been formulated using this formula

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{1(x-\mu)^2}{2\ \ \sigma^2}\right)}$$

The Gaussian function was improved by composing the expositional function with a concave quadric function[21].

The pdf median is 87.25%, which shows the design is perfect and gives better results than other existing circuits. The same design was analyzed regarding V(S6) adder-based cells and used 2582 samples. The mean value of this design is 2.35190, and the probability of the design has improved by 94.01% Adder-based design. The variation voltages are changed between 1.5V to 3V, which gives better results.

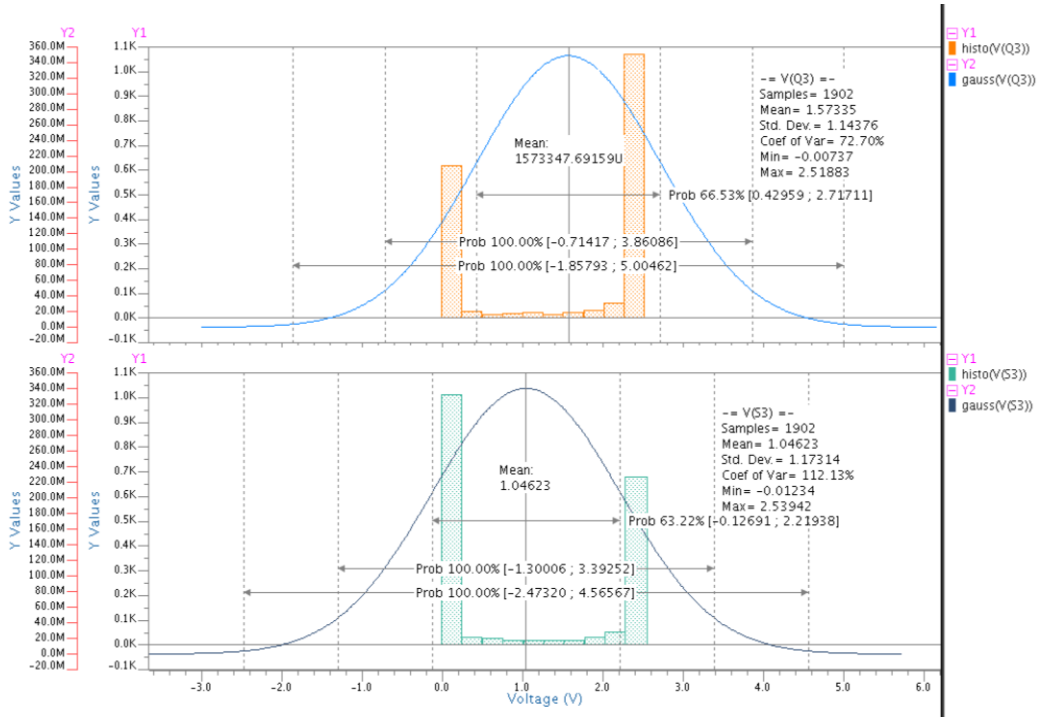


Figure 15. Monte Carlo simulation for NRAD

The layout of RAD and NRAD is shown in Figure 16. The layout has been drawn using mentor graphics using a schematic-driven layout (SDL). The layouts are analyzed using DRC, which verifies all the polygons and layers[22] [23]. The adder and subtractor circuits are implemented into RAD and NRAD cell structures.

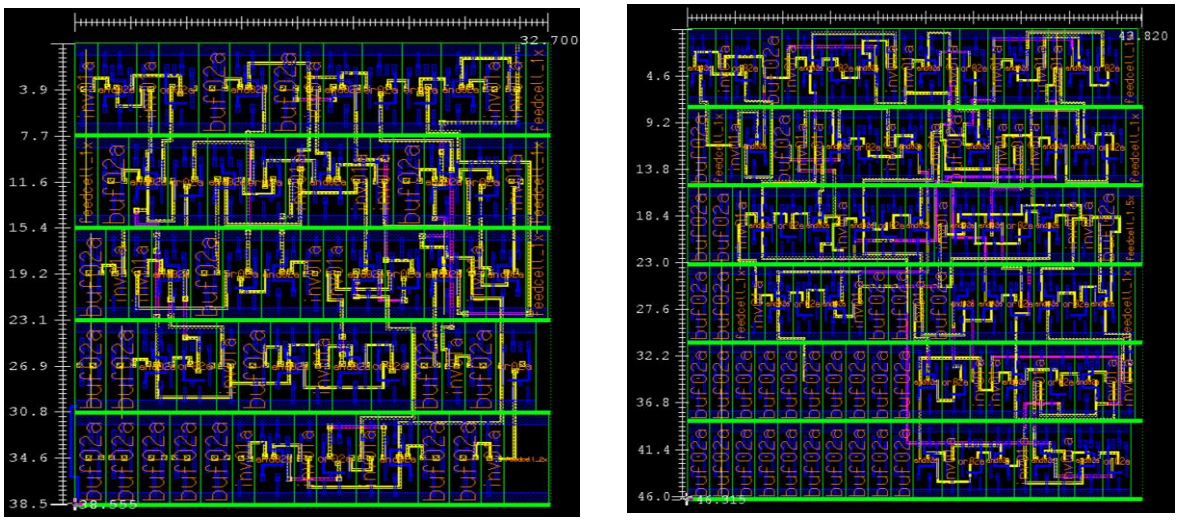


Figure 16. The layout of RAD and NRAD

CONCLUSION

The suggested Adder and subtractor circuit is used in the design of both the retaining and non-retaining array dividers that are being proposed. The suggested adder and subtractor circuit went through the process of being schematized and analyzed in the Mentor graphics tool. The performance of the proposed Adder and subtractor circuit, which is implemented in both the NRAD and RAD circuit, is superior to that of work by other authors' circuits that were published. By employing mentor visuals, the designers can observe the design trade-offs, such as power dissipation, chip size, and delay. LVS was used to determine the connection between various metrics such as capacitance, power dissipation, VDD, and IDD. Based on the performance analysis, it has been determined that better results can be obtained using alternative provided data. It was also seen that when there is an increase in temperature, the number of transistors and the VDD, the power dissipation, also increases.

Data availability statement:

This paper based on simulation results, there is no data available in this paper

-Conflict of interest:

No conflict of interest

References

[1] Shin-ichi Minato, "Binary Decision Diagram and Application for VLSI CAD", NTT LSI Laboratories Kanagawa Japan, Kluwer Academic Publisher,1996

[2] Markovi D, Nikoli B, Oklobdija VG. A general method in synthesis of pass-transistor circuits. Microelectr J 2000; 31: 991-998.

[3] C.Senthilpari, K.Diwakar, Deivasiganmani, Velrajkumar and Rajendren ayyavoo "Power Efficiency Top-Down ALU for Error Correction and Detection Circuit- IEEE conference, Proceedings of Third IEEE International Conference on Communication, Computing & Industry 4.0 (C2I4) – 2022 ISBN: CFP22Y51-DVD 978-1-6654-7496-2; EXPLORE: CFP22Y51-ART 978-1-6654-7497-9

[4] Reto Z, Wolfgang F. Low-power logic styles: CMOS versus pass transistor logic. IEEE J Solid-State Circuits 1997; (32): 71079-1090.

[5] Stoffel, D.; Kunz, W. "Verification of Integer Multipliers on the Arithmetic Bit Level" Computer-Aided Design, 2001. ICCAD. IEEE/ACM International Conference on 2001, pp. 183 - 189

[6] Ying-An Chen "Arithmetic Circuit Verification Based on Word Level Decision Diagrams",[Online] http://reports-archive.adm.cs.cmu.edu/anon/1998/CMU-CS-98-131.pdf [November 21 2011]

[7] C.Y. Lee, *Representation of Switching Circuits by Binary-Decision Programs*, Bell System Technical Journal, Vol. 38, July 1959, pp. 985-999

[8] S.B. Akers, *Binary Decision Diagrams*, IEEE Transactions on Computers, Vol. C-27, No. 6, June 1978, pp. 509-516.

[9] Randal E. Bryant *Graph-Based Algorithms for Boolean Function Manipulation*, IEEE Transactions on Computers, 1986 http://www.cs.cmu.edu/~bryant/pubdir/ieeetc86.pdf in 1986.

[10] Babar, Junaid & Jiang, Chuan & Ciardo, Gianfranco & Miner, Andrew. (2019). Binary Decision Diagrams with Edge-Specified Reductions. 10.1007/978-3-030-17465-1_17.

[11] S.U., TIANKAI, "Analysis And Verification of Arithmetic Circuits Using Computer Algebra Approach" (2020). *Doctoral Dissertations*. 1868. https://doi.org/10.7275/15875490

[12] Bryant, R.E. (2018). Binary Decision Diagrams. In: Clarke, E., Henzinger, T., Veith, H., Bloem, R. (eds) Handbook of Model Checking. Springer, Cham. https://doi.org/10.1007/978-3-319-10575-8_7

[13] G. Narmadha, S. Deivasigamani, K. Balasubadra, M. Selvaraj," A low power and high-speed approximate Adder for image processing applications". Journal of Engineering Research, 10(1A), pp.150-160, March 2022.

[14] Thangavel, Bhuvaneswari & Soong, Lim & Singh, Ajay Kumar & Prasad, V.C.. (2015). Design of binary decision diagram (BDD) optical adder. 2014 International Conference on Advances in Engineering and Technology Research, ICAETR 2014. 10.1109/ICAETR.2014.7012834.

[15] Bhagyalakshmi V. and D.B. Mantri, A BDD-based Design of An Area-Power Efficient Asynchronous Adder, International Journal of Electronics and Communication Engineering. ISSN 0974-2166 Volume 6, Number 1 (2013), pp. 93-103.

[16] Ismail Gassoumi, Lamjed Touil, Abdellatif Mtibaa, "An Efficient Design of QCA Full-Adder-Subtractor with Low Power Dissipation", *Journal of Electrical and Computer Engineering*, vol. 2021, Article ID 8856399, 9 pages, 2021. https://doi.org/10.1155/2021/8856399.

[17] J Nageswara Reddy a, Dr. Mukesh Tiwari b and Dr. Vundela Padmanabha Reddy, Power Efficient Two Transistor Exclusiveor Gate for Full Adder Usinggdi in 45NM , Turkish Journal of Computer and Mathematics Education Vol.12 No.2 (2021), 1342-1347.

[18] Vishnupriya Shivakumar, C. Senthilpari, Zubaida Yusoff, Test power and area optimized logic built-in self-test with higher fault coverage for automobile SoCs, Microelectronics Journal, Volume 124, 2022, 105430, ISSN 0026-2692, https://doi.org/10.1016/j.mejo.2022.105430.

[19] Senthilpari C, Deena R and Lini L. Low power, less occupying area, and improved speed of a 4-bit router/rerouter circuit for low-density parity-check (LDPC) decoders [version 2; peer review: 2 approved]. *F1000Research* 2022, **11**:7 (https://doi.org/10.12688/f1000research.73404.2)

[20] C Senthilpari (2011) A Low-power and High-performance Radix-4 Multiplier Design Using a Modified Pass-transistor Logic Technique, IETE Journal of Research, 57:2, 149-155, DOI: 10.4103/0377-2063.81744

[21] C Senthilpari, K. Diwakar and A. K. Singh, "High speed and high throughput 8×8 bit multiplier using a Shannon-based adder cell," *TENCON 2009 - 2009 IEEE Region 10 Conference*, Singapore, 2009, pp. 1-5, DOI: 10.1109/TENCON.2009.5396177.

[22] Shivakumar, V., C. Senthilpari, & Yusoff, Z. . (2021). Design a 1.9 GHz low-power LFSR circuit using the Reed-Solomon algorithm for Pseudo-Random Test Pattern Generation. *International Journal of Integrated Engineering*, *13*(6), 220–232. Retrieved from https://publisher.uthm.edu.my/ojs/index.php/ijie/article/view/6748.

[23] V. Mrazek, "Optimization of BDD-based Approximation Error Metrics Calculations," *2022 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, Nicosia, Cyprus, 2022, pp. 86-91, DOI: 10.1109/ISVLSI54635.2022.00028.

[24] Neelam Arya, Teena Soni,Manisha Pattanaik, G.K. Sharma, Bit significance based reconfigurable approximate restoring dividers andsquare rooters, Microelectronics Journal 104 (2020) 104861