Research Article

Implementation Of Power Efficient Accurate And Approximate Full Adders For Image Processing Applications

[1]Mr. K Madhava Rao, [2]S Shiva Prasad, [3]Velamati Avinash, [4]Y V Sudershan Reddy [5]B V Sumanth

[1]Assistant Professor, [2345]B.Tech Students
Department of ECE
B.V. Raju Institute of Technology, Vishnupur, Narsapur, Medak.

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 16 April 2021

Abstract: In Integrated circuits (IC)s system Computational performance is limited by its performance and since the execution time is dominated by the multiplication factor because of that high-speed adder is much more important in DSP systems. This paper presents an efficient full adder design in CMOS 25nm technology using footed quasi resistance-based gate diffusion input (FQR-GDI). This design uses a smaller number of transistors than the conventional CMOS based Adder designs. By employing the FQR-GDI technique there is an extensive decrease in power and delay of the circuit. Design also solves the threshold drops problem of Original GDI cell resulting in better output signals. The proposed methodology implemented in Tanner Tools using TSMC library.

1. INTRODUCTION

In present generation the growth of integrated circuit devices has increased a lot. The VLSI applications are given as digital signal processing and microprocessors. These applications are most widely used to perform arithmetic operations. Along with that multiplication, subtraction operations are also performed. By using some modules all these operations are performed on the 1-bit full adder circuit. Approximate adders are implemented in a way to balance the trade-off between accuracy Vs performance/power. The areas of image and video processing appear as good case studies for the use of approximate results. The circuit design is addressed at different levels. The design parameters of the circuit produce good trade off in terms of speed and area [1]. Multiplication has been given careful consideration by utilizing analysts, because of the reality expansion which is essentially bitwise activity among two subject components and the additional complex tasks, reversal and might be finished with a few multiplications. Montgomery's adder is classified into three types, they are bit-serial, bit-parallel and digit serial architectures. Bit-parallel shape is rapid; however, it's far steeply-priced in phrases of vicinity. Bit-serial structure is region efficient, but it's far too sluggish for plenty packages. The digit-serial structure is flexible which may change the space and velocity, consequently, it achieves a moderate pace, reasonable price of implementation and hence it is most appropriate for practical use. Montgomery presented a technique for figuring modular multiplication productively. He introduced to move the portrayal of numbers from the Zn to an alternate area called Montgomery Residual portrayal or Montgomery Domain [2-3].

Here for the purpose of security, the computers and communication system brought with a demand from private sector [4]. The Montgomery multiplication is the calculation that permits effectively for registering. The expense of the particular duplication is equivalent to three whole numbers which increases in addition to the expense of the change in the Montgomery area. Yet, in the event that the large-scale task is an exponentiation at point the change cost is insignificant contrasted with the quantity of augmentations executed in the Montgomery area. In the process of Montgomery multiplication, preprocessing unit and post-processing units are used [5]. The pre-processing unit produces N-Residue operands and in the same way post processing unit will eliminate the constant factor 2n. Hence to form N Residue operands in the system, modular exponentiation is used.

The adoption of lower technology nodes has put the VLSI industry in a position to deal with various challenges. When compared to planar CMOS devices the deployment of Footed quasi resistance (FQR) has increased the power and performance characteristics but by creating new technical and collaborative challenges such as device types, material physics, electro migration, lithography process, double patterning, etc. The gate of FQR device is wrapped around the drain-source which is a conducting channel. This yields significant improvement in performance by providing better electrical properties, low threshold voltages, increased electrostatic control, higher density and reduced dynamic and leakage power. FQR can easily be replaced by the bulk CMOS devices due to its benefits like high ON and low OFF currents, intra-die variability, low power consumption, high integration density and reduce in short channel effects. The VLSI design cycle primarily comprises of two design stages: frontend design and the backend design.

Rest of the paper organized as follows section 2 gives the detailed analysis of conventional methods and its problems. Section 3 gives the detailed analysis of proposed array adder with respect to the various block level operations using FQR full adder formulations. Section 4 gives the detailed analysis of Tanner EDA software simulation of the proposed method with area, power and delay analysis with simulation waveforms and comparative analysis with various literatures. Section 5 concludes the paper with possible future studies.

2. LITERATURE SURVEY:

The foremost step in physical design is the floorplan. Floorplan stage involves relative placement of the macros, sub-blocks, IPs, I/O pads, and pins. It also specifies the aspect ratio, die size, shape, and core utilization area of the chip. Soft, hard, or partial placement blockages are added to reduce the density in specified areas [1]. This stage includes creating prototype that uses multiple iterations of floorplan to be performed to obtain less congestion, wiring length and with the focus on rout ability. This stage is followed by power planning where the Power Grid (PG) network is built by adding power rings and stripes to allocate the power supply to each component of the chip [3-4]. The next stage is the placement which is a process of placing the standard cell instances on the die. The purpose of placement stage is to place the cells in design without overlap, minimize the area of layout, and reduce the timing of critical nets. The register grouping can be done after placement stage to reduce area and power [5]. The next step is to deliver the clock signals equally through Clock Tree Synthesis (CTS) to all sequential components present in the system. The aim of CTS is to control the skew, insertion delay and optimize clock and data path to achieve better Power, Performance and Area (PPA) [6]. The CTS is followed by Routing where the precise paths for each net are connected through metal layers. The routing is carried out by considering logical connectivity well-defined in the netlist without violating the rules of design [7]. The interconnection of instances i.e., routing takes place through two steps: first the global routing and later the detailed routing. In a design there should be n-well continuity, so Filler cells i.e., a type of special cell used in Physical Design (PD) are added to avoid the gaps [8-9]. Few other special cells such as Tie High and Tie Low cells have its usage in design to sustain the constant VDD and VSS respectively, tap cells are used to avoid latch up problem, End cap cells avoid cell damages at the end of the row, Decap cells act as charge reservoir to resolve dynamic IR drop, Spare Cells are used at Engineering Change Order (ECO) stage. The aim of post layout optimization/verification is to reduce delay, perform DRC, LVS checks and fix the violations, check for timing constraints, fix setup and hold violations, minimize the effects of IR drop and electromigration. To perform timing closure with no violation buffer sizing, VT swapping, increasing Drive strength, Insertion of buffers or repeaters and few other techniques are carried out. The design is ready for fabrication, after layout and verification. Traditionally the layout data to be fabricated is sent on a tape. Thus, this data release event is called as tape out. The VLSI chips manufacturing process consists of sequential steps such as lithography, etching, deposition, chemical mechanical polishing, oxidation, ion implantation and diffusion. The correctness and efficacy of the test is most necessary for quality goods. In a fabrication center, the fabrication of wafer takes place which is then transformed into individual chips. Every chip is afterwards packaged and tested to make sure it follows all the specific model requirements and works as intended. The semiconductor industry relies profoundly on EDA, and scripting forms an essential part of EDA. Perl is the most common approach for this job, but TCL is also widely used.

3. FQR BASED GDI

To design low power circuits are highly challenging and one does not know whether the timing constraints can be satisfied until the final routing has completed. In Deep Submicron (DSM) processes of FQR the most influential on consumption of total power is the leakage power. Depending on unique design and application, devices have different GDI goals and one of these parameters is always a tradeoff. From RTL to signoff stage multiple optimization techniques are performed - upsize, downsize, VT swap, and many more, and all these factors impacts or tweaks design PPA. In a VLSI IC, electrical energy during circuit operations is converted to heat energy and the rate at which this energy is removed from the source and converted into heat is known as power dissipation. To avoid the increase in chip temperature heat energy must be dissipated from the chip which otherwise can affect the circuit leading to its failure. The two major sources of power dissipation are the static power and dynamic power dissipation. The dynamic power dissipation is mainly due to switching activity of the capacitance and short circuit current while static power dissipation is because of subthreshold, gate and junction leakage currents.

FQR-GDI Technique was mainly focused on reduction of power consumption, area and complexity of digital combinational circuits. FQR-GDI uses a CMOS inverter circuit to implement various complex logic functions listed in Table 1 using only two transistors based on the inputs given to G, P and N nodes. The transistors PMOS and NMOS will share a single gate terminal input G, while N and P are inputs for NMOS and PMOS source terminals respectively as shown in Figure 1a.

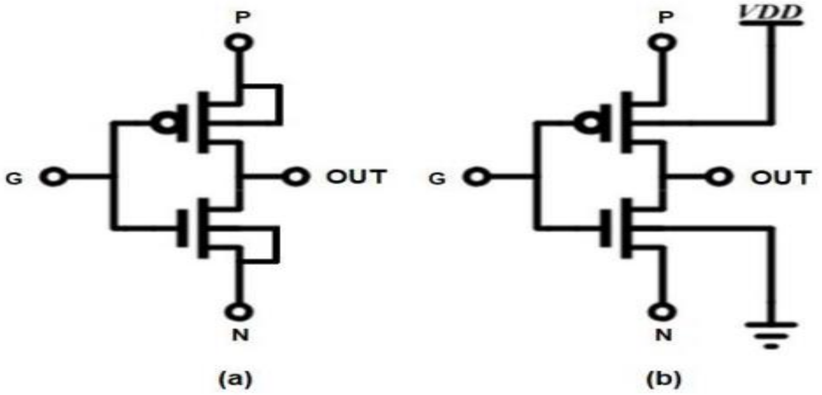


Figure 1. (a) GDI cell (b) FQR-GDI cell

However, the drawback of the original FQR-GDI cell was dropping of threshold which tends to reduction in performance and increases static power dissipation. Improvising this, FQR-GDI approach was proposed [2] by connecting the PMOS and NMOS transistors bulk nodes to supply voltage and low constant voltage respectively, shown in Figure 1b.

Table 1. Modified-GDI cell Logic Functions

(G)	(P)	(N)	Outputs
A	B	B’	A’B + AB’
A	0	B	AB
A	1	0	A’
A	B	C	A’B + AC

Using the Modified-GDI cell we can implement AND gate and XOR gates by giving input to respective nodes in M-GDI cell as per the Table I. Here, Figure 2a shows the AND gate and Figure 2b shows XOR gates designed using M-GDI.

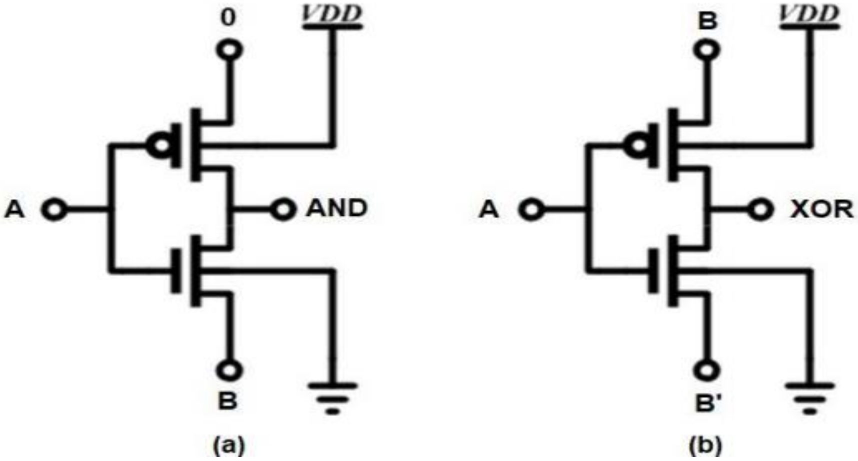
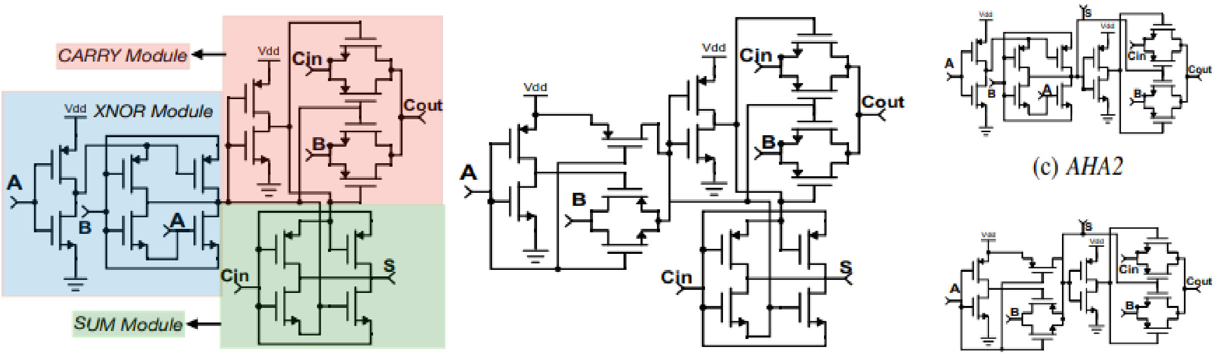


Figure 2. (a) AND gate using FQR-GDI (b) XOR gate using FQR-GDI

4. ADDERS USING FQR-GDI

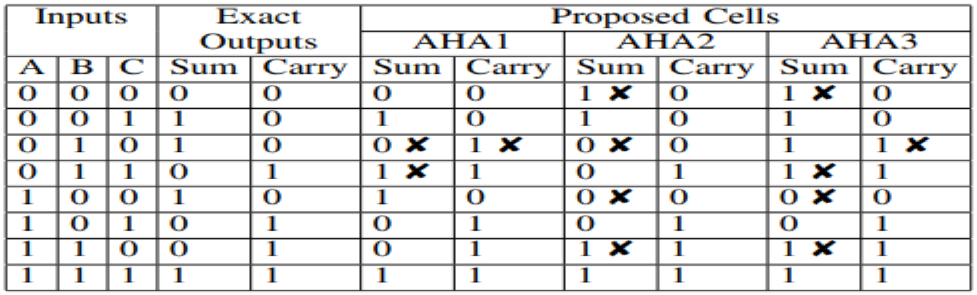


(a) *EHA [11]* (b) *AHA1* (c) *AHA2* (d) *AHA3*

Fig. 2: Adder Designs

In this section we discuss the design of the approximate hybrid CMOS based adders. The general approach, as seen from the prior works in the design of approximate adders is to first select an existing exact adder and then remove some transistors and/or replace them to reduce area and power. We have used a similar approach to design three different approximate adders with varying levels of inaccuracies. The proposed adders are named approximate hybrid adders (AHA1, AHA2, AHA3), with a numeral at the end for the different designs. Our approximate adders are based on the EHA design proposed [11]. EHA has three separate modules namely XNOR, Carry, and Sum which have been highlighted separately as seen in Fig. 2a. Among these modules XNOR is shown to have the highest energy consumption [11] and we have carefully designed the approximate XNOR to reduce energy consumption. Our first approximate hybrid adder AHA1 was designed by replacing the exact XNOR with the approximate XNOR in EHA. The circuit diagram of AHA1 is shown in Fig. 2b. The replacement of the XNOR module introduces errors in the output as shown in Table. 2. The second design AHA2 was made by removing the Sum generation module from EHA as shown in Fig. 2c. This introduces errors only in the Sum as seen in Table. 2 and as the Carry generation module is not disturbed, we do not see any error in Carry. The third design is implemented by removing the Sum generation module from AHA1 as seen in Fig. 2d. Since this design also has an approximate XNOR we see that it has errors in both Sum and Carry as seen in Table. 2.

TABLE 2: Truth Table for Proposed Approximate Adders



Inputs			Exact Outputs		Proposed Cells					
					AHA1		AHA2		AHA3	
A	B	C	Sum	Carry	Sum	Carry	Sum	Carry	Sum	Carry
0	0	0	0	0	0	0	1 ✗	0	1 ✗	0
0	0	1	1	0	1	0	1	0	1	0
0	1	0	1	0	0 ✗	1 ✗	0 ✗	0	1	1 ✗
0	1	1	0	1	1 ✗	1	0	1	1 ✗	1
1	0	0	1	0	1	0	0 ✗	0	0 ✗	0
1	0	1	0	1	0	1	0	1	0	1
1	1	0	0	1	0	1	1 ✗	1	1 ✗	1
1	1	1	1	1	1	1	1	1	1	1

The use of FQR -GDI technique lets us simplify complex designs and scale down the number of transistors used in the circuit. Full Adder implementation using FQR-GDI based 4- Transistor XOR gate and AND gate are shown in Figure 3. FQR -GDI based Half Adder consists of six transistors. XOR gate used in the Half Adder uses the 4 Transistors and Full adder consists of 10 transistors. 4-Transistor XOR design is implemented by taking an inverter circuit and common gate node is given with first input signal (A), PMOS source node is given with second input signal (B), NMOS source node is given with complement of second input signal (B'). To generate complement of signal B, a NOT gate is used. AND gate is implemented based on GDI technique where inverter common gate node is used as first input signal (A) port and PMOS, NMOS source nodes are used as second input (B) signal port.

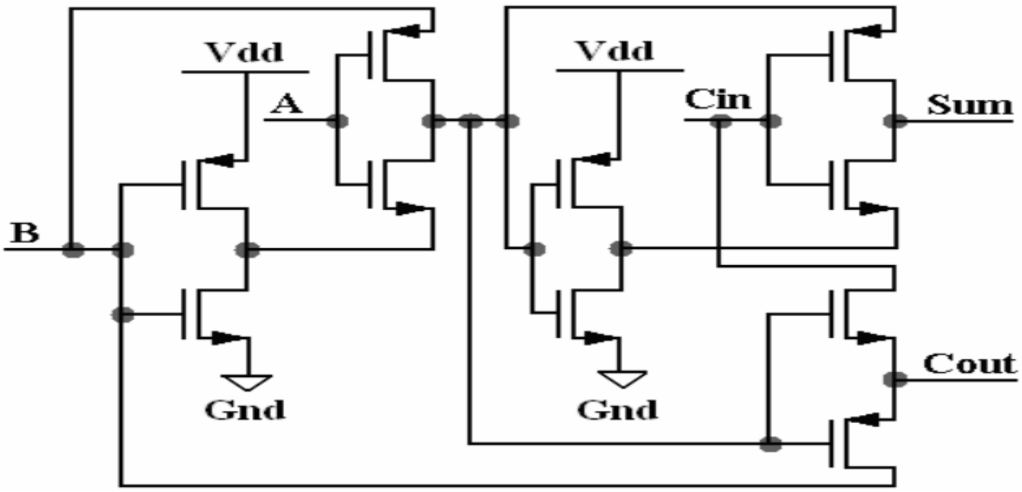


Figure 3: Full Adder using FQR-GDI

Application in image processing

Image processing is an imprecision-tolerant application and would be a suitable application for approximate computing. In this Section, the application of the proposed circuit in image processing is perused and compared with its counterparts. For this purpose, we consider an approximate Gaussian filter, based on the considered approximate designs, to filter out the noisy images. The output image in this application is computed as

$$Y(x,y)=\frac{1}{16}\sum_{i=-1}^{1}\sum_{j=-1}^{1}MASK(i+2,j+2)X(x-i,y-j)$$

where X and Y are the input and output images, and the MASK is

$$MASK=\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

It is notable that the approximate adders are used to perform the addition operation as the main arithmetic operation in this application, and the other operations are performed accurately. We use four methods, and they are all different from each other based on the number of approximate least significant bits (LSB) applied. In the first method, we consider 4 approximate LSBs out of 16, and we carry on to 9 LSBs in the fifth method. Peak signal to noise ratio (PSNR) and structural similarity index metric (SSIM) parameters are used for evaluation and comparison of the approaches under investigation. The results for Cameraman image are given in Tables 1. The results indicate that using the proposed approach leads to better image quality metrics in comparison with the previous approaches under investigation.

Figure 4 shows the output images of different approaches for the cameraman image as an example. Based on the performance and quality metrics evaluated in this section, the proposed approach provides an effective trade-off between efficiency and quality, as it has a simple and energy-efficient structure, while it provides an acceptable quality in applications like approximate image processing. To evaluate the tradeoff attained between the efficiency and quality using each of the full adders under investigation, we use the figure of merit.

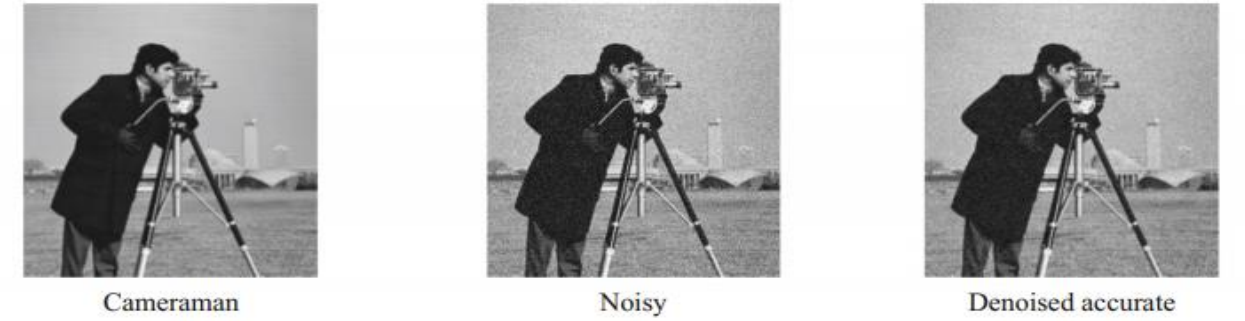


Figure 4: Denoised image using Approximate and accurate full adders

Table 1: PSNR and SSIM comparison

Method	PSNR	SSIM
EHA [11]	27.8	0.65
AHA1	26.8	0.71
AHA2	29.4	0.75
AHA3	28.9	0.83
proposed EFA	32.6	0.91

5. Simulation Results

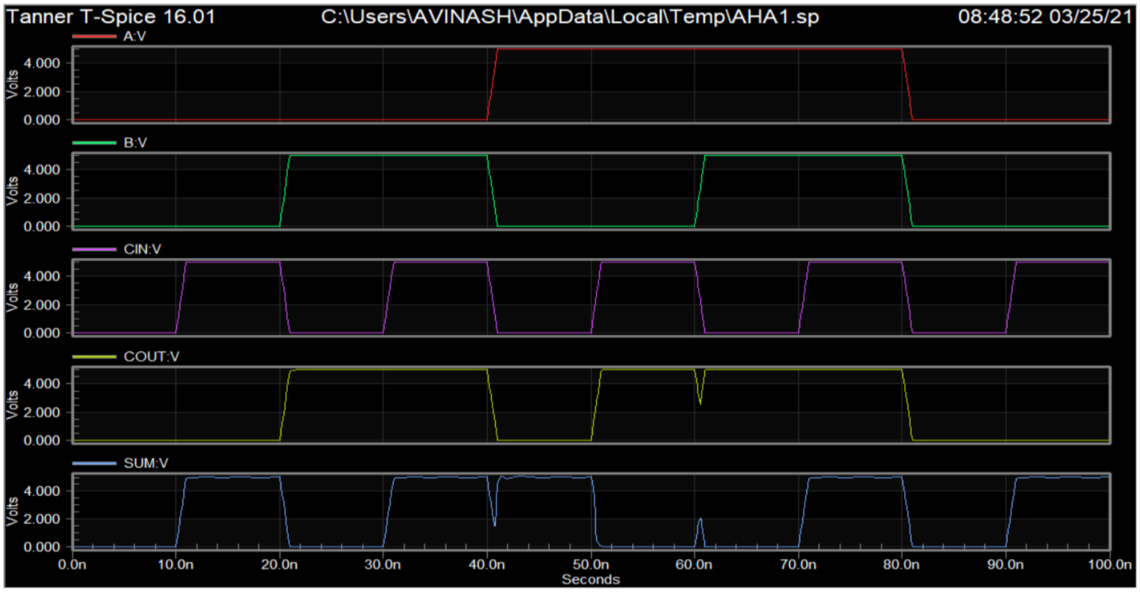


Figure 4: Simulation output of AHA1

The results presented in the above graph are generated according to the table 2 of AHA1 for all combination of inputs respectively.

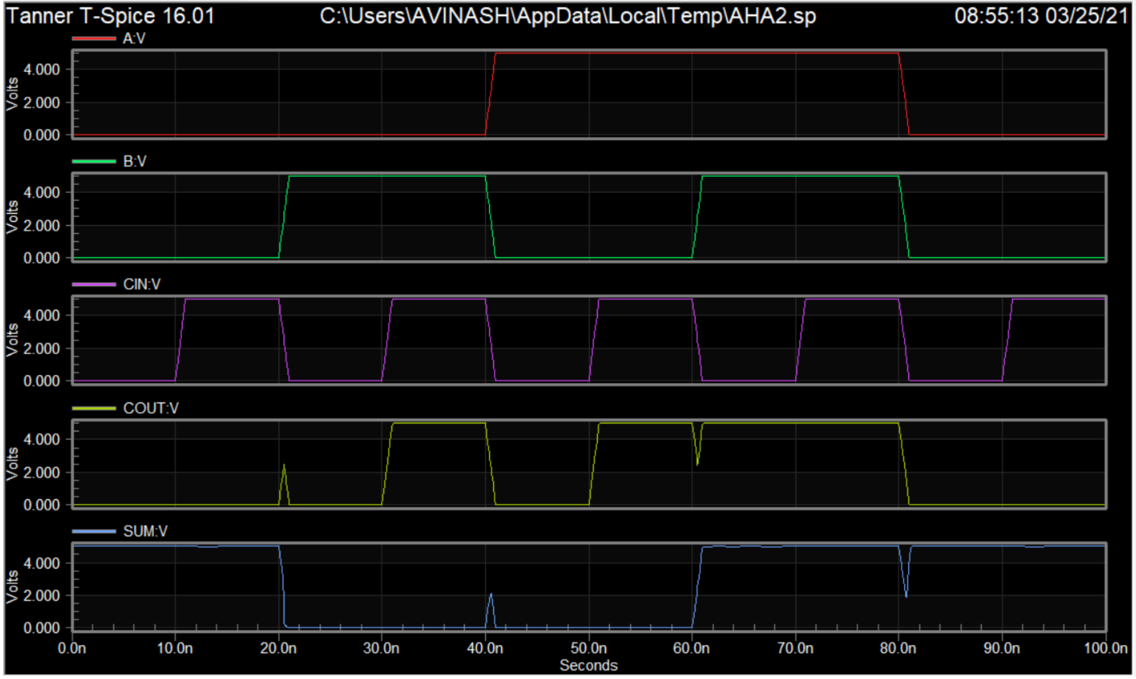


Figure 5: Simulation output of AHA2

The results presented in the above graph are generated according to the table 2 of AHA5 for all combination of inputs respectively.

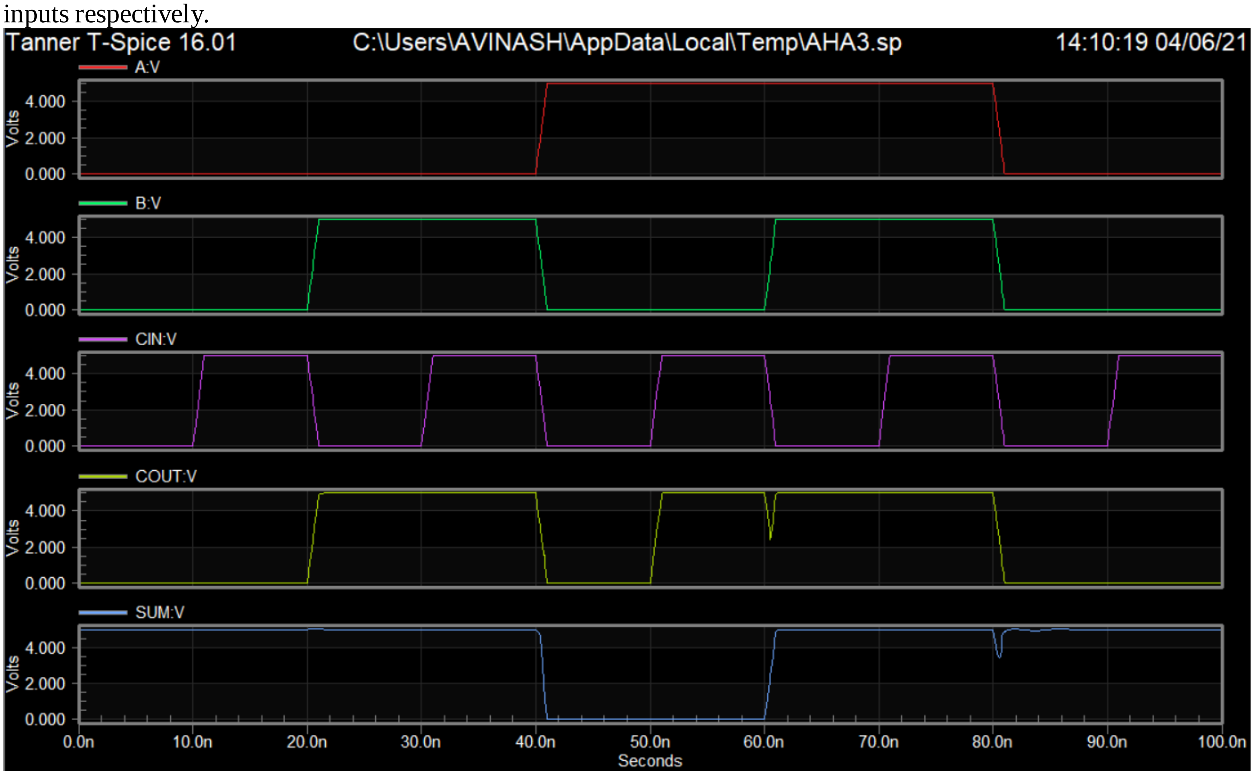


Figure 6: Simulation output of AHA3

The results presented in the above graph are generated according to the table 2 of AHA3 for all combination of inputs respectively.

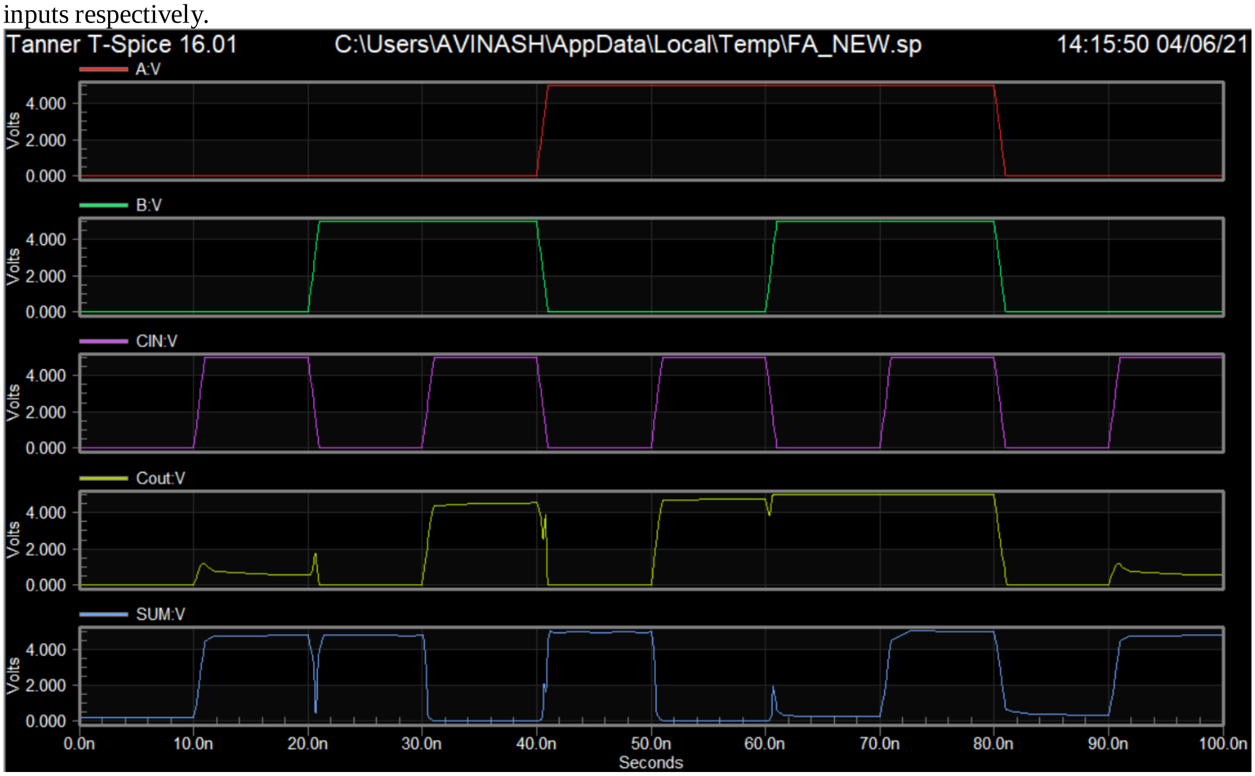


Figure 7: Simulation output of proposed EFA

Table 3: Comparison Table

METHOD	Transistor count	Average power	Max power	Delay
EHA [11]	16	15.2uw	1.11mw	20.05ns
AHA1	15	13.8uw	1.17mw	60.16ns
AHA2	12	5.477uw	0.963mw	20.21ns
AHA3	11	4.90uw	0.510mw	40.28ns
proposed EFA	10	2.90uw	0.421mw	12.34ns

6. CONCLUSION

This paper presents an efficient Adder design in FQR-GDI 25-nm technology on Tanner EDA. This design uses a smaller number of transistors than the conventional Vedic Adder design. By employing the FQR-GDI technique there is an extensive decrease in power and delay of the circuit. There is an improvement of 30% in power and 28% in delay compared to Exact Full Adder using GDI Technique. This design also solves the threshold drops problem of Original GDI cell resulting in better output signals.

REFERENCES

1. A. Morgenshtein, A. Fish, and I. A. Wagner, “Gate-diffusion input (GDI): A power-efficient method for digital combinatorial circuits,” IEEE Trans. Very Large Scale Integr. Syst., vol. 10, no. 5, pp. 566–581, 2002.
2. A. Morgenshtein, I. Shwartz, and A. Fish, “Gate Diffusion Input (GDI) logic in standard CMOS nanoscale process,” 2010 IEEE 26th Conv. Electr. Electron. Eng. Isr. IEEEI 2010, pp. 776–780, 2010.
3. G. Ganesh Kumar and V. Charishma, “Design of High Speed Vedic Adder using Vedic Mathematics Techniques,” Int. J. Sci. Res. Publ., vol. 2, no. 1, pp. 2250–3153, 2012.
4. E. Masurkar and P. Dakhole, “Implementation of optimized vedic adder using CMOS technology,” in International Conference on Communication and Signal Processing, ICCSP 2016, 2016, pp. 840–844.
5. A. Jain and A. Jain, “Design, implementation & comparison of vedic adders with conventional adder,” in 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017, 2018, pp. 1039–1045.
6. A. Kant and S. Sharma, “Applications of vedic adder designs - A review,” in 2015 4th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2015, 2015.
7. P. A. I. Khan and S. K. Dilshad, “Design of 2x2 Vedic Adder u sing GDI Technique,” 2017 Int. Conf. Energy, Commun. Data Anal. Soft Comput., pp. 1925–1928, 2017.
8. S. R. Ghimiray, P. Meher, and P. K. Dutta, “Energy efficient, noise immune 4×4 Vedic adder using semi-domino logic style,” in IEEE Region 10 Annual International Conference, Proceedings/TENCON, 2017, vol. 2017-Decem, pp. 1037–1041.
9. M. S. N. Gadakh and A. S. Khade, “FPGA implementation of high speed vedic adder,” IET Conf. Publ., vol. 2016, no. CP700, pp. 184–187, 2016.
10. S. N. Gadakh and A. Khade, “Design and optimization of 16×16 Bit adder using Vedic mathematics,” in International Conference on Automatic Control and Dynamic Optimization Techniques, ICACDOT 2016, 2017, pp. 460–464.