Journal of Engineering Sciences
Vol 11, Issue 1, JAN/2020
ISSN NO:0377-9254

4-Bit Arithmetic and Logic Unit design with optimized area and less power consumption by Using GDI Technique

[1]Dr.P PRASANNA MURALI KRISHNA, [2]Dr.P SRINIVASULU , [3]BUKKA PRAVEEN KUMAR, [4]BANDI VENKATA SAI KRISHNA REDDY, [5]BANDI RAMANJANEYA REDDY, [6]ALLU PAVAN KUMAR REDDY

[1]Professor , [2]Professor ,[3,4,5,6]B.Tech Student

Department of ECE

Kits college, Markapur

Email:pprasannamurali@gmail.com,Psrinivas.vlsi@gmail.com

ABSTRACT

Power dissipation and circuit area are the main problems in the electronics industry, this paper presents a 4-bit Arithmetic Logic Unit (ALU) model using Full-Swing GDI Method, which considers an effective method for low power digital design while reducing the circuit area compared to other logic types. The suggested ALU design consists of 2x1 Multiplexer, 4x1 Multiplexer and Full Adder cell with lower power to perform arithmetic and logic activities. The simulation was conducted using the 65 nm TSMC method of Cadence Virtuoso. The results show that the design proposed consumes less power with fewer transistors, thus achieving full swing activity compared to the previous research.

I. INTRODUCTION

Arithmetic circuits, like adders and multipliers, are one of the basic components in the design of communication circuits. Recently, an overwhelming interest has been seen in the problems of designing digital systems for communication systems and digital signal processing with low power at no performance penalty. Designing low power high-speed arithmetic circuits requires a combination of techniques at four levels; algorithm, architecture, circuit and system levels. This thesis presents layout and simulations of a multiplication algorithm, which is suitable for high-performance and low–power applications [2]. Digital multipliers are the most commonly used components in many digital circuit designs. They are fast, reliable and efficient components that are utilized to implement any operation. Depending upon the arrangement of the components, there are different types of multipliers available. Particular multiplier architecture is chosen based on the application. The power dissipation in a multiplier is a very important issue as it reflects the total power dissipated by the circuit and hence affects the performance of the device

Most digital signal processing (DSP) systems incorporate a multiplication unit to implement algorithms such as correlations, convolution, and filtering and frequency analysis. In many DSP algorithms, the multiplier lies in the critical delay path and ultimately determines the performance of the algorithm. The speed of multiplication operation is of great importance in DSP as well as in the general processors today, especially since the media processing took off. In the past, multiplication was implemented generally with a sequence of addition, subtraction and shift operations. Recently, many multiplication algorithms have been invented and developed, each having pros and cons in different fields. The multiplier is a fairly large block of a computing system. The amount of circuitry involved is proportional to the square of its resolution; i.e. a multiplier of size n bits has $O(N^2)$ gates [1]. For multiplication algorithms performed in DSP applications, latency and throughput are the two major constraints from delay perspective. Latency is the real delay of computing a function, a measure of how long after the inputs to a device are stable, is the final result available on outputs. Throughput is the measure of how many multiplications can be performed in a given period of time. Multiplier is not only a high-delay block but also a significant source of power dissipation. That's why, if one also aims to minimize power consumption, it is of great interest to identify the techniques to be applied to reduce delay by using

various delay optimizations.

Minimizing power consumption for digital systems involves optimization at all levels of the design. This optimization includes the technology used to implement the digital circuits, the circuit style and topology, the architecture for implementing the circuits and at the highest level the algorithms that are being implemented. The most important technology consideration is the threshold voltage and its control, which allows the reduction of supply voltage without significant impact on logic speed. Since energy is consumed only when capacitance is being switched, power can be reduced by minimizing this capacitance through operation reduction, choice of number representation, exploitation of signal correlations, resynchronization to minimize glitching, logic design, circuit design, and physical design.

Design Considerations in Integrated Circuits

After guaranteeing correct digital functionality, the primary consideration for system designers has always been speed. A circuit is specified to operate at a particular delay, otherwise the entire system may not work; further reduction is beneficial but not strictly necessary. Other factors may have equal or greater importance than power dissipation; area of implementation and reliability issues are subjects which designer must take into account. It's worth to note that power reduction techniques are not necessarily negatively correlated to delay reduction. For example, one method to reduce delay in a circuit's critical path is to upsize the driving strength of gates, which results in increased power reduction. However, reducing interconnect capacitance, which is another way to lower delay, reduces both power and delay. Generally, great power savings can be achieved if delay is not an issue, but optimizing power without delay consideration is insignificant.

Why Low Power?

Power dissipation limitations come in two ways. The first is related to cooling considerations when implementing high performance systems. High-speed circuits dissipate large amounts of energy in a short amount of time, generating a great deal of heat. This heat needs to be removed by the package on which integrated circuits are mounted. Heat removal may become a limiting factor if the package cannot sufficiently dissipate this heat or if the required thermal components are too expensive for the application.

The second failure of high-power circuits relates to the increasing popularity of portable electronic devices. Laptop computers, portable video players and cellular phones all use batteries as a power source. These devices provide a limited time of operation before they require recharging. To extend the battery life, low power operation is desirable in integrated circuits.

II. LITERATURE REVIEW

Circuit design plays an important role in the design of digital circuits like multiplier. First, to guarantee the multiplier to work at the desired clock rate, the designer has to know the delay of the critical path and the required time of inserting a pipeline stage. Second, to reduce the area of the multiplier, several architectures of adders are investigated. Circuit analysis helps the designer verify the functions and performances of the adders. The architecture of the adder has to be determined first. Then the number of the pipeline stages can be decided by the speed of the adder. The size of the multiplier should be as small as possible if all the requirements can be met. Fast arithmetic requires fast circuits. Fast circuits require small size, to minimize the delay effects of wires. Small size implies a single chip implementation, to minimize wire delays, and to make it possible to implement these fast circuits as part of a larger single chip system to minimize input/output delays. The increasing demand for low-power VLSI asks, among others, for power efficient logic styles [2]. Performance criteria for logic styles are circuit speed, circuit size, power dissipation, and wiring complexity as well as ease-of-use and generality of gates in cell-based design techniques. Dynamic logic styles are often a good choice for high-speed, but not for low-power circuit implementations due to the high node activity and large clock loads [2]. This chapter focuses review of various logic styles suitable for low power.

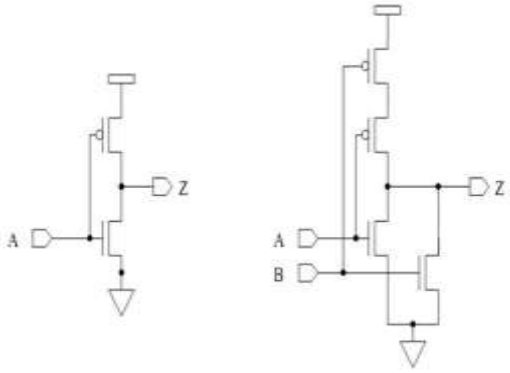
CMOS LOGIC STRUCTURE

Today CMOS (Complementary Metal Oxide Semiconductor) [11] is the primary technology in the Semiconductor industry. Most high speed microprocessors are implemented using CMOS. Contemporary CMOS technology is characterized by:Small minimum sized transistors, allowing for dense layouts, although the interconnect limits the density. Low Quiescent Power - The power consumption of conventional CMOS circuits is

largely determined by the AC power caused by the charge and discharge of capacitances:

$$\text{Power} = CV^2f \quad (3.1)$$

Where f is the frequency at which a capacitance is charged and discharged. As the circuits get faster, the frequency goes up as does the power consumption.Relatively simple fabrication process.
Large required transistors - In order to drive wires quickly, large width transistors are needed, since
Large voltage swings - Typical voltage swings for contemporary CMOS are from 3.3 to 5volts (with even smaller swings on the way). All other things being equal, equation 3.2 says that a smaller voltage swing will be proportionally faster.
Good Noise Margin
Figure 3.1 (a) shows a CMOS inverter. There is a pull-up PMOS transistor and a pull-down NMOS transistor. The steady state output will be independent of the ratio of the pull-up and pull-down transistor sizes. Because of this, CMOS complementary logic does not have to worry about signal degradation problems in pass-transistor logic. Because the power-to-ground path only closes during the transition, it almost consumes no static power. The CMOS complementary gate has two function determining blocks an n-block and a p-block. There are normally 2n transistors in an n-input gate.



(a) (b)

Figure 1: (a) CMOS Inverter (b) CMOS NOR Gate

III. EXISTING METHODS

Vedic Wallace Multiplier

In general, Wallace tree addition uses full adders to extensively reduce the partial products.

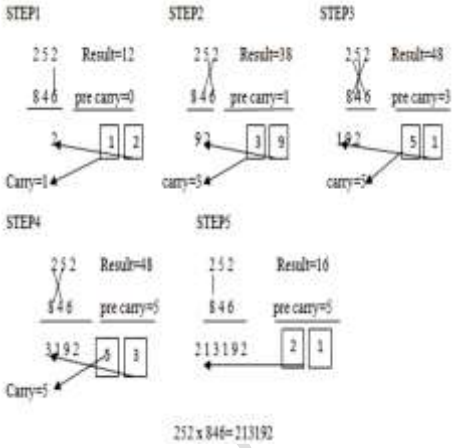


Figure 2 Multiplication of Two Decimal Numbers: 252x846

When the critical path is compared between the critical path in 4 bit conventional and Vedic multiplier, for a 4-bit multiplier, 4 partial products will be generated, as shown in Figure 2 and are named as p0 to p3. For Wallace tree multiplier, a 3:2 reduction is used, so that the partial products are reduced from 4 to 3. The Delay in critical path is given by the addition of 3 full adder sums, 2 full adder carry, and half adder carry. The critical path for Vedic mathematics as shown in Figure 3, is given by 2FAS is reduced by 3HAS and in terms of XOR gates, Vedic-Wallace uses 3XOR gates instead of 4XOR i.e., less carry propagation delay than the conventional method. Hence, Vedic-Wallace has a variable improvement over design ware depending upon the number of bits in multiplication

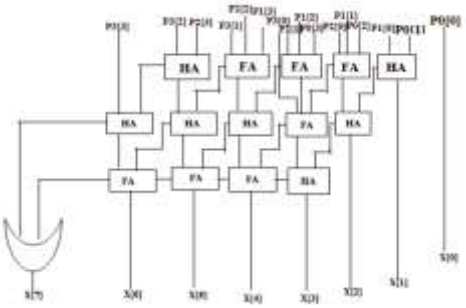


Figure 3 4x4 Multiplier using Wallace Tree

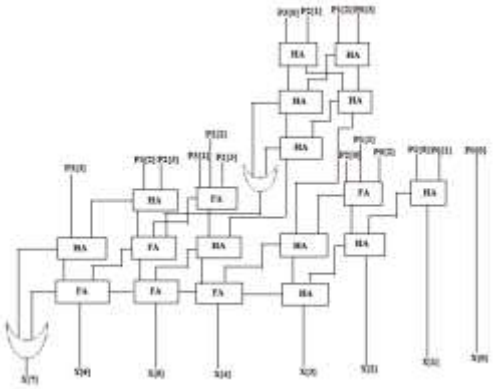


Figure 4. 4x4 Multiplier using Vedic Reduction

Different Multiplier Architectures

The hardware architecture of 2x2, 4x4, 8x8, 16x16, 32x32 bit Vedic Wallace multiplier (VW) modules are displayed in below sections. Here, "Urdhva-Tiryakbhyam" (Verticallyand crosswise) sutra is used to propose such an architecture for the multiplication of two binary numbers. The beauty of Vedic Wallace multiplier is that, here partial product generation and additions are done concurrently. Hence, it is a well adapted parallel processing. The features make it more attractive for binary multiplications. This reduces delay and this is the primary motivation behind this work.

3.3 Vedic Wallace Multiplier for 2x2 Bit Module

The method is explained for two, 2-bit numbers A and B where A=aa and B=b b as shown in Figure 4. Firstly, the 1 0 1 0 least significant bits are multiplied which gives the least significant bit of the final product (Vertical). Then, the LSB of the multiplicand is multiplied with the next higher bit of the multiplier and added to, the product of LSB of the multiplier and the next higher bit of the multiplicand (crosswise). The sum gives the second bit of the final product and the carry is added to the partial product obtained by multiplying the most significant bits to give the sum and carry. The sum is the third corresponding bit and the carry becomes the fourth bit of the final product.

s0=a0b0 (1)

c1s1=a1b0+a0b1 (2)

c2s2=c1+a1b1 (3)

The final result will be c2s2s1s0. This multiplication method is applicable for all the cases. The 2x2 bit Vedic Wallace multiplier is implemented by using four input AND gates along with two half-adders. In the same way, 4, 8, 16, 32 and N bit multipliers are designed with a little modification

Vedic Wallace Multiplier for 4x4 Bit Module

The 4x4 bit Vedic Wallace multiplication unit is furtherrealized by incorporating four similar modules of 2x2 multipliers. The processing in the form of a block diagram is depicted in Figure 4 for the 4x4 multiplier [8].

Vedic Wallace Multiplier for 8x8 Bit Module

The 8x8 Vedic Wallace Multiplier modules are realized using four 4x4 multiplier modules. The processing of 8x8 multiplier based on Vedic Wallace methodology is depicted in Figure 5. Demonstrating with an example of the two digits consisting of 8 bits, the output is obtained in 16 bits length

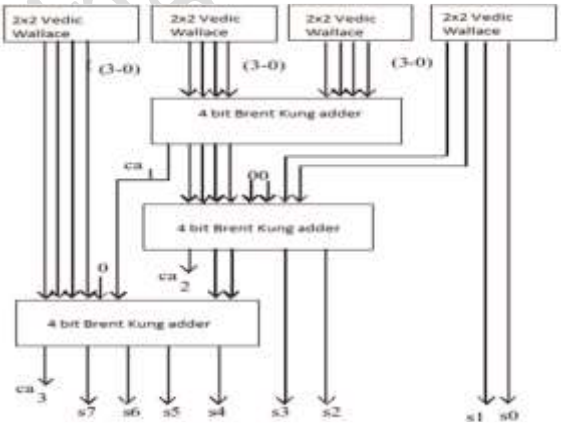


Figure 5. Schematic Diagram of 4x4 bit Multiplier

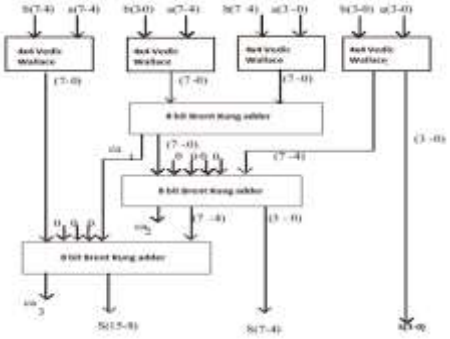


Figure 6 Schematic Diagram of 8x8 bit Vedic Wallace based multiplier diagram of N*N bit vedic Wallace based mulitiplier

IV. PROPOSED METHOD

ARITHMETIC LOGIC UNIT

In this paper the Full-Swing GDI technique is used to realize the circuits required to design the ALU as follows:

A. 2x1 Multiplexer

A multiplexer is a digital switch chooses the output from several inputs based on a select signal [4], shown in Fig. 2 a 2x1 multiplexer consists of 6 transistors.

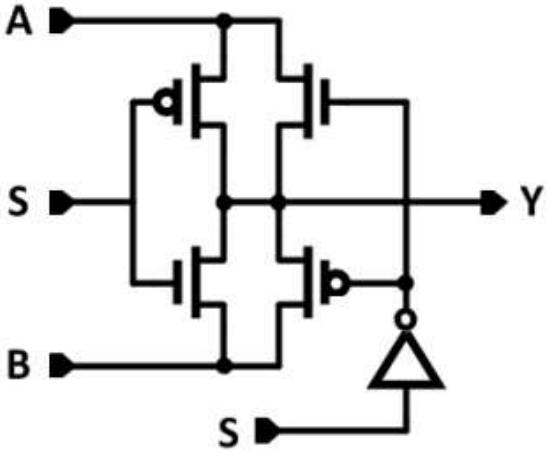


Fig. 7. Full-Swing GDI 2x1 Multiplexer

B. 4x1 Multiplexer

Using the previously discussed 2x1 multiplexer a 4x1 multiplexer realized as shown in Fig. 3 consists only of 16 transistors.

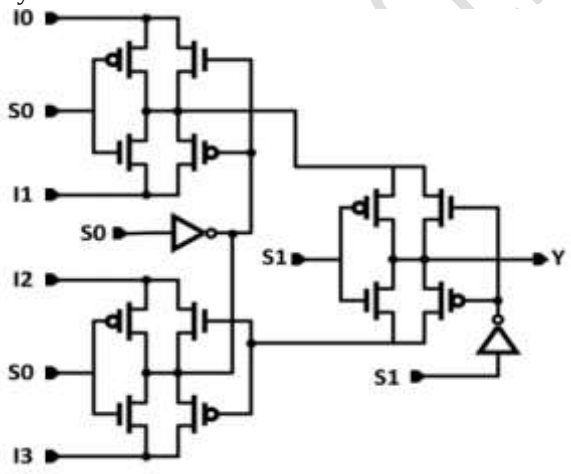


Fig. 8. Full-Swing GDI 4x1 Multiplexer

C. Full Adder

A full adder is a combinational circuit that performs the arithmetic sum of three input bits. It consists of three inputs and two outputs. The adder cell used in this design realized using full-swing AND, OR, and XOR gates. This design was chosen among 3 designs to maintain low power operation, it has the lowest delay among the three designs [5], and with some modifications it performs the logic operations as well, these modifications will save large area of the ALU design.

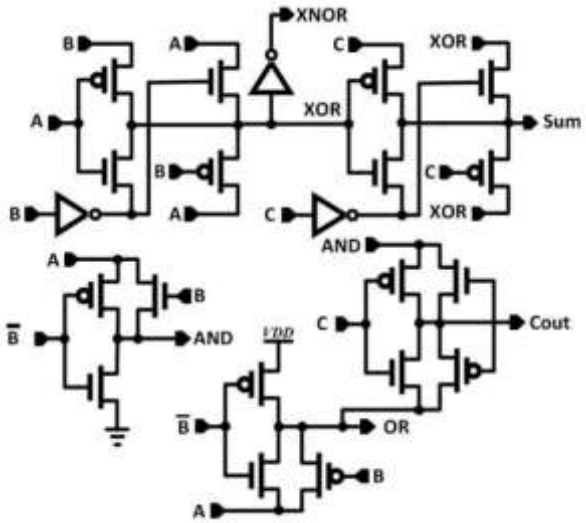


Fig. 9. Full-Swing Full Adder cell

D. Design of Arithmetic Logic Unit

An ALU is a key component in the Central Processing Unit (CPU) of any computer; even the simplest microprocessors contain one. It performs arithmetic operations such as addition, subtraction, increment, decrement and logic operations such as AND, OR, XOR and XNOR [6]. The proposed design of the 4-Bit ALU consists of 4 stages, each stage is an 1-Bit ALU realized using the previously discussed circuits as follows: Each 1-Bit ALU stage consists of two 2x1 multiplexers, two 4x1 multiplexers and one full adder cell, this design requires 48 transistors as depicted in Fig. 5. Any desired operation can be performed based on the selection line S0, S1, S2 code; Table II summarizes the truth table of the proposed ALU.

TABLE II. TRUTH TABLE OF THE PROPOSED 4-BIT ALU.

S2	S1	S0	Operations
0	0	0	DECREMENT
0	0	1	ADDITION
0	1	0	SUBSTRACTION
0	1	1	INCREMENT
1	0	0	AND
1	0	1	XOR
1	1	0	XNOR
1	1	1	OR

The 4x1 multiplexer at the input responsible for the B input based on the values of S0 and S1 selection lines it selects from logic 1, B, ⍜ And logic 0 to perform the Decrement, Addition, Subtraction and the Increment operations respectively, S2 chooses between the arithmetic and the logic operations.

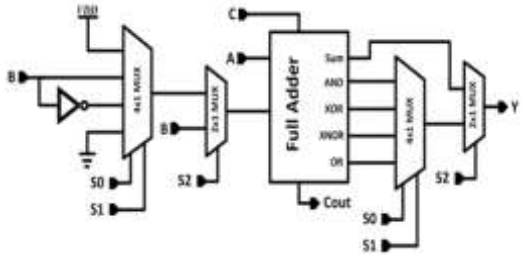
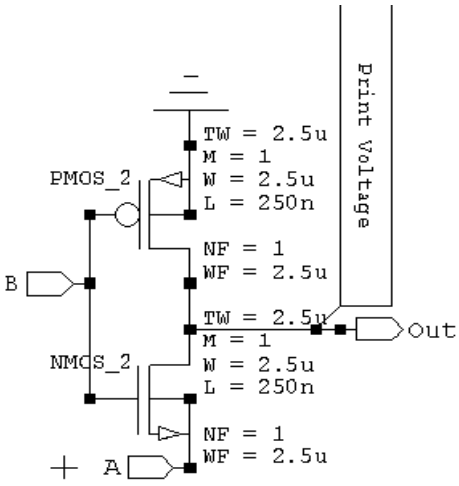


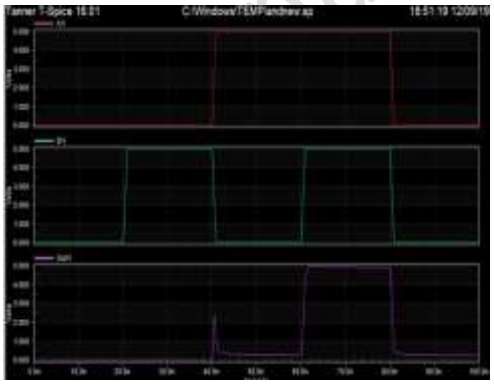
Fig. 10. Schematic of 1-Bit ALU Stage

To realize the 4-Bit ALU four stages were used as shown in Fig. 6. While the carry input of ALU0 connected to selection line S1 to obtain logic 1 which needed for subtraction and increment operations, however the other values don't affect the results of the logic operations

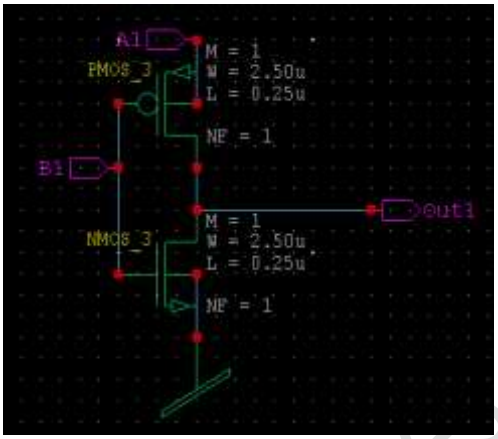
V. SIMULATION RESULTS

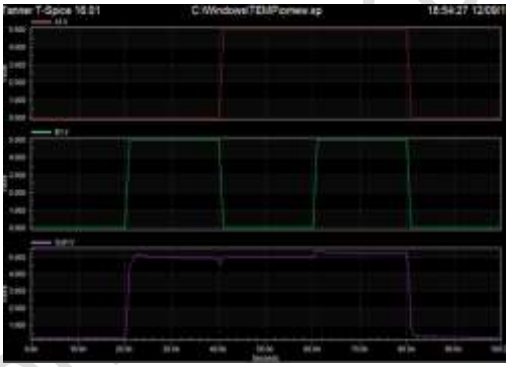
Proposed AND



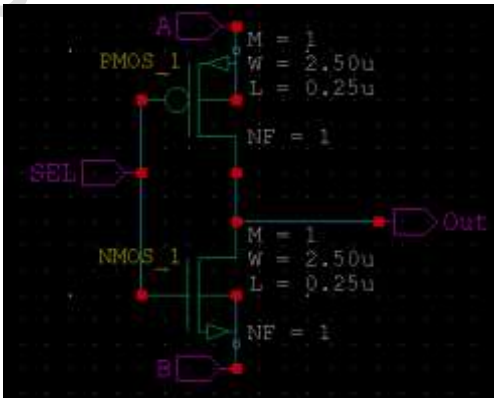


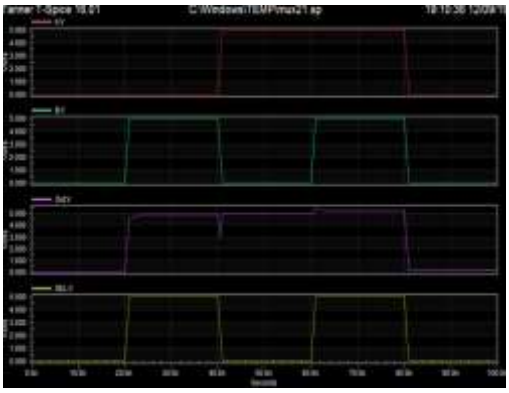
Proposed OR



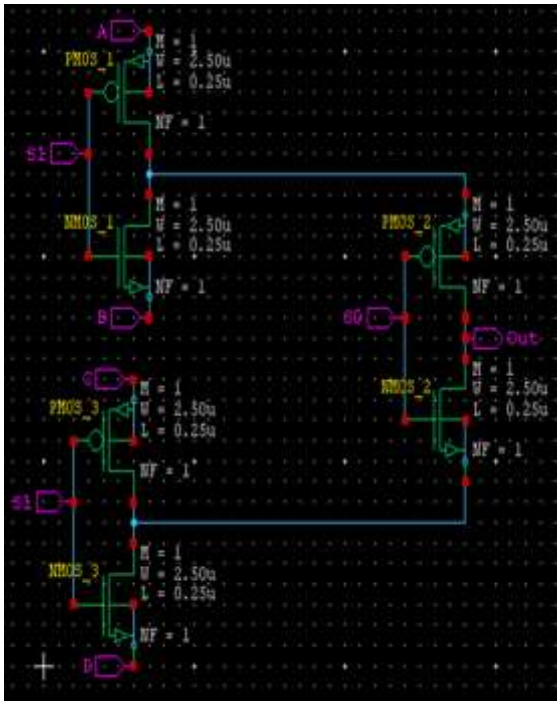


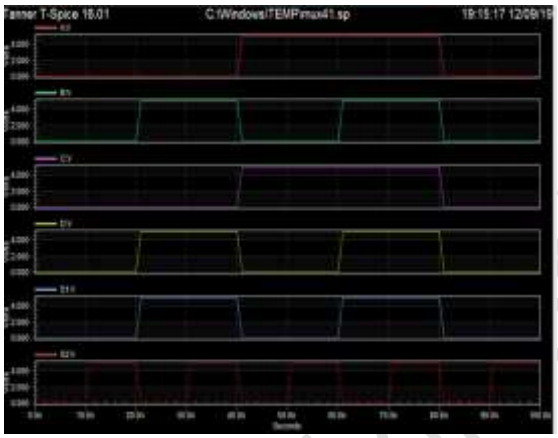
Proposed 2to1 mux



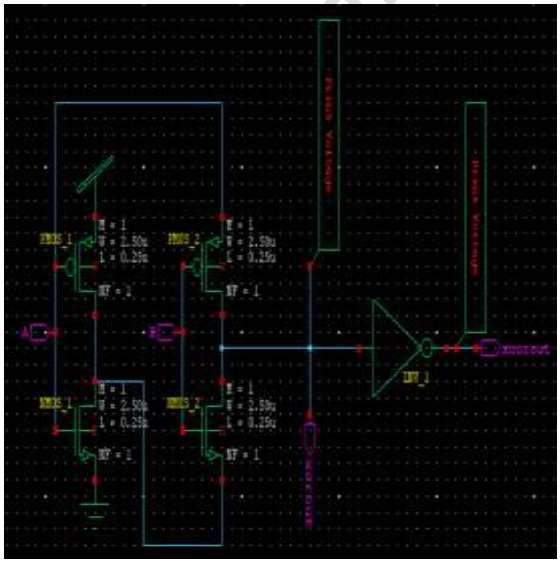


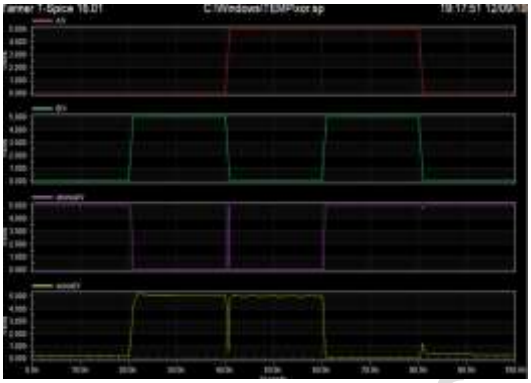
Proposed 4 TO 1 MUX



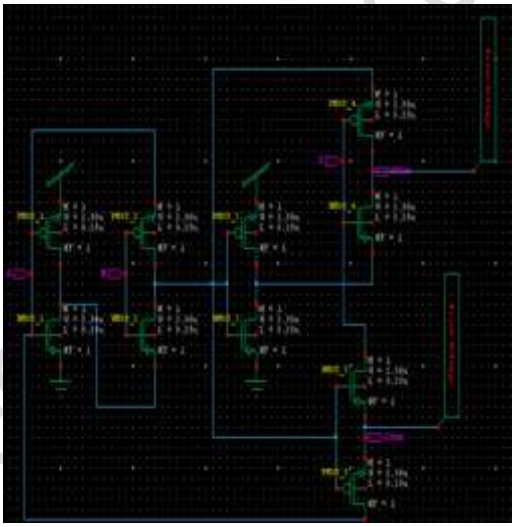


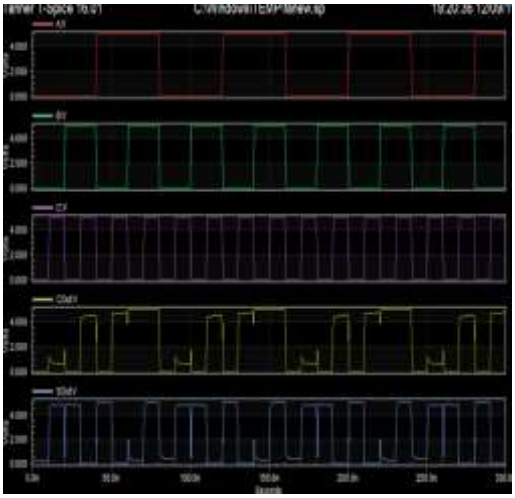
Proposed xor xnor



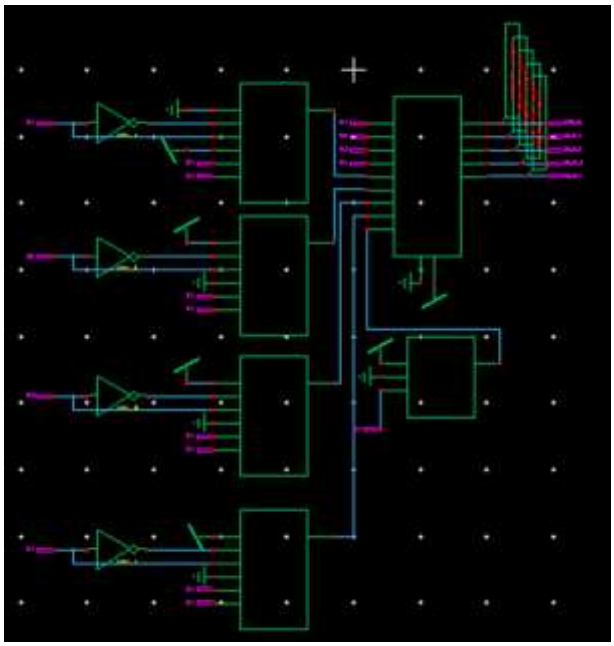


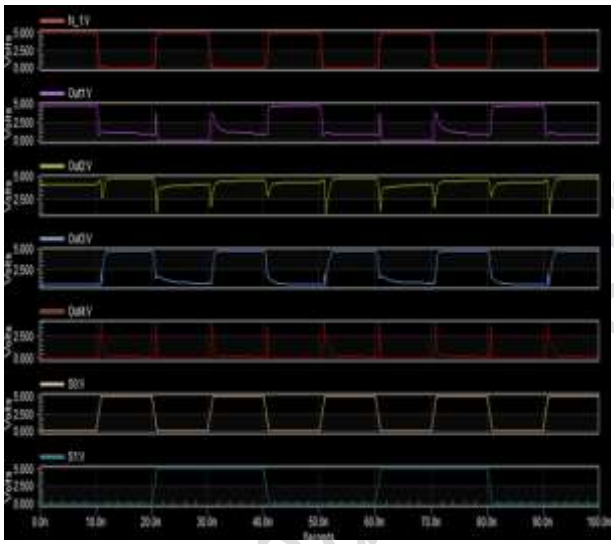
Proposed full adder





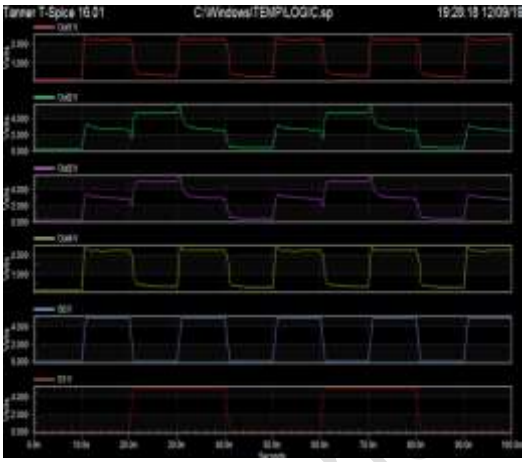
Proposed ARTHEMATIC UNIT

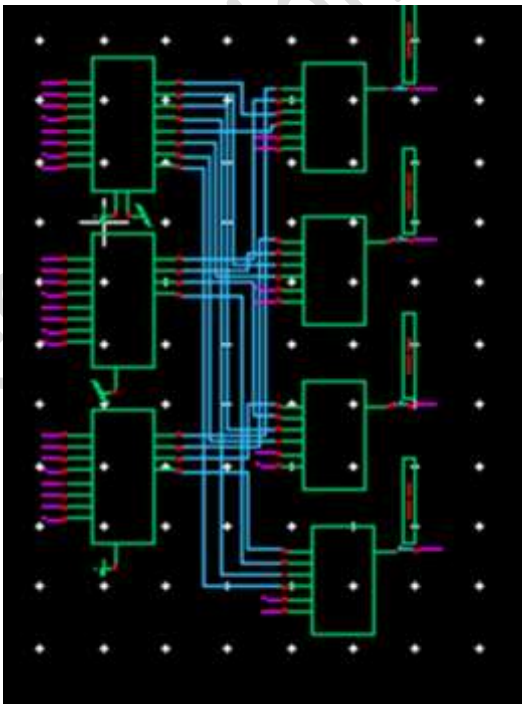




```
Device and node counts:
                MOSFETs -       74
   MOSFET geometries -          3
     Voltage sources -         11
          Subcircuits -        14
   Model Definitions -          2
     Computed Models -          2
   Independent nodes -        107
      Boundary nodes -         12
          Total nodes -        119
```

Proposed LOGIC UNIT





```
Device and node counts:
                MOSFETs -       64
   MOSFET geometries -          3
     Voltage sources -         11
          Subcircuits -        15
   Model Definitions -          2
     Computed Models -          2
   Independent nodes -        76
      Boundary nodes -         12
          Total nodes -        88
```

PARAMRTER	180nm	90nm	45nm
power	0.201 mw(201uw)	11.264 uw	6.33uw
delay	6ns	4.894ns	3.817ns
current	0.234ma	0.987ma	0.172ma
voltage	1.17v	1.14v	0.4v
temperature	31°c	27°c	7°c

VI. CONCLUSION

This work presents a 4-Bit ALU designed in TSMC 65 nm CMOS process using the Full-Swing GDI technique and simulated using the Cadence Virtuoso simulator. The simulation results showed a benefit in terms of energy consumption and transistor count of the new ALU design while maintaining Full-Swing Operation. The architecture proposed is made up of 138 transistors and operates under 1V supply voltage.

FUTURE SCOPE

As a future scope of the work, Power of the circuit can be further reduced by sizing the transistor keeping logical effort in mind. We can consider the GDI technique in sequential logic designs and mixed circuits. Also, we can compound this technique with other methods used in low-power circuits such as using sleep transistors, dual threshold CMOS, dynamic threshold CMOS. The proposed circuit is implemented in regular p-well CMOS processes, which casts a limitation on a GDI cell library. Still, even in limited-library-based GDI circuits, significant improvements of performance are observed. Implementations of GDI circuits in SOI or twin-well CMOS processes are expected to supply more power-delay efficient design, due to the use of a complete cell library with reduced transistor count. The wasted area problem of large Wallace tree multipliers can be solved using a new method of tree construction

REFERENCES

[1]. GarimaRawat, KhyatiRathore, SiddharthGoyal,Shefali Kala and Poornima Mittal, (2015). “Design andAnalysis of ALU: Vedic Mathematics”. IEEE Int. Conf. on Computing, Communication and Automation (ICCCA2015), pp. 1372-1376.

[2]. Rahul Nimje and ShardaMungale, (2014). “Design of arithmetic unit for high-speed performance using Vedic mathematics”. International Journal of Engineering Research and Applications, pp. 26-31.

[3]. Poornima M, Shivaraj Kumar Patil, Shivukumar, Shridhar K P and Sanjay H, (2013). “Implementation of multiplier using Vedic algorithm”.International Journal of Innovative Technology and Exploring Engineering, Vol. 2, No. 6.

[4]. M. Sowmiya, R. Nirmal Kumar, S.Valarmathy and S. Karthick, (2013). “Design of Efficient Vedic Multiplier by the analysis of Adders”. International Journal of Emerging Technology and Advanced Engineering, Vol. 3, No.1.

[5]. PushpalataVerma and K. K. Mehta, (2012). “Implementation of an Efficient Multiplier based on Vedic Mathematics Using EDA Tool”. International Journal of Engineering and Advance Technology, Vol.1, No. 5.

[6]. Abhishek Gupta,UtsavMalviya and VinodKapse, (2012). “A novel approach to design high-speed arithmetic logic unit based on ancient Vedic multiplication technique”. International Journal of Modern Engineering Research, Vol. 2, No. 4.

[7]. SuchitaKamble and N. N. Mhala, (2012). “VHDL implementation of 8-bit ALU”.IOSR Journal of Electronics and Communication Engineering, Vol. 1, No. 1.

[8]. PushpalataVerma, (2012). “Design of 4x4 bit Vedic Multiplier using EDA Tool”. International Journal of Computer Applications, Vol. 48, No. 20.

[9]. AniruddhaKanhe, Shishir Kumar Das and Ankit Kumar Singh, (2012). “Design and Implementation of Low Power Multiplier Using Vedic Multiplication Technique”.International Journal of Computer Science and Communication (IJCSC), Vol. 3, No. 1, pp. 131-132.

[10]. UmeshAkare, T.V. More and R.S. Lonkar, (2012).“Performance Evaluation and Synthesis of Vedic Multiplier ”. National Conference on Innovative Paradigms in Engineering & Technology (NCIPET-2012), Proceedings

published by International Journal of Computer Applications (IJCA), pp. 20-23.

[11]. Anvesh Kumar and Ashish Raman, (2010). "Low Power ALU Design by Ancient Mathematics". IEEE, 978-1- 4244-5586-7/10

[12]. Parth Mehta and DhanashriGawali, (2009). "Conventional versus Vedic mathematics method for hardware implementation of a multiplier".International Conference on Advances in Computing, Control, and Telecommunication Technologies, pp. 640-642.