ISSN 2277-2685
IJESR/July-September. 2026/ Vol-16/Issue-3/27-31
Boggarapu Sankeerthana *et. al.*, / International Journal of Engineering & Science Research
Full Length Article

Area Optimization of Arithmetic Logic Units Using Majority Logic Gates

Boggarapu Sankeerthana[1], Jyothula Sowjanya[2]
[1]M. Tech Student; GVR&S College of Engineering and Technology, Budampadu-522017, Guntur (Dt), A.P., India.
[2]Associate professor; GVR&S College of Engineering and Technology, Budampadu-522017, Guntur (Dt), A.P., India.
Mail Id: sankeerthanaboggarapu@gmail.com[1] ,jyothulasowjanya@gmail.com[2]
Accepted 22-06-2026

Abstract— *Arithmetic Logic Units (ALUs) are critical to digital systems' performance, power efficiency, and space utilisation. In recent years, alternative logic design strategies such as reversible logic and majority logic have gained popularity for low-power VLSI applications. This paper compares an ALU developed using reversible logic gates to an ALU implemented with majority gates. The reversible-gate-based ALU uses Peres, Fredkin, and CNOT gates to perform arithmetic and logical operations while mini missing information loss. However, the actual implementation includes extra trash outputs and intricate linkages, which increases space and power consumption. In contrast, the majority-gate-based ALU employs compact majority logic structures to efficiently perform arithmetic and logic operations while reducing hardware complexity. Both designs are built and synthesised on an FPGA platform, and their performance is assessed using area and power metrics. The findings show that the majority-gate-based ALU has a much lower logic utilisation and power consumption than the reversible-gate-based ALU. As a result, the current majority-gate ALU is more suited to low-power, area-efficient VLSI applications.*

Keywords*: majority gates, ALU, Verilog HDL.*

I. Introduction

The fast expansion of digital systems has created an ever-increasing need for high-performance, low-power, and space-saving hardware designs. New electronic systems, such as embedded systems, IoT devices, HPC, and SPUs, need to provide enhanced computational capabilities while meeting stringent power and silicon area constraints. The core processing unit of contemporary digital computers, the Arithmetic Logic Unit (ALU) executes logical and mathematical operations. The total efficiency of a processor or digital system is greatly affected by an ALU's performance, power consumption, and space utilisation..As CMOS technology continues to scale, traditional logic design methodologies encounter a number of hurdles, including increasing leakage power, heat dissipation, and reliability concerns.

As device dimensions decrease to the nanometre range, power consumption has emerged as a significant limiting factor in Very Large Scale Integration (VLSI) systems. As a result, there is an increasing need to investigate alternative logic design techniques that may overcome the limits of standard CMOS-based Boolean logic while providing increased energy efficiency and small hardware implementation.In this context, new logic paradigms like reversible logic and majority logic have received a lot of interest in recent years. These logic types offer lower power consumption, more logical efficiency, and compatibility with next-generation computer systems. Reversible logic is especially appealing for low-power and quantum computing applications since it potentially allows for processing with no information loss. In contrast, majority logic has developed as a very efficient design strategy for small and power-optimized digital circuits, advanced CMOS implementations. The ALU, as a key computing module, is a perfect platform for investigating and comparing the efficacy of various diverse logic design methods. A comparison of reversible-logic-based and majority-logic-based ALU designs provides insight into their practical practicality, performance trade-offs, and applicability for low-power VLSI systems

II. Literature Survey

The majority function, the building block of majority logic, takes as inputs the values that are in the majority and returns an outcome depending on those values. A logic '1' is generated by a three-input majority gate when at least two of its inputs are '1'. When paired with inverters, majority gates constitute a functionally complete logic set capable of carrying out any Boolean function. Majority logic has received a lot of interest because of its simplicity, small construction, and

applicability for new nanotechnologies like QCA, nanomagnetic logic, and spintronic devices. In these technologies, the majority gate translates readily to the underlying physical behaviour, making it a useful building piece.In addition to developing technologies, most logic has been effectively translated to CMOS and FPGA platforms. Researchers have shown that majority-gate designs may minimise logic depth, connection complexity, and power consumption when compared to standard logic implementations.

III. Majority-Gate-Based Alu Design

This section describes the design technique for the Arithmetic Logic Unit (ALU) built using majority logic gates. The proposed majority-gate-based ALU takes use of majority logic's functional completeness and compact structure to achieve decreased hardware complexity, lower power consumption, and higher space efficiency. In contrast to reversible logic, which requires rigorous input-output bijection, majority logic emphasises efficient Boolean function realisation using fewer logic pieces and reduced linkages.A majority gate generates an output that is equal to the majority of its input values. The output of a three-input majority gate is logic '1' if at least two of the inputs are '1', and logic '0' otherwise. The majority function may be stated as:

$$M(A, B, C) = AB+BC+CA$$

When paired with inverters, majority gates constitute a functionally complete logic set capable of carrying out all Boolean operations. This characteristic allows for the creation of complicated arithmetic and logic circuits with fewer gate levels and lower connectivity overhead than typical CMOS logic.The suggested majority-gate-based ALU was created with the goal of reducing logic depth, gate count, and switching activity. All arithmetic and logic functions are split into majority-based expressions, which allows for efficient translation to hardware platforms like FPGAs.

The ALU design is modular, with specialised arithmetic and logic submodules. A control unit chooses the required action based on the input selection signals. Majority gates are widely utilised in arithmetic calculation and logic realisation, but inverters are used only when absolutely essential.

The majority-gate-based ALU consists of the following functional blocks:

- Majority-Gate-Based Adder.
- Majority-gate-based subtractor.
- Majority-Gate-Based Multiplier.
- Majority-Gate-Based Logic Unit.
- Operational Selection Unit.

Each block is developed individually using majority logic before being combined to build the overall ALU architecture.

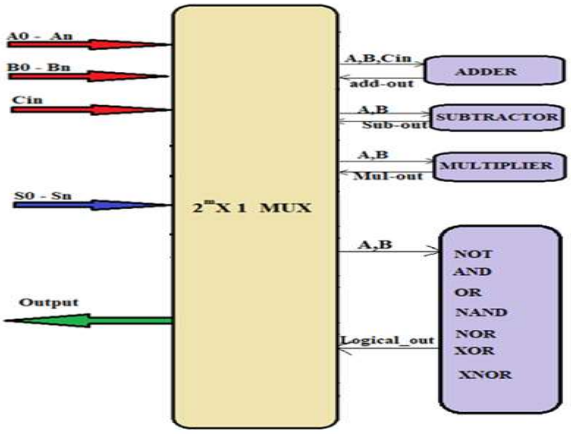

Fig1. ALU Architecture

Table I ALU Function

S3 S2 S1 S0	FUNCTION
0 0 0 1	ADDITION
0 0 1 0	SUBTRACTION
0 0 1 1	MULTIPLICATION
0 1 0 0	NOT GATE
0 1 0 1	OR GATE
0 1 1 0	AND GATE
0 1 1 1	NAND GATE
1 0 0 0	NOR GATE
1 0 0 1	XOR GATE
1 0 1 0	XNOR GATE

Addition is a basic operation in the ALU, and it is easily performed using majority logic. The suggested architecture employs a majority-based complete adder, with the carry output created directly via a majority gate:

$$Carry=M(A, B, Cin)$$

The sum output is achieved by combining majority gates with XOR capabilities implemented in majority-inverter structures. In comparison to traditional complete adders, the majority-gate-based adder uses fewer logic parts and has a shorter propagation time.Multi-bit addition is accomplished by cascading majority-based complete adder blocks, resulting in efficient carry propagation with minimum hardware overhead.

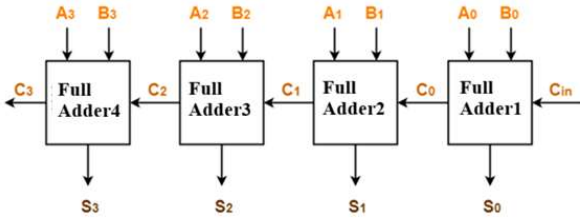

Fig2; bit adder

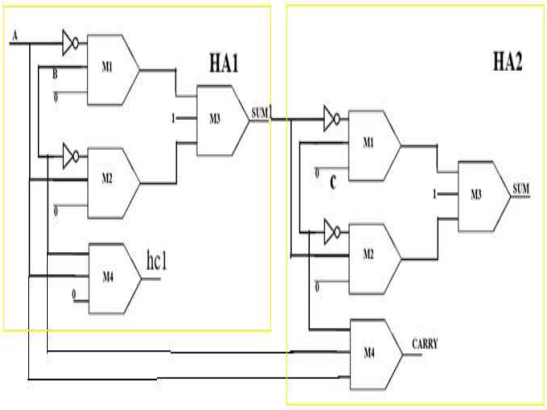

Fig 3. Full adder using QCA majority gates

Subtraction is performed using the two's complement approach, which is comparable to the reversible ALU design. One operand is complemented and added to the other operand, followed by an initial carry-in of logic '1'.In the majority-logic approach, operands are inverted using inverters, and the majority-based adder is reused for subtraction. This reuse of the adder module decreases overall hardware complexity while increasing space efficiency.

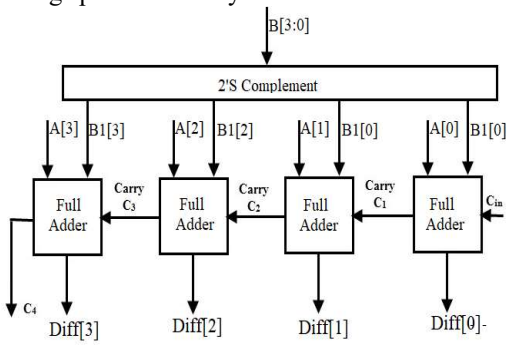

Fig4; bit subtractor

The multiplication action is carried out utilising a multiplier architecture based on majority gates. Partial products are created utilising AND operations implemented in majority gates with one input set to logic '0'. These incomplete products are collected using majority-based adders.The multiplier design prioritises simplicity and decreased logic depth above aggressive performance optimisation. Majority logic's compact carry generation and propagation capabilities allow for efficient partial product accumulation.

The logic unit can perform seven basic logic operations: AND, OR, XOR, XNOR, NAND, NOR, and NOT. Each logic function is implemented using majority gates and inverters.

- The AND operation is accomplished using M(A,B,0).
- The OR operation is accomplished using M(A,B,1).
- XOR and XNOR procedures use majority gates and inverters.
- To get NAND and NOR operations, flip the AND and OR outputs.
- The NOT operation is achieved using a simple inverter.

This majority-based realisation minimises the amount of logic elements needed for the logic unit.

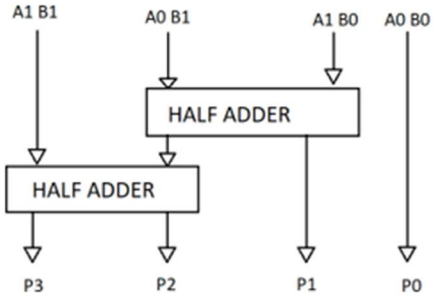

Fig 5; 2x2 Binary Multiplier

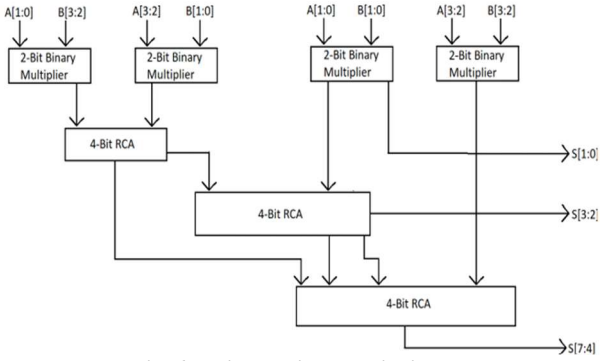

Fig 6 ; Bit Vedic Multiplier.

IV. RESULTS

RTL schematic: The RTL schematic, or register transfer level, is the architecture's blueprint and is used to compare the projected design to the ideal architecture being built. The hdl language converts the architectural description or summary into a working summary using verilog or vhdl. To simplify analysis, the RTL schematic includes internal connection blocks.

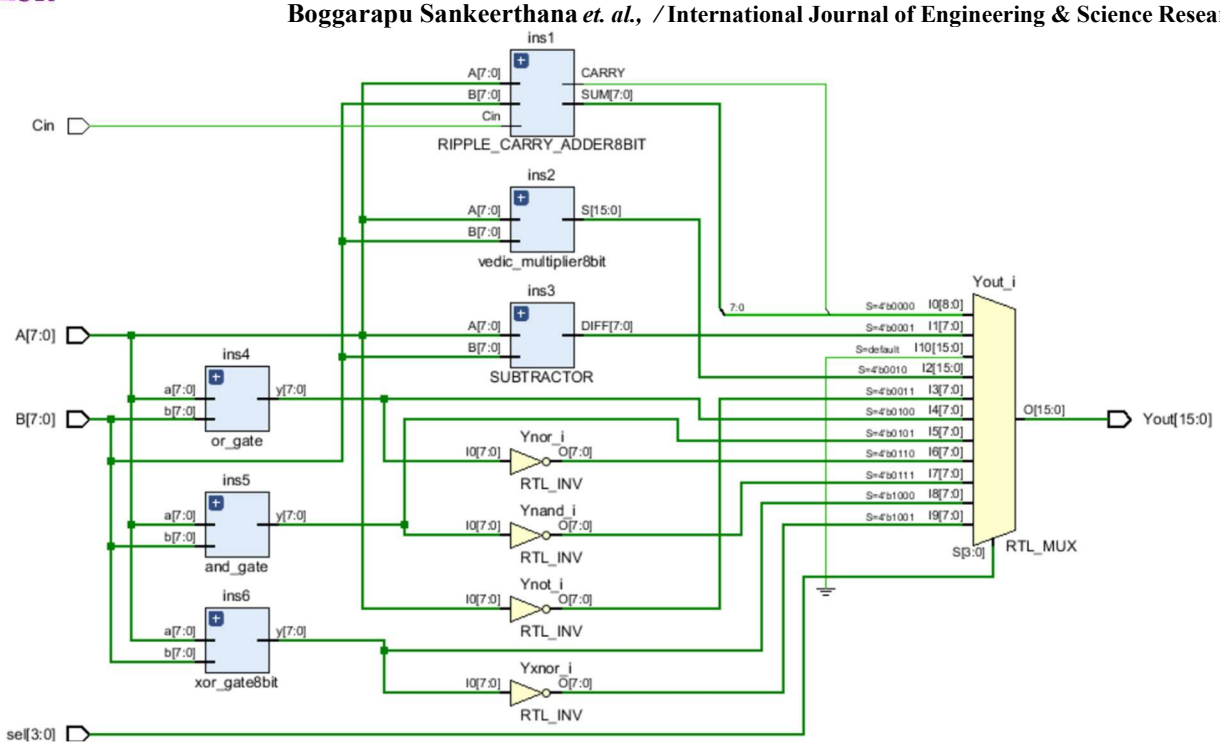

Fig 7. RTL Schematic of ALU using majority gates

SIMULATION:Simulation checks system operation, whereas schematics check connections and blocks. Switching from implantation to simulation on the tool's main screen opens the simulation window, which displays wave forms. It supports many radix number systems.

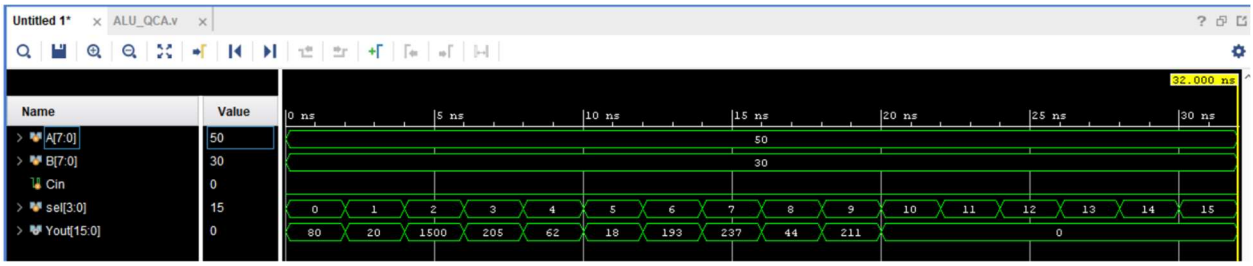

Fig 8. Simulated wave form of ALU using majority gates

Table II Parameter comparison

Parameter	ALU using reversible gates	ALU using majority gates
No of LUTS	175	161
Power(mW)	7.569	7.212

The synthesis findings shown in Table X clearly show that the majority-gate-based ALU outperforms the reversible-logic-based ALU in terms of area and power efficiency. The reversible ALU uses 175 LUTs, but the majority-gate ALU only needs 161 LUTs, resulting in an area decrease of around 8.0%. This improvement is mostly due to the compact implementation of arithmetic and logic operations utilising majority gates, which considerably decreases logic depth and interconnection complexity.

In terms of power usage, the reversible-logic-based ALU uses 7.569 mW, while the majority-logic-based ALU uses 7.212 mW, resulting in a power savings of roughly 4.72%. Although reversible logic is theoretically related with reduced energy dissipation, the existence of garbage outputs and extra control circuitry increases switching activity in actual FPGA implementations. In contrast, a majority-gate-based design reduces duplicated logic and switching transitions, resulting in higher power efficiency.Overall, the findings show that the majority-gate-based ALU is a more efficient option for low-power, area-constrained VLSI applications than reversible-logic-based ALU designs.

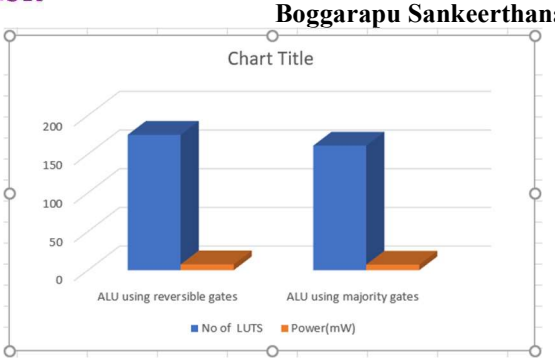

Fig 10. LUT and Power comparison bar graph

Conclusion

This article provided a comprehensive comparison of ALU architectures built using reversible logic gates and majority logic gates. The reversible-logic-based ALU was built using CNOT, Fredkin, and Peres gates to allow arithmetic and logical operations while maintaining information reversibility. Although reversible logic has theoretical benefits in terms of less information loss, its actual implementation incurs extra hardware expense in the form of garbage outputs and complicated inter-connections. In contrast, the majority-gate-based ALU took use of the functional completeness and compact nature of majority logic to effectively realise arithmetic and logic operations while reducing hardware complexity. To guarantee a fair comparison, both ALU designs were constructed and synthesised using the same FPGA platform.

The synthesis findings show that the majority-gate-based ALU beats the reversible-logic-based ALU in terms of both area utilisation and power consumption. Specifically, the majority-logic-based architecture reduced LUT utilisation by 8.0% and power consumption by 4.72% when compared to the reversible-logic-based ALU. These enhancements demonstrate that majority logic is more suited for low-power and area-efficient VLSI implementations of ALUs, especially in FPGA-based architectures. Overall, this research demonstrates that, although reversible logic remains intriguing for future computing paradigms, majority logic is a more efficient and practical answer for today's low-power digital systems.

References

[1] Dhoumendra Madal, Sumana Mandal and Sisir Kumar Garai, "Design of all-optical binary one bit comprator using eversible logic gates", IEEE, 2017. DOI: 978-1-5386-1703-8/17.

[2] A Kamaraj and P.Marihamy, "Design and implementation of Arithemetic and Logic Unit (ALU) using novel rversible logic gates in quantum cellular automa", IEEE,2017. DOI: 978-1-5090-4559 4/17.

[3] Mojtaba Valinataj, Mahboobeh and Hamid Jazayeri, " Novel low-cost and fault tolerant reversible logic adders", Computers nd Electrical Engineering, Elsevier, 2016, pp 56-72.

[4] Zhijin Gaun, Weiping Ding, Yueqin Hang, "An Arithemetic logic unit design based on reversible logi gates," IEEE Pacific RIM Conference on recent trends in electronics informaton communication technology, 2011, pp 925-931.

[5] Deeptha A, Drishika Muthanna, Dhrithi M, Pratiksha M and B S Kariyappa, "Design and Optimization of 8-bit ALU using Reversible logic," IEEE International cinference on Recent trends in electroics information communiction tchnology, 2016. DOI: 978-1-5090=0774 5/16.

[6] Y. Shamala and A.V.N Tilak, "Reversibl arithemetic logic unit," n proceedings of the 3rd international onference on electronics computer tehnology, Vol.5, 2011, pp. 207-211.

[7] R.Jayashree and M.Kiran Kumar, "Design and analysis of flip-flops using reversible logic," International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682 Vol.23, No23, March 2014.

[8] Avinash G. Keskar and Vishal R. Satpute, "Design of eight-bit novel reversible arithmetic and logic unit," 978-0-7695-4561- 5/11, IEEE, 2014.

[9] T. Naga Babu, D. Sounder, B. Subhakara Rao and P. Bose Babu, "A Low Power Adder Using Reversible Logic Gates," International Journal of Research in Engineering and Technology ISSN: 2319 1163, Volume: 01, Issue: 03, Nov 2012.

[10] Lenin Gopal, Nor Syahira Mohd Mahayadin, Adib Kabir Chowdhury,Alpha Agape Gopalai, Ashutosh Kumar Singh "Design and Synthesis of Reversible Arithmetic and Logic Unit," IEEE International Conference on computer, communication, control technology, pp 139 – 143, 2014.

[11] Umesh Kumar and Lavisha Sahu and Uma Sharma, "Performance Evaluation of Reversible Logic Gates," 978-1-5090- 5515-9/16, IEEE, 2016.