International Journal of Neural Networks and Advanced Applications
DOI: 10.46300/91016.2022.9.28 Volume 9, 2022

Design of high-performance GDI logic based 8-tap FIR filter at 45nm CMOS technology using Nikhilam Multiplier

[1*]Bindu Swetha Pasuluri, [2]V.J.K. Kishor Sonti

[1]School of Electrical and Electronics Engineering, Sathyabama Institute of Science and Technology Chennai-600119, India
[2]School of Electrical and Electronics Engineering, Sathyabama Institute of Science and Technology Chennai-600119, India

Received: July 24, 2021. Revised: Aaugust 13, 2022. Accepted: September 21, 2022. Published: October 27, 2022.

Abstract- Over the past few decades, advances in IC technology have steadily shrunk feature sizes, necessitating the placement of more operational circuits on every chip. In designing digital circuits, a novel GDI based circuit is indeed the center of consideration, since it requires less power and achieves greater efficiency. GDI-based circuits mimic CMOS transistors but feature fewer transistors with a greater capacity for performance and reliability. This paper investigates the modelling and implementation of a Finite Impulse-Response (FIR) block developed utilizing GDI-based circuits as well as basic blocks. In this study, an eight-tap FIR architecture relying on GDI cells is created. The results reveal that even a FIR architecture with eight taps and GDI delivers reduced power consumption and performance improvement.

Keywords: FIR filter design, Full adder, GDI technique, MAC, Multiplier.

I. INTRODUCTION

The most significant blocks in the designing of DSP comprise FIR units. These are extensively utilized in industries as well as digital systems like automobiles, mobile phones, laptops, speech synthesis, Wireless headphones, and so on. The necessity to build modern electronic system focuses on two key elements, the first of which is determined by technology as well as the other by consumer demands. In terms of innovation sectors, numerous sectors are enhancing their techniques and infrastructure in response to increased sophistication. Improved complexity, greater density to fit transistors on a smaller device size, improved performance, and reduced power consumption are all benefits. Because of market requirements, each innovative problem must always be considered credible and quickly addressed, as losing the window"s market may be highly expensive. Looking through the GDI technology literature reveals there has been no study or initiative on GDI implementation in the DSP component [1, 2].

Furthermore, with prior technologies, multiple transistors are required to construct fundamental blocks. Nevertheless, in the current research, fundamental blocks may be built with as little as two transistors. Multiple advantages of GDI, including improved efficiency, persuaded the author of the most recent study to use GDI with in FIR filter. The TSPICE software of PTM 45nm CMOS Technology had been used to develop and build basic blocks including FIR filters [3, 4].

The following is how this document is organized. Section II describes in detail the underlying principle of Gate Diffusion Input (GDI) as well as its associated activities that are comparable to CMOS inverters. Section III also discusses the system level architectural architecture of an 8-tap digital FIR. Section IV depicts the hardware architecture of fundamental FIR components including the latch, adder, as well as multiplier. Following Section V also includes simulation data and an examination of the FIR architecture. Furthermore, Section VI contains our closing remarks.

II. GDI TECHNIQUE

As an alternative to static CMOS logic, gate diffusion input (GDI) has been employed. The primary contrast between CMOS as well as GDI is that the allocation of supply as well as ground to pMOS and nMOS throughout GDI logic is not fixed. To perform distinct complex logic processes, just two transistors are required. As a result, this is a smaller transistor number circuitry, and the basic GDI circuit shows power savings owing to logic adaptability. However, it concerns from performance fluctuation due to threshold potential drops [5,6].

The input signals in GDI logic are:

G- Inputs to nMOS and pMOS gates

P- input to pMOS drain/source

N- input to nMOS drain/source

The basic AND, OR, XOR gates are designed to maximise swing.

A digital circuit based on GDI consumes less power, has less delay, and area when compared to a conventional CMOS-based device. Basic logic gates have been designed employing GDI technology and mimicked utilizing CMOS 45nm technique in this section.

A. GDI -AND Gate

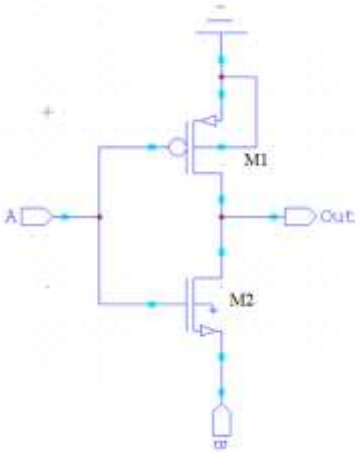

Fig. 1 AND gate circuit using GDI method.

Fig 1 depicts an AND gate that employs the GDI approach. It is made up of two M1 and two M2 transistors, with the A input linked to both the M1 as well as M2 transistor gate terminals. The B input is linked to the M2 transistor source, whereas the M1 transistor source is grounded. The drains of M1 and M2 are connected together to assess the output logic.

B. GDI -OR gate

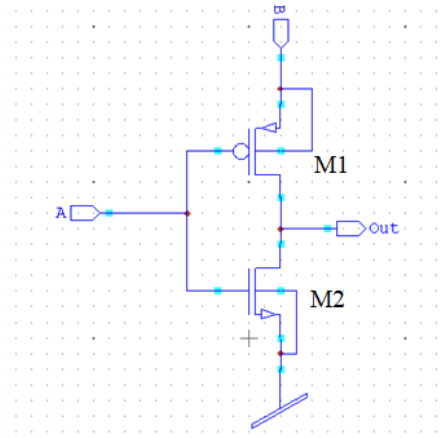

Fig. 2 OR gate based on the GDI approach.

Fig 2 depicts the GDI technique OR gate. It is made up of two M1 as well as M2 transistors, with the gate terminals of both M1 and M2 transistors connected to the A input. Input B has been connected to transistor source M1, while transistor source M2 has been connected to VDD. To evaluate the performance logic, the drains of M1 M2 are attached together.

C. GDI based XOR gate

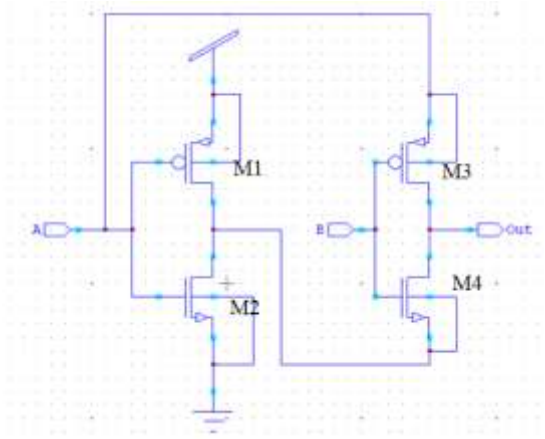

Fig. 3 GDI - XOR gate

Fig 3 depicts an XOR gate based on the GDI approach. It is made up of four transistors, M1, M2, M3, as well as M4, with input A connected to both gate terminals of transistors M1 and M2. Input B is linked to both the M3 as well as M4 transistor gate terminals. Input A is indeed connected to the transistor"s source M3, as well as the transistor"s source M4 has been connected to the drains M1 and M2. M3 as well as M4 drains are linked together to verify the efficiency logic.

III. FIR FILTER

DSP has evolved and essentially replaced previous analogue signal processing methods in a number of applications. Due to the nature of the impulse response, FIR and IIR constitute two significant types of filters. FIR filters are among the two fundamental kinds of digital filters used throughout DSP applications. These are filtering architectures which can be used to computationally design almost every form of frequency response. Because there is no feedback, FIR filters are limited. As a result, if you transmit an impulses (a single spike) through circuit, the output would inevitably turn zero immediately soon even as impulses passes through the filters. The impulse response remains limited in this scenario owing to the absence of feedback with in FIR. In the lack of feedback, the impulsive response is constrained. Therefore, "restricted impulse response" is virtually equivalent with "absence of feedback." Feedback-based DSP filters may alternatively be referred to as "Infinite Impulse Response" (IIR). Both FIR as well as IIR have advantages and disadvantages. FIR filters offer more advantages than disadvantages, hence why these are used more than IIR filters.

IV. FUNDAMENTAL CONCEPT OF FIR

FIR filters are crucial in signal processing applications. These are, in those other terms, the major components of a DSP processor. The current research intends to employ a 8 -tap filter, that will result in a reduction in circuitry complexity. As a consequence, the memory designed to store the coefficients would be cut in half. Several DSP systems use the finite impulse response (FIR) filter to perform various signal operations. An FIR Filter is amongst

the most important components of a DSP system, and it may be theoretically demonstrated using the following formulas. A FIR filter"s fundamental structure consists of a sequence of multiplications preceded by an additions. Consider formula to illustrate a FIR filter function:

$$y[n] = x[n] * h[n] \tag{1}$$

While x, h, as well as y are the corresponding input signal, the transfer function, as well as output signal. The following formula is a straightforward simplified expression of a FIR filter:

$$y[n] = \sum_{i=0}^{i=M} b_i\, x[n-i] = \sum_{i=0}^{i=M} h[i]\, x[n-i] \tag{2}$$

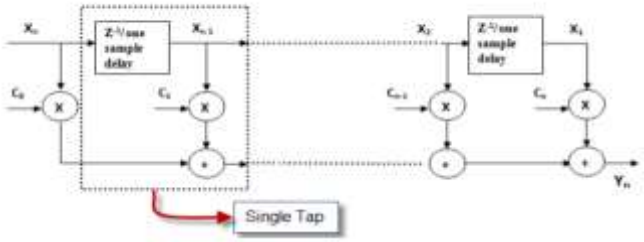

Fig. 4 FIR Structure in Direct Form

Fig 4 depicts the fundamental block diagram of an N-length FIR filter construction. As a consequence of the delays, operations are performed on previous input sequence. Due to the fact that the hN values represent the multiplication coefficients, the output for time n would be the total of preceding delay samples where they are multiplied by the relevant coefficients. Fig 4 provides an explanation of a FIR filter"s tapped delay line architecture. x (n) represents the input sample sequence, h (n) denotes the filter coefficients, whereas m would be the number of taps. An example FIR filter with L=8 and eight input samples is shown. As a result, it is known as an 8-tap filter. Every register provides a unit sampling delay. To generate the output, the delay inputs have been multiplied by its corresponding filter coefficients then joined together. To generate the filter"s output, FIR filter has often been accomplished using just a succession of digital hardware parts such as the delays, the multipliers, as well as adders. An order N FIR filter contains N+1 multipliers, N adders, as well as N delays. Every element of a single tap FIR filter would be detailed in further depth in the next subsection.

V. GDI IMPLEMENTATION OF FIR FILTER

The output of each register is regarded as a tap, this is expressed as X [n-k], wherein (n-k) has been the tap number, with every tap having multiplied the coefficient h (n), and indeed the responses combined together. Using only a few phrases, the terms below demonstrate the functioning of a FIR filter.

Filter Coefficients: A set of constants, sometimes known as tap weights, that are multiplied by obtained delay data. A FIR filter"s filter coefficients should show the filter"s impulse response.

Impulse Response: A collection of FIR coefficients covering each possible speed. Each impulse would"ve been made up of a single sampling with such a frequency of one accompanied by sample with only a value of zero. The impulse response of a FIR filter would"ve been a sequence of filter coefficients.

Tap: A delay and coefficient combination. The quantity of FIR taps, N, often offers data about filter. It also shows the amount of RAM consumed, the number of computations completed, and the quantity of "filtering" which may be done. The more taps a filter has, the stronger its stop band attenuation, a less rippling, as well as the sharper the roll off. Additional taps lead in greater stop band attenuation, lower ripple, and so on.

Multiply and Accumulate (MAC):

A "MAC" would be the act of multiplication of a coefficient by the appropriate delayed sample data as well as aggregating the result in the framework of FIR Filters as shown in Fig 5 and 6. Every tap typically contains a single MAC. The MAC procedure is typically implemented in a one instruction cycle by most DSP microprocessors. These following statements might be useful in gaining a better grasp of the situation:

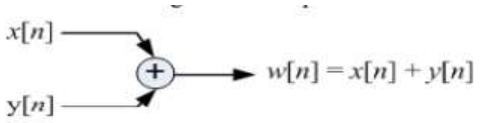

Fig. 5 Single Multiplier Unit

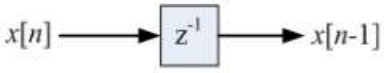

Fig. 6 Single Adder Unit

Delay Line: The collection of memory components that conduct the FIR computation"s Z -1 delay elements. As shown in Fig 7, each tap FIR is made up of three basic components.

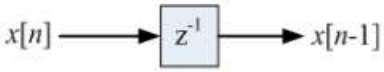

Fig. 7 A Sample Unit Delay

A. GDI implementation of a single tap -FIR filter

The next sections demonstrate the implementation of a Full Adder, Latch, as well as a Multiplier. Various logic types have already been utilised to create entire adder cells. Every contemporary design has advantages and disadvantages. In [7], a highly efficient 10T complete adder cell with 10 transistors is demonstrated. The suggested cell offers minimal power consumption as well as a high operational speed. Furthermore, because to the low number of transistors, it takes up less space. Just at circuit level, a reduced power

improvement is achieved by lowering the number of capacitances in the internal node, removing direct pathways between both the supply voltage as well as the ground, and keeping a low switching frequency in the circuitry. In comparison to the preceding several kinds of transistor adders, the 14T Adder proposed in [8] uses much less power within range of microwatts, has faster speed, as well as minimises the threshold loss issue. Complementary pass transistor logic (CPL) [9] refers to a kind of low-power CMOS logic devices. CPL concept predicated on the use of NMOS networks only for logic functional representations and the absence of the PMOS latch. The use of just NMOS transistors minimises input loads. CPL would have complementary outputs and inputs, with pass transistors acting as both pull-up as well as pull-down devices [10]. Because the pass transistor"s threshold voltage is substantially lower than the voltage supply, the pass transistor"s output terminal voltage must always be amplified using CMOS inverters. The output inverters have a great driving capacity. The primary distinction between pass transistor logic and complementary CMOS logic is that the supply side of the pass logic transistor network is now coupled to input signals rather than power lines. The benefit is that the logic function may be performed with a single pass transistor network (either pMOS or nMOS), resulting in fewer transistors and a lower input load. In contrast, pass transistor logic suffers from a fundamental threshold voltage drop issue. When "1" is delivered via a nMOS, the output is a weak logic "1," whereas "0" is sent through a pMOS, the output is a weak logic "0." As a result, output inverters are used to assure driveability.

The complementary CMOS full adder (CCMOS) [11] is a standard CMOS structure with pull-up and pull-down transistors that provides full swing output and high driving capabilities. The transmission-function full adder (TFA) [12] and the transmission-gate full adder (TGA) [13] are two further options. Another full adder design, as shown in [11], is the Hybrid full adder. The majority of these systems depend on the simultaneous creation of XOR and XNOR outputs, which are then routed to either a transmission-gates or a static CMOS final stage to generate Sum and Carry outputs.

B. Design of GDI based 1-bit full adder

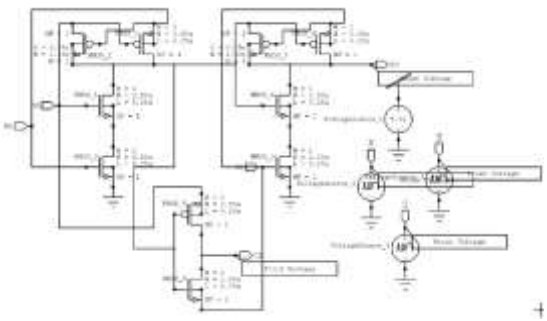

Fig. 8 10T GDI Logic based Adder

Fig 8 demonstrates the design of the proposed binary fast full adder at the transistor level. It is made up of two modules that use the GDI method. The module M1 generates sum. The carry function is realised in the module M2. The circuit generates output carry from an input carry (Cin) signal. This circuit is tiny enough that it may be represented as a simple primitive. Circuity based on the GDI approach generate the required functionalities in the proposed complete adder cell. Because of their resistance to voltage scaling as well as transistor sizing, these circuits can run consistently at very low voltage. In addition, the output inverters give sufficient power to the cascaded cells.

1. D-Latch:

D-Latch would be used for unit sampled delay. The D-Flip-Flop is one of the most important state-holding components. Each coefficient in the current design proposal for an eight-tap filter consists of four bits, and there"s only four coefficients since the filter is often produced with linear phase. The memory was built using an 8-bit register, with every cell comprised of a D-Flip-Flop. Fig 9 depicts the D-Flip-GDI Flop"s circuit configuration.

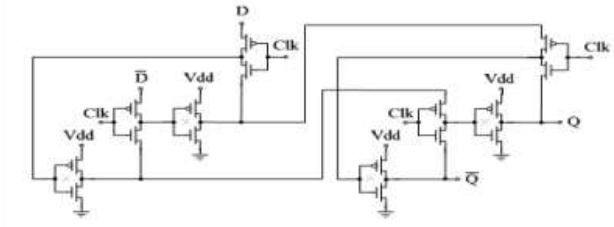

Fig. 9 GDI Logic based D-Latch

2. Multiplier

A 4-bit array multiplier has been employed in the current architecture. The one-to-one mapping of the topological link between array multiplication hardware architecture and manual multiplication simplifies implementation. Although there are higher effective multiplier architectures in terms of performance as well as power, we picked an array multiplier because of its simplicity. Fig 10 displays a four-bit array multiplier which has been enlarged from a two-bit multiplier.

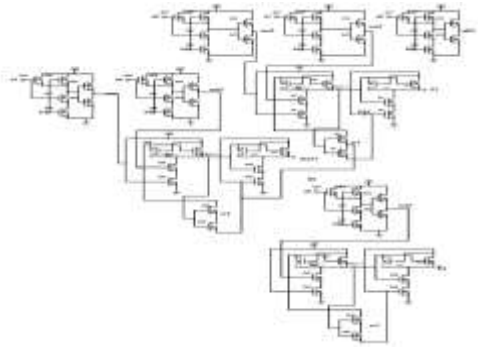

Fig. 10 A 2-bit Array multiplier extended to 4-bit

C. GDI logic based 8-Tap FIR design

Fig 11 illustrates a Direct Form FIR Filter with each tap comprising a D-Latch, Multiplier, and Full Adder-Latch, as well as a 4-bit input and 4-bit output. Multiplier is composed of two 4-bit inputs and two 8-bit outputs. There were eight successive taps in this project, with each D-Latch and Adder flowing into the next tap.

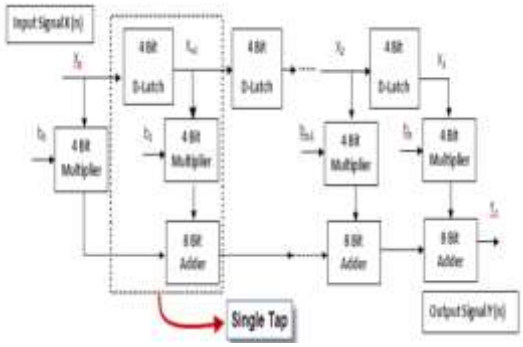

Fig. 11 Direct form FIR Implementation

VI. SIMULATION RESULTS

In this study, we utilised w/L= 2.21 to replicate the W as well as L values assigned to everytransistors in GDI technology. The simulation results of GDI based full adder, multiplier and FIR filter is shown in Figs 12, 13, 14 respectively.

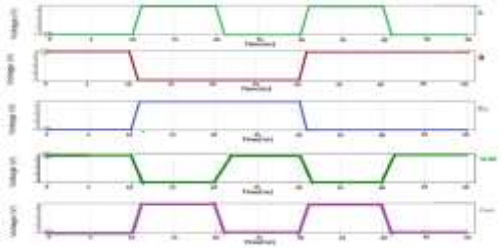

Fig. 12 Simulation results of GDI based full adder

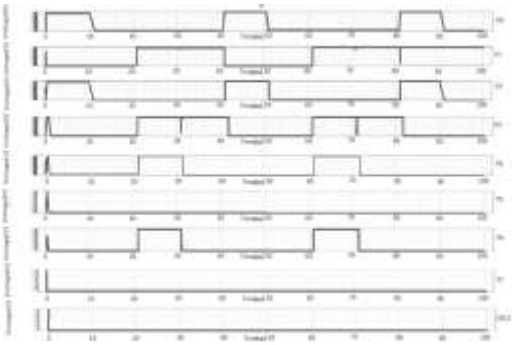

Fig. 13 Simulation of GDI based Array Multiplier

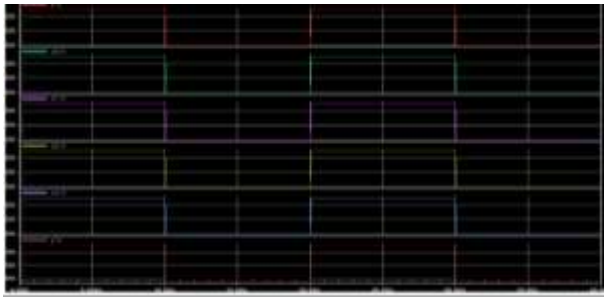

Fig. 14 Simulation of GDI based FIR Filter by using Direct Form

Table 1: Performance comparison of proposed adder

Full adder	Ref	No. of Transistors	Average Power(µW)			Delay (ps)			PDP		
			0.4V	0.8V	1.2V	0.4V	0.8V	1.2V	0.4V	0.8V	1.2V
CPL	16	32	0.488	1.72	3.89	612.1	84.8	37.3	298.7	145.9	145.1
CCMOS	17	28	0.159	0.68	1.91	809.3	125.8	39.7	145.7	85.5	75.8
TGA	18	20	0.163	0.64	1.41	893.4	139.7	58.3	151.9	89.4	82.2
TFA	19	16	0.155	0.61	1.33	957.5	145.6	66.8	155.1	88.8	88.8
24-T	20	24	0.202	0.76	1.68	908.3	128.4	65.9	183.5	97.6	110.7
DPL	21	22	0.256	0.87	2.11	835.6	91.9	45.6	213.9	79.9	96.2
SRCPL	21	20	0.193	0.7	1.79	869.6	132.3	50.4	167.8	104.5	90.2
New-HPSC	22	24	0.217	0.77	2.04	998.5	193.6	71.5	216.7	149.1	145.9
Hybrid 1	23	24	0.267	0.78	2.31	932.4	189.1	60.2	248.9	145.6	139
Hybrid 3	24	22	0.169	0.68	1.53	811.8	101.3	48.6	146.9	68.9	74.4
Hybrid 4	25	16	0.113	0.44	0.98	675.3	81.6	38.7	76.3	35.9	37.9
Hybrid 5	26	21	0.146	0.58	1.31	697.5	96.8	43.9	108.1	56.1	57.5
Hybrid 6	26	23	0.133	0.54	1.35	683.4	81.4	35.1	98.4	43.9	47.4
GDI D1	27	18	0.127	0.46	1.09	599.8	98.8	31.8	76.17	43.5	34.7
GDI D2	27	22	0.165	0.63	1.49	547.6	77.3	28.6	90.4	48.7	42.6

GDI D3	27	21	0.152	0.61	1.32	708.3	90.53	39.7	114.7	55.2	52.4
Scalable Hybrid	28	22	0.129	0.48	1.17	523.8	65.7	25.3	67.6	31.5	29.6
Proposed		10	0.005874	0.01471	0.01547	585.38	97.053	31.52	3.4385	1.4276	0.4876

Table 2: Performance Comparison of Multiplier

	Average Power (W)	**Delay (ns)**
Wallace tree [29]	**0.6791m**	**72.47**
RCW [30]	**0.6740m**	**72.37**
CBW [31]	**0.6080m**	**73.38**
[32]	**0.1475m**	**72.21**
[32]	**0.1477m**	**72.40**
Proposed Multiplier	**1.96µ**	**47.89**

Table 3: Performance Comparison of FIR

	Power (mW)	**Delay (ns)**
Conventional FIR Filter [33]	**27**	**108.92**
Proposed	**1.21**	**93.32**

From the tables 1,2 and 3, it is evident that proposed adder, multiplier and FIR filter yields better results in terms of power, delay.

VII. CONCLUSION

This research designed and developed an 8-tap digital FIR filter. GDI circuits have been used to create fundamental cells like as adder, a D-latch, as well as multiplier. The simulation results demonstrate that the propagation delay as well as power consumption substantially reduced as compared to the current circuits. As per the results of the literature and analysis for this research, GDI technology is applicable to a wide range of fields. GDI was also used in a number of Signal processing applications known as FIR. Finally utilised a digital FIR filter GDI technique with eight taps. In addition, area, power, as well as latency have been taken into account. Using CMOS 45nm technology, the simulation was conducted.

References

[1] N. S. Rai, P. S. B. S, M. Y. P, A. P. Chavan, and H. V. R. Aradhya, "Design and implementation of 16 tap FIR filter for DSP Applications," 2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAECC), 2018.

[2] R. A. S. Durai singh, A. J. Deva, J. P. Jeganatha, M. A. B. Saleem, Tharmavel Abirami, "Modelling and Analysis of Modified Baugh-Wooley Multiplier Using Gate Diffusion Input and Improved Shannon Adder," International Journal of Pure and Applied Mathematics, vol. 118, no. 22, 2018, pp. 773-777.

[3] B. K. Mohanty, and P. K. Meher, "A High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications," IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, 24(2), pp. 444–452.

[4] Mohan Shoba and Rangaswamy Nakkeeran, "GDI based full adders for energy efficient arithmetic applications," Engineering Science and Technology, an International Journal, 19(1), 2016, pp. 485–496.

[5] A. Morgenshtein, A. Fish and I. A. Wagner, "Gate-Diffusion Input (GDI)- Annual power efficient method for digital circuits," Proc. 14th Annual IEEE Int. ASIC/SOC Conf. 2001, pp. 39-43.

[6] A. Morgenshtein, A. Fish, and I. A. Wagner, "Gate-Diffusion Input (GDI): A Power Efficient Method for Digital Combinatorial Circuits," IEEE Transactions on Very LargeScale Integration (VLSI) Systems, vol. 10, no. 5, 2002.

[7] H. T. Bui, Y. Wang, and Y. Jiang, "Design and analysis of lowpower 10-transistor full adders using novel XOR-XNOR gates," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. vol. 49, no. 1, Jan. 2002, pp. 25–30.

[8] M. Vesterbacka, "A 14-transistor CMOS full adder with full voltage swing nodes," in Proc. IEEE Workshop Signal Processing Systems, Oct. 1999, pp. 713–722.

[9] A. P. Chandrakasan and R. W. Brodersen, Low Power Digital CMOS Design. Norwell, MA: Kluwer, 1995.

[10] K. Yano et al., "A 3.8 ns CMOS 16x16 multiplier using complimentary pass-transistor logic," IEEE J. Solid-State Circuits, vol. 25, no. 2, Apr. 1990. pp. 388–395.

[11] Chip-Hong Chang, Jiangmin Gu, and Mingyan Zhang, "A Review of 0.18-_m Full Adder Performances for Tree Structured Arithmetic Circuits," IEEE Trans. On Very LargeScale Integration (VLSI) Systems, vol. 13, no. 6, June 2005, pp. 686–695.

[12] N. Zhuang and H. Hu, "A new design of the CMOS full adder," IEEE J. Solid-State Circuits, vol. 27, no. 5, May 1992. pp. 840–844.

[13] C. Chang, J. Gu, M. Zhang, "Ultra Low-Voltage Low-Power CMOS 4-2 and 5-2 Compressors for Fast Arithmetic Circuits," IEEE Transactions on Circuits and Systems, vol. 51, no. 10, Oct. 2004. Jan, pp. 1985-1997.

[14] M. Rabaey, A. Chandrakasan and B. Nikolic, "Digital Integrated Circuit. Prentice Hall Publications: India, 2006, Chapter 11, pp. 590.

[15] F. Moradi.Dag. T. Wisland, H. Mahmoodi, S. Aunet, T. Vucao, A. Peiravi, "Ultra Low Power Full Adder Topologies" San Francisco State Univ.2009 IEEE, pp. 3160.

[16] R. Zimmermann and W. Fichtner, "Low-power logic styles: CMOS versus pass-transistor logic," IEEE J. Solid-State Circuits, vol. 32, no. 7, Jul. 1997, pp. 1079–1090.

[17] N. H. E. Weste and D. M. Harris, “CMOS VLSI Design: A Circuits and Systems Perspective, 4th ed. Boston, MA, USA: Addison-Wesley,” 2010.
[18] A. M. Shams, T. K. Darwish, and M. A. Bayoumi, “Performance analysis of low-power 1-bit CMOS full adder cells,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. vol. 10, no. 1, Feb. 2002. pp. 20–29.
[19] M. Alioto, G. Di Cataldo, and G. Palumbo, “Mixed full adder topologies for high-performance low-power arithmetic circuits,” Microelectron. J. vol. 38, no. 1, Jan. 2007. pp. 130–139.
[20] C. K. Tung, Y. C. Hung, S. H. Shieh, and G. S. Huang, “A low-power high-speed hybrid CMOS full adder for embedded system,” in Proc. IEEE Conf. Design Diagnostics Electron. Circuits Syst. vol. 13. Apr. 2007, pp. 1–4.
[21] M. Aguirre-Hernandez and M. Linares-Aranda, “CMOS full-adders for energy-efficient arithmetic applications,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. vol. 19, no. 4, Apr. 2011. pp. 718–721.
[22] S. Goel, A. Kumar, and M. A. Bayoumi, “Design of robust, energyefficient full adders for deep-submicrometer design using hybrid-CMOS logic style,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst. vol. 14, no. 12, Dec. 2006, pp. 1309–1321.
[23] I. Hassoune, D. Flandre, I. O”Connor and J. Legat, "ULPFA: A New Efficient Design of a Power-Aware Full Adder," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 8, , Aug. 2010. pp. 2066-2074
[24] R. F. Mirzaee, M. H. Moaiyeri, H. Khorsand and K. Navi, “A New Robust and Hybrid High-Performance Full Adder Cell,” Journal of Circuits, Systems and Computers, vol. 20, no. 4, 2011. pp. 641-655.
[25] M. C. Parameshwara and H. C. Srinivasaiah, “Low-Power Hybrid 1-Bit Full Adder Circuit for Energy Efficient Arithmetic Applications,” Journal of Circuits, Systems and Computers, vol. 26, no. 1, 2017, pp. 1-15.
[26] P. Kumar and R. K. Sharma, “An Energy Efficient Logic Approch to Implement CMOS Full Adder,” Journal of Circuits, Systems and Computers, vol. 26, no. 5, 2017. pp. 1-20.
[27] M. Shoba and R. Nakkeeran, “GDI based full adders for energy efficient arithmetic applications,” Engineering Science and Technology, an International Journal, vol. 19, no. 1, 2016. pp. 485-496.
[28] Mehedi Hasan, M. D. Jobayer Hossein, Mainul Hossain, U. Hasan Zaman, Senior Member, IEEE and Sharnali Islam,” Design of a Scalable Low-Power 1-bit Hybrid Full Adder for Fast Computation,” IEEE Transactions on Circuits and Systems II: Express Briefs 1, 2019.
[29] C. S. Wallace, “A suggestion for a fast multiplier,” IEEE Trans. Elect. Comput. vo. 13, no. 1, 1964, pp. 14–17.
[30] R. S. Waters, E. E. Swartzlander, “A reduced complexity wallace multiplier reduction,” IEEE Trans. Comput. vol. 59, no. 8, 2010, pp. 1134–1137.
[31] S. Asif, Y. Kong, “Analysis of different architectures of counter based wallace multipliers,” In Proceedings of the 2015 Tenth International Conference on Computer Engineering Systems (ICCES), 2015, pp. 139–144.
[32] S. Asif, Y. Kong, “Design of an algorithmic wallace multiplier using high speed counters,” In Proceedings of the 2015 Tenth International Conference on Computer Engineering Systems (ICCES), 2015, pp. 133–138. https://doi.org/10.1109/ICCES.2015.7393033
[33] C. Fritz, A. T. Fam, “Fast binary counters based on symmetric stacking,” IEEE Trans. Very Large Scale Integ. (VLSI) Syst. vol 25, no. 10, 2017, pp. 2971–2975. https://doi.org/10.1109/TVLSI.2017.2723475
[34] M. Madheswaran and D. Malathi, “Design of Low Power FIR Filter Using 6T Full Adder,” Australian Journal of Basic and Applied Sciences, vol. 8, no. 18, December 2014, pp. 72-78.
[35] B.S. Pasuluri, V.J.K. Kishor Sonti, “Performance Analysis of 8-Bit Vedic Multipliers Using HDL Programming. In: Kumar A. Paprzycki M. Gunjan V. (eds) ICDSMLA 2019. Lecture Notes in Electrical Engineering, vol. 601. Springer, Singapore. 2020. https://doi.org/10.1007/978-981-15-1420-3_114
[36] Binduswetha Pasuluri, V. J. K. Kishor Sonti, “Design of Vedic multiplier-based FIR filter for signal processing applications,” Journal of Physics: conference series IOP publishing Ltd.1921(2021) 012047 doi: 10.1088/1742-6596/1921/1/012047. impact factor: 1.1. ISSN 1742588, pp. 17426596. https://iopscience.iop.org/article/10.1088/1742-6596/1921/1/012047
[37] Binduswetha Pasuluri, V. J. K. Kishor Sonti, “Design and Analysis of Instrumentation amplifier using 45nm technology,” in Informatica Journal 2021, vol. 32, no 11, year 2021, ISSN:0868-4952. IF: 1.928. pp. 37-48.
[38] B. S. Pasuluri and V. J. K. K. Sonti, "Design and Performance Analysis of Analog Filter and Digital Filter with Vedic Multipliers in Bio-Medical Applications," 2022 International Conference for Advancement in Technology (ICONAT), 2022, pp. 1-8. doi: 10.1109/ICONAT53423.2022.9725831.https://ieeexplore.ieee.org/document/9725831
[39] Binduswetha Pasuluri, V. J. K. Kishor Sonti, 1570780638 “Reducing computational complexity in digital circuit designing using Ancient Mathematics” in IEEE Delhi Section Conference DELCON 2022, IEEE Explore.978-1-6654-5883-2/22. https://ieeexplore.ieee.org/document/9753237/
[40] Binduswetha Pasuluri, V. J. K. Kishor Sonti, “FPGA implementation of enhanced throughput design of AES architecture using nikhilam sutra,” in Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/Journal of Tianjin University Science and Technology. vol. 55, no. 06, 2022, ISSN : 0493-2137. https://tianjindaxuexuebao.com/details.php?id=DOI:10.17605/OSF.IO/5MEJW