International Journal of Gender, Science and Technology

..ISSN: 2040-0748

UGC Care Group I Journal

Vol-12 Issue-01 June 2023

8 BIT ARITHMETIC LOGIC UNIT DESIGN USING MODIFIED GATE DIFFUSION INPUT (M-GDI) TECHNIQUE

1st Dr.Y.Murali mohan babu, 2nd G.Munirathnam

[1]Professor, [2]Research Scholar,
[1]ECE Department
[1]CREC,Tirupati,Affiliated to JNTUA, Ananthapuramu, A.P,India
[2]JNTUA, Ananthapuramu, A.P, India
[1]kisnamohanece@gmail.com,[2]muni.446446@gmail.com,

Abstract: A proposal is made to develop an 8-bit Arithmetic Logic Unit (ALU) with Gate Diffusion Input (GDI) technology. The ALU is designed with minimal power consumption and transistor count thanks to the application of the GDI approach. They have the effect of lowering power consumption and chip size, two of the key factors in digital VLSI design. The entire adder in this design uses 2T XOR. Additionally, a unique decoder circuit has been incorporated into the design. An 8-bit ALU with eight distinct operations is ultimately constructed after a significant number of research articles are examined and compared between different logic families.

I. INTRODUCTION:

The Arithmetic Logic Unit (ALU), which is the primary component of a Central Processing Unit (CPU), is capable of carrying out a wide range of logical and arithmetic operations, including multiplexing, XOR, AND, subtraction, addition, and division. An ALU is required for any processing device, whether it be a VLSI chip or simpler circuits tailored to a particular purpose. Enhancing the ALU's design leads to a significant increase in the processor's overall performance and power consumption. Creating larger and more complex circuits with compact implementation and low power dissipation is a fundamental goal, and ALUs are no different. For the past few decades, a lot of work has been put into making typical CMOS-based circuits smaller and more energy-efficient. Consequently, a variety of strategies have been devised and put into practise to improve the performance of CMOS-based circuits, including Transmission Gate Logic, Domino Logic, Double Pass Transistor Logic, and pass Transformer logic. The Gate Diffusion Input (GDI) technique is a novel approach to circuit design that significantly lowers power requirements. Additionally, GDI leads to fewer transistors, which reduces chip area and gives the designs an advantage over traditional techniques. A basic GDI cell consists of just two transistors: an NMOS and a PMOS. G, N, P, and D are its four terminals; the first three serve as input terminals. However, the swing problem with the GDI approach is a serious drawback. It occurs as a result of the weak logic 0 produced by the PMOS and the weak logic 1 produced by the NMOS. This issue is resolved by making some adjustments to the current GDI method. It turns out that the modified GDI method is more effective than CMOS and PTL. Furthermore, an XOR gate can be created with just three transistors by utilising the m-GDI approach. As can be seen in, the several traditional designs of the XOR gate require more than three transistors. The quantity of transistors needed to create a full adder in CMOS. It is possible to create a comprehensive adder with just 8 transistors by utilising the 3T XOR. In the last few decades, the field of electronics has advanced in ways never seen before—in an explosive, exhilarating way. The way electronics concepts are implemented and designed has undergone numerous significant modifications. The creation of these solutions has been made feasible by effective modelling and digital system design methodologies. A hardware description language, or HDL, is a language used in electronics that is used to formally describe standard text-based expressions of the temporal and spatial organisation and behaviour of electronic systems. It explains how an electronic circuit or system behaves, providing insight into the physical circuit or system. The main characteristic of an HDL is its ability to describe hardware function without regard to implementation. The functionalities, interface, and general architecture of the digital circuit to be created are all described in abstract form in the specifications, which are written first in the VLSI design pipeline. Regarding its behaviour, the circuit descriptions are written in Verilog or VHDL utilising HDL. The behavioural level design needs to be clarified in terms of recognised and acknowledged functional blocks. Circuit simulators are used to verify the design's functionality after it is finished. Using a logic synthesiser, the RTL description is transformed into a gate-level net list following a functionality test. An explanation of the circuit in terms of gates and the connections between them can be found in a gate-level net list. The process of converting an abstract representation of intended circuit

behaviour—typically register transfer level (RTL)—into a logic gate-based design implementation is called synthesis. The gate level net list is checked by a logic synthesis tool to make sure timing, area, and power requirements are met. If, after multiple annotations, the desired result can be determined, FPGA or ASIC is used for the final implementation. Generally, the biggest impact on a simulator's performance comes from varying the modelling of HDL code. The goal of event-driven simulation is to remove pointless gate simulations without adding an excessive amount of extra testing. The foundation of this simulation method is the idea of an event. An event-driven simulator's primary job is to identify events and plan gate simulations in reaction to them. No gates will be simulated if there are no events, which suggest that there are no net changes. Certain hardware components can be modelled in a variety of ways, some of which will perform better in simulation than others because of variations in the number of events they produce. The greatest coding flexibility is available to designers when creating procedural RTL. This provides numerous options to develop code that simulates inefficiently and to tweak the code for simulation performance. This study focuses on how an inefficient coding style can negatively affect synthesis and simulation, leading to slow circuits, and how to balance performance difficulties with optimised coding style while maintaining hardware quality. Arithmetic operations are widely used in the majority of VLSI applications, including microprocessors, digital signal processing, and image and video processing. The most often utilised operations are addition, subtraction, multiplication, and multiply and accumulate (MAC). The basis for each of these modules is a 1-bit full-adder cell. Therefore, improving its performance is essential to improving the performance of the module as a whole. A complete adder cell's design requirements are typically multifaceted. Of course, a major factor that significantly influences the design complexity of larger circuits is the number of transistors. Topology selection, power dissipation, and speed are crucial factors in this submicron CMOS technology area for high-speed and low-power applications. The application of the Gated Diffusion Input (GDI) technology can solve these problems. The literature reports on a number of complete adder design optimisation strategies. One lowest power design method that provides better logic swing and less static power dissipation is Gate Diffusion Input (GDI). This method allows for the implementation of several logic functions with fewer transistor counts. With this approach, you can create quick, low-power circuits with fewer transistors than you would with TG and CMOS.

Many logic design strategies have been proposed to lower the power consumption, including modified Domino logic (PTL), cascade voltage switch logic (CVSL), pseudo-n MOS, dynamic CMOS, clocked CMOS logic (C2 MOS), CMOS complementary logic, and pass transistor logic (PTL). Over the past three decades, static CMOS logic has been the most widely used design technique. However, numerous attempts have been made to suggest an improved alternative that would result in lower power dissipation, a smaller footprint, and higher stated performance. Low logic level swing can be solved by circuits designed with transmission gates (TGs), which use both PMOS and NMOS. However, this implementation requires a true and complemented control signal, which takes up more space than pass transistor logic. Pseudo-NMOS is quick and easy to use, but it uses more power and has smaller noise margins. Pass-transistor logic works well with particular circuit types (MUX/adders). However, it was discovered that PTL versions of logic gates, such NANDs and NORs, were slower and used more power than CMOS equivalents. This was mostly because single-channel pass transistors have smaller output swings as a result of their threshold drop. This paper's primary contribution is the circuit-level implementation of five distinct complete adder topologies and modified primitive cell designs, all based on the GDI approach. Modified GDI primitive cells are built and their notable differences between CMOS and traditional GDI are contrasted. Despite the low power, low transistor count, and rapid speed of the GDI approach, the fabrication process presents the most obstacles. To realise a chip using the GDI technique, a silicon on insulator (SOI) or twin-well CMOS process is needed, which raises the cost and complexity of manufacture.

As digital technology has advanced, it is now crucial to design circuits that use the least amount of power, run quickly, and take up the least amount of space on a chip. Scaling should be incorporated into circuit designs in order to improve performance. The design characteristics of a digital circuit are significantly impacted by the quantity of transistors utilised to implement logic gates. Over the past few decades, VLSI technology—which entails placing more than a million transistors on a chip—has been the primary focus of designers' attention. diverse logic design techniques, such as complementary MOS (CMOS), pseudo-NMOS, transmission gate, dynamic CMOS logic, and pass transistor logic, are used by designers to enhance diverse circuits. Each of these methods has a unique design mechanism, along with some benefits and drawbacks. CMOS technology is used in the fabrication of over 90% of all semiconductor devices. NMOS and PMOS, which complement one another architecturally and functionally, are combined to form CMOS. On the other hand, pull-up transistor static power consumption, decreased output voltage swing and gain, and high speed and low transistor count of pseudo-NMOS logic make the gates more vulnerable. An NMOS and a PMOS coupled in parallel form

a bidirectional switch known as a transmission gate. It circumvents the issue of a smaller noise margin and higher dissipation of static power, but it does necessitate the availability of the control and its complement. This technology's drawbacks include slower speed and the inability to connect more than three transmission gates in a cascade because of a charge sharing issue. By getting rid of unnecessary transistors, pass transistor logic lowers the number of transistors needed to create distinct logic gates. Instead of using switches that are directly attached to the voltage source, PTL designers use transistors as switches to transfer logic levels between circuit nodes. The primary drawback of PTL is that as stages progress, the voltage differential between high and low logic levels decreases. Arithmetic operations (such as addition, subtraction, multiplication, division, etc.) are widely employed in a variety of applications in digital technology. A subtractor is a key component in digital signal processing and image processing in microprocessor operations. The fundamental unit needed to perform the subtraction of two binary digits is a one-bit subtractor. There is one AND gate, one XOR gate, and one NOT gate in it. A full subtractor is a combinational circuit that generates two outputs, difference (D) and borrow out (B), by performing subtraction using three bits: the borrow from the previous stage, the minuend bit, and the subtrahend bit. More logic gates are needed to construct a full subtractor, which increases the number of transistors and thus increases the area of a device. As a result, there is an increase in delay, power consumption, and needed space. By employing the modified gate diffusion input technique, we may lower the number of transistors (M-GDI). This paper's primary contribution is the presentation of an ideal solution for the construction of a full subtractor utilising eight or nine transistors, utilising the three-transistor-based XOR gate design technique and the M-GDI technique. Using a three-transistor-based XOR gate design technique and a modified GDI technique, 9T and 8T full subtractor designs were created.

The proliferation of integrated circuit (IC) devices is a direct result of the flourishing field of computer science. Arithmetic operations are widely used in the majority of Very Large Scale IC (VLSI) applications, including digital-signal processing and microprocessors. Furthermore, of these frequently performed operations, multiplication and subtraction are used the most. The foundation of these operation modules is the 1-bit full adder. As such, improving its performance is essential to improving the performance of the modules as a whole. As portable integrated circuit (IC) devices become more commonplace—such as MP3 players, cell phones, PDAs, and so forth—IC engineers are under pressure to enhance the functionality of current operation modules in a few key areas, most notably power depletion and size. IC designers have faced additional challenges due to the slower advancement of battery technology compared to microelectronics technology. These limits include the need for fast speed, high throughput, small silicon area, and low power dissipation. Thus, there is a lot of research being done on the creation of low-power, high-performance adder cells. Approaching this task from a structural perspective is a fruitful strategy to finish it. Future research and development can be made possible by this method of design and analysis, which divides an adder cell into smaller modules. An optimised entire adder cell can then be created by connecting these improved smaller components.

XOR gates are the basic component of complete adders. The performance of the adder can be greatly increased by optimising the XOR gates. A review of the literature shows that several distinct kinds of XOR gates have been developed over time. The eight or six transistors that are often utilised in most designs served as the foundation for the early XOR gate designs.

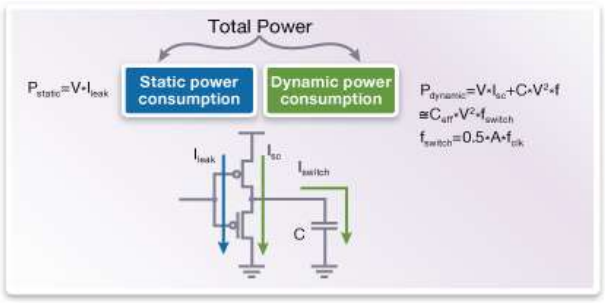

Fig1.Power dissipation in CMOS

1. Dynamic power consumption: The quantity of power used when a device is in use. More precisely, when logic circuits transition states, this is the total power used to charge and discharge capacitances in transistor architectures. Since CMOS logic circuits only use power when switching, a device's overall power consumption can be decreased by lowering the number of switching events and the ON-state voltage.

2. Static power consumption: The total of leakage current and operating voltage determines this. A little amount of current leaks through gates even when transistors are turned off, dissipating as heat. Though scaling presented difficulties, CMOS chip layouts offer lower leakage currents than much earlier bipolar devices.

II. BACKGROUND:

The Gate Diffusion Input (GDI) technique is a novel approach to circuit design that significantly lowers power requirements. Additionally, GDI leads to fewer transistors, which reduces chip space and gives the designs an

advantage over traditional techniques [9]. A basic GDI cell consists of just two transistors: an NMOS and a PMOS. The first three of its four terminals—G, N, P, and D—serve as input terminals [9]. However, the swing problem with the GDI approach is a significant drawback [10]. It occurs as a result of the weak logic 0 produced by the PMOS and the weak logic 1 produced by the NMOS. By making certain adjustments to the current GDI method, this issue is resolved. It is found that the improved GDI approach outperforms both PTL and CMOS. Furthermore, an XOR gate with just three transistors can be created utilising the m-GDI approach. Eight transistors are all that are needed to construct a complete adder with the aid of the 3T XOR.

Additionally, this approach makes use of a unique 1-to-8 demultiplexer architecture that makes use of the GDI methodology, allowing two select lines to be used to choose just the desired module in the ALU. The principal components of the suggested m-GDI ALU are

Arithmetic Unit: This unit carries out addition and subtraction as well as other fundamental arithmetic operations. Logical Unit: This unit executes logical procedures. OR, AND, XOR, NOT, and NAND gates are presented in this work.

The CMOS implementation of the inverter circuit is comparable to the fundamental GDI primitive cell. However, it is far more powerful than that. The way that the circuit functions depends on the inputs that are provided to G, N, and P. The common gate connecting the PMOS and NMOS is G. The source terminals of the NMOS and PMOS are N and P, respectively. Each of these serves as an input terminal. The common drain D is where the output is gathered. Fig. 2 depicts the schematic of a simple GDI primitive cell.

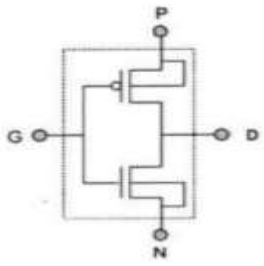

Fig. 2: A Basic GDI primitive cell

This method uses the GDI Technique to design the ALU. Below is a block diagram of an ALU using the GDI approach.

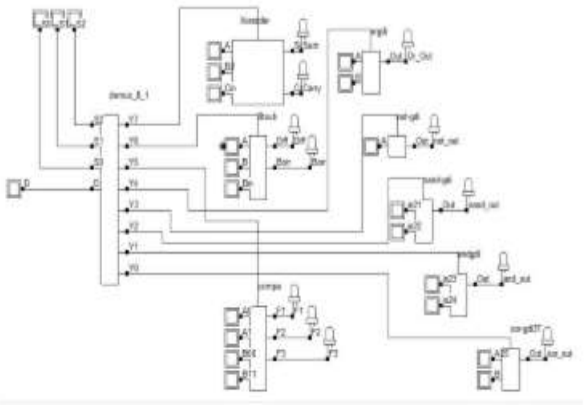

Fig.3 ALU design using GDI

Three transistors, one fewer than those required to build the XOR gate using GDI, are utilised in the ALU's implementation. Additionally, full subtractor and full adder modules use it. Fig. 4 below displays the 3T XOR design.

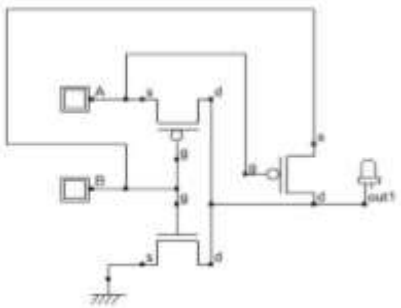

Fig.4: 3T XOR gate

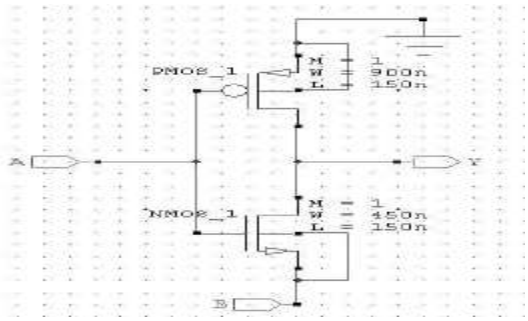

Fig.5: AND gate

The suggested ALU uses 3T XOR gates for all of its XOR gates. As a result, both the overall power consumption and the total number of transistors in the design have significantly decreased.

To implement the XOR, one input (A) is fed to the pMOS's source and gate terminals, and the other input (B) is fed to the nMOS's and the pMOS's shared gate input terminal in addition to the pMOS's source terminal. The output terminal

of the transistors is their common drain terminal. The resulting 3T XOR only has a single transistor's worth of delay. It does, however, come with a warning that the output logic level could occasionally become distorted. Nonetheless, Dan Wang et al. have demonstrated in [14] that it is always possible to guarantee the required logic level at output by modifying the W/L ratio.

Xor gates and multiplexers are used in the design of the full adder and full sub tractor cells. Below is a circuit diagram of a mux that uses the GDI approach.

The entire adder and full subtractor that were created with the aid of this 3T XOR are displayed in Figs. 4 and 5, respectively, below.

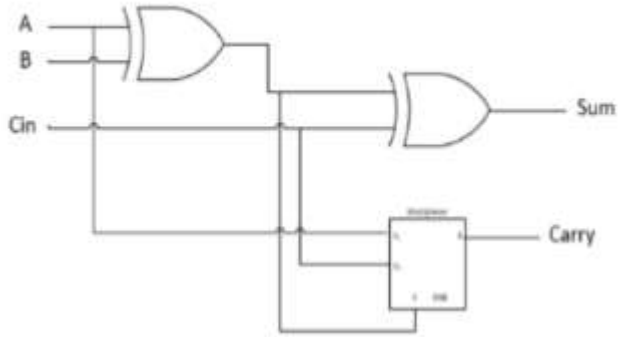

Fig.6: Full adder design

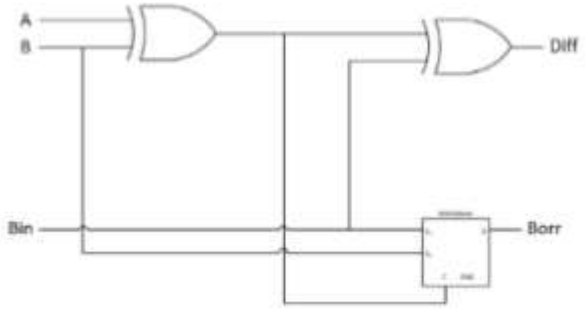

Fig.7: Full subtractor design

III. PROPOSED SYSTEM:

The XOR utilised in the ALU in the current technique is designed using 3T. The complete adder and full subtractor circuits, which are also utilised in the ALU to carry out arithmetic operations, employ the XOR gate. Therefore, in our suggested method, the xor gate is developed utilising 2T to improve the efficiency in terms of power and area. Certain features cannot be incorporated inside the allotted chip area if designing a basic circuit requires additional space since we are unable to fit extra logic within the chip. We must design the fundamental circuits of the device to be as area-efficient as possible in order to add more functionality.

Another crucial factor to take into account while building any digital circuit is power consumption. There has been a recent upsurge in interest in low-power electronics and design methodologies. The Arithmetic Logic Unit (ALU), which is the primary component of a Central Processing Unit (CPU), is capable of carrying out a wide range of logical and arithmetic operations, including multiplexing, XORing, ANDing, addition, subtraction, multiplication, and division, as well as inversion and other Boolean operations. An ALU is required for any processing device, whether it be a VLSI chip or simpler circuits tailored to a particular purpose. Enhancing the ALU's design leads to a significant increase in the processor's overall performance and power consumption. Therefore, we recommend replacing the 3T xor gate with a 2T one and the decoder with a demultiplexer in order to reduce both power usage and the number of transistors. This is the suggested 2Txor gate.

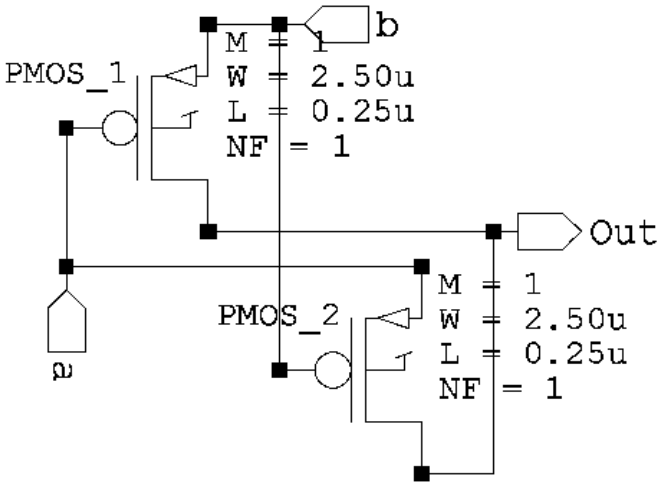

Fig.8: Circuit diagram of 2T XOR gate.

We created the complete adder and entire subtractor circuits that are utilised in the ALU to lower the transistor count by utilising the suggested 2T XOR gate.

Below are the schematics for the suggested full adder and full subtractor circuits.

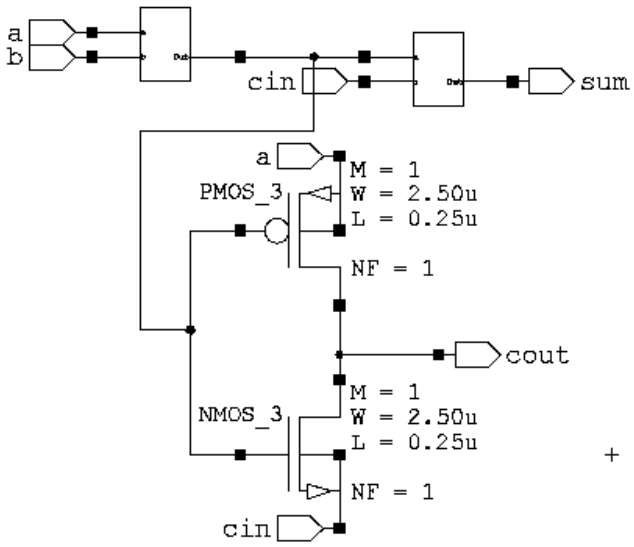

Fig.9: proposed full adder circuit using 2T XOR

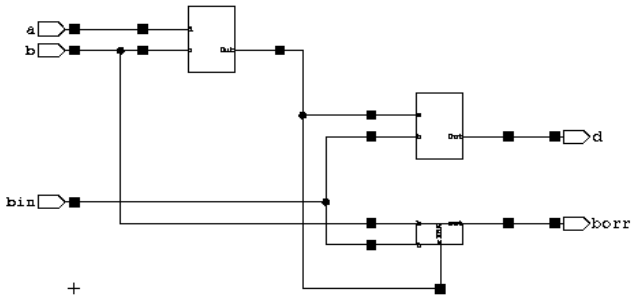

Fig.10: proposed full Subtractor circuit using 2T XOR

Three inputs (A, B, and C) and eight outputs (DO to D7) make up a 3 to 8 decoder. One of the eight outputs is chosen based on the three inputs. The circuit diagram of 3: 8 decoder is shown below.

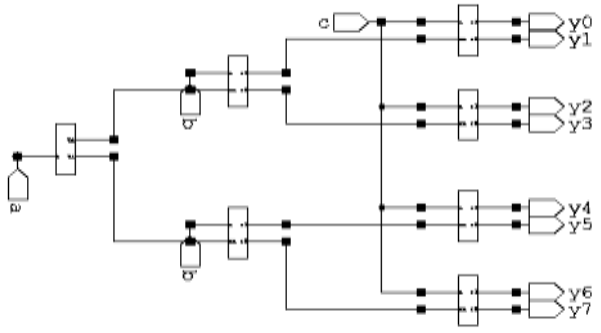

Fig.11: Circuit diagram of 3 to 8 decoder.

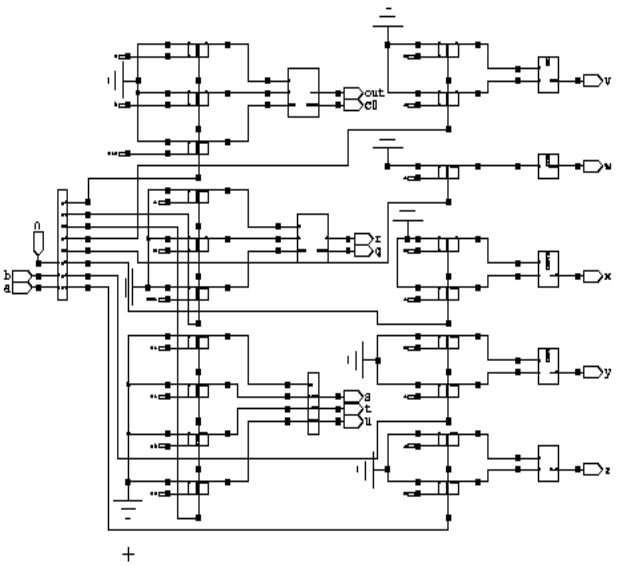

Fig.12: Circuit diagram of Proposed ALU.

IV. CONCLUSION:

The suggested ALU performs better than the others in terms of power and critical path delay in addition to having the fewest transistors overall. It is evident from this that the suggested model outperforms the traditional ones in terms of efficiency. Tanner EDA simulates the designs. According on simulation results, the suggested ALU uses less space and electricity than the current ALU.

V. REFERENCES:

[1] L. Bisdounis, D. Gouvetas and O. Koufopavlou, “A comparative study of CMOS circuit design styles for low power high-speed VLSI circuits” Int. J. of Electronics, Vol.84, No.6, pp 599- 613,1998.

[2] N. Weste and K. Eshraghian, Principles of CMOS digital design. Reading, MA: Addison-Wesley, pp. 304–307.

[3] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, “Low- power CMOS digital design,” IEEE J. Solid-State Circuits, vol. 27, pp. 473–484, Apr. 1992.

[4] A. P. Chandrakasan and R.W. Brodersen, “Minimizing power consumption in digital CMOS circuits,” Proc. IEEE, vol. 83, pp. 498–523, Apr. 1995.

[5] Oh-Hyeong Kwon, “A Boolean Extraction Technique for MultipleLevel Logic Optimization” IEEE 2003.

[6] R.Uma, "4-Bit Fast Adder Design: Topology and Layout with Self- Resetting Logic for Low Power VLSI Circuits"International Journal of Advanced Engineering Sciences and Technology, Vol No. 7, Issue No. 2, 197 205, 2011.

[7] R.Uma and P. Dhabachelvan "Analysis on Impact of Behavioral Modeling in Performance of Synthesis Process The second International Conference on Advances in Computing and Information Technology, July 2012, Published in Advances in Intelligent Systems and Computing book series of Springer, Pg593-602.

[8] R.Uma and P. Dhavachelvan, "Modified Gate Diffusion Input Technique: A New Techniqu for Enhancing Performance in Full Adder Circuits", Procedia Technology 6 (2012) 74 – 81.

[9] ArkadiyMorgenshtein, Alexander Fish and Israel A. Wagner, "Gate-Diffusion Input (GDI): A Power-Efficient Method for Digital Combinatorial Circuits", Ieee Transactions On Very Large Scale Integration (Vlsi) Systems, Vol. 10, No. 5, October 2002.

[10] S. Sweta and Md. Afreen Begum, "Design of High Speed, Area Optimized and Low PowerArithmetic and Logic Unit", Advances in Industrial Engineering and Management,Vol.6, No. 1 (2017), 26-31.

[11] R.Uma, Vidya Vijayan, M. Mohanapriya, Sharon Paul, "Area, Delay and Power Comparison of Adder Topologies", International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.1, February 2012

[12] Shubham Sarkar, Sujan Sarkar, Tarunesh Pati, Indranil Monda, Asif Iqbal, Joy Sengupta, Alojyoti Mistry, "Low Power implementation of Multi-Bit Hybrid Adder using Modified GDI Technique", 2018 International Conference on Electronics, Materials Engineering & Nano-Technology.