ISSN NO: 2249-3034

DESIGN AN AREA EFFICIENT KOGGE STONE ADDER USING PASS TRANSISTOR LOGIC

Dr.WINSTON DUNN

Department of Electronics and Communication Engineering
Faculty[1]Narasaraopeta Engineering College,Narasaraopet,A.P, India
Student[2-6] Narasaraopeta Engineering College, Narasaraopet, A.P, India

ABSTRACT:

This Paper represents the development of an area-efficient KOGGE Stone adder utilizing pass transistor logic. Here we use cadence software with 45nm technology. In recent technologies of Electronics applications, Adder is an important source of any devices such as DSP, and VLSI applications. Adders are fundamental components in digital circuits. Adder circuits are relatively simple to design and implement, making them cost-effective and efficient. For which, many electronics application devices used the high speed and low power consumption adders namely Parallel Prefix Adder (PPA). The speed and performance of these systems depend largely on the efficiency and accuracy of the adders. Generally, PP Adders have less delay due to less waiting time of carry for the next addition. But the area consumption is more, in which the performance of the adders will decrease for higher-order bits addition and many transistors required for the prefix network. One way to reduce the area of a KSA is to use Pass Transistor Logic (PTL) instead of conventional CMOS logic. The proposed area efficient KSA design used the Pass Transistor Logic (PTL) and analysed the performance of particular design. The Performance results of PTL with PP-KSA design used the reduce number of MOS devices which yields less area consumption compared to basic design of 4-bit PP-KSA.

Keywords: Pass transistor logic, KOGGE stone adder, Area consumption, Very Large-scale integrated design, Parallel prefix adders, Micro-wind.

I. INTRODUCTION

A KOGGE Stone Adder (KSA) is a kind of parallel prefix adder that generates and propagates the carry bits in parallel, enabling quick binary addition. However, because a KSA's prefix network requires a lot of transistors, it consumes a lot of space. By substituting Pass Transistor Logic (PTL) for traditional CMOS logic, a KSA's size can be decreased. Transistors are switches used in PTL logic design to achieve Boolean functionalities. Less size, less delay, and lower power dissipation are among the benefits that PTL offers over CMOS. To create the KSA architecture, we employ a combination of both and or gates together with AND gates, OR gates, XOR gates, and half adders. Using pass transistor logic in a modified KSA. We constructed gates (pass transistor and gate, or gate, half adder, XOR gates) by utilizing pass transistors.

A thorough analysis of adder architectures and pass transistor logic concepts is done at the outset of the project. The design process for the KOGGE Stone adder is then covered, with an emphasis on reducing the number of transistors while keeping performance parameters like delay and power consumption. Before we discuss the design of KSA using PTL, let us first review the design of KSA using CMOS, which is the conventional logic design style. CMOS is a logic design style that uses complementary pull-up and pull-down networks of PMOS and NMOS transistors to realize Boolean functions. CMOS has some advantages, such as full-swing output, noise immunity, and zero static power dissipation. However, CMOS also has some disadvantages, such as high area, high power consumption, and high delay. To get around this, we construct the KSA using Pass Transistor Logic (PTL) rather than traditional CMOS logic. In PTL logic design, transistors are used as switches instead of complementary pull-up and pull-down networks to realize Boolean functions. PTL is superior to CMOS in a few ways, including less space, and reduced power dissipation. Static power dissipation, charge sharing, and threshold loss are some of the difficulties PTL faces. To get over these problems and outperform CMOS, PTL must be properly designed and optimized.

The KOGGE Stone adder, characterized by its recursive structure and efficient carry lookahead mechanism, provides a solid foundation for high-speed addition. Integrating pass transistor logic into this architecture introduces the potential for a compact and power-efficient design. The choice of AND gates, XOR gates, OR gates, HALF ADDER circuits, and combinations thereof, as the basic building blocks, enables us to capitalize on the benefits of pass transistor logic while preserving the inherent advantages of the KOGGE Stone adder. In this context, the KOGGE Stone adder has emerged as a noteworthy architecture, offering parallelism and efficient carry propagation. This project seeks to enhance the area efficiency

ISSN NO: 2249-3034

of the KOGGE Stone adder by employing pass transistor logic—a versatile logic style known for its potential in reducing area and power consumption.

II. EXISTINGMETHODS

Existing Parallel Prefix KOGGE Stone Adder:

1) Two input and gate:

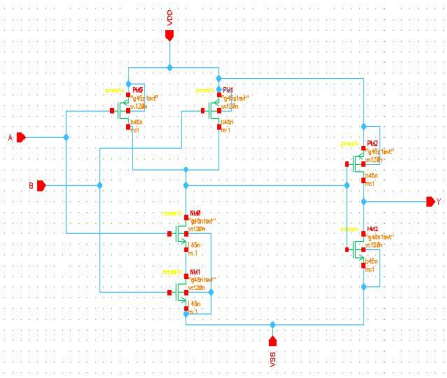


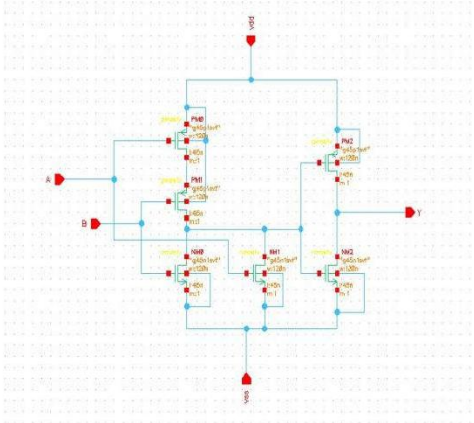
Fig 1:Schematicof two input AND gate.

A CMOS (Complementary Metal-Oxide-Semiconductor) 2-input AND gate is a digital logic gate that performs the AND operation using two input signals. CMOS technology utilizes both NMOS (N-type Metal-Oxide-Semiconductor) and PMOS (P-type Metal-Oxide-Semiconductor) transistors to implement logic gates. The output is 1 only when both Input A and Input B is 1.

2) Two input OR gate:

Fig 2: Schematicof two input OR gate.

A CMOS (Complementary Metal-Oxide-



Semiconductor) 2-input OR gate is a digital logic gate that performs the OR operation using two input signals. In CMOS technology, both NMOS (N-type Metal-Oxide-Semiconductor) and PMOS (P-type Metal-Oxide-Semiconductor) transistors are used to implement logic gates. The output is 0 only when both Input A and Input B is 0. If either Input A or Input B is 1, or if both are 1, the output is 1.

3) Two input XOR gate:

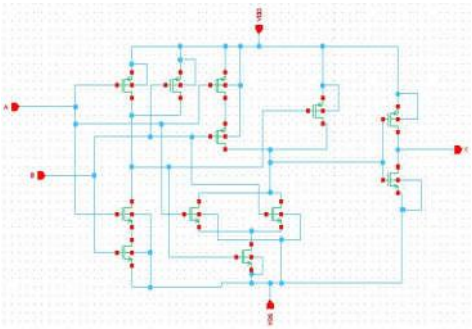


Fig 3:Schematicof two input XOR gate.

A CMOS (Complementary Metal-Oxide-Semiconductor) 2-input XOR gate is a digital logic gate that performs the exclusive OR (XOR) operation using two input signals. In CMOS technology, both NMOS (N-type Metal-Oxide-Semiconductor) and PMOS (P-type Metal-Oxide-Semiconductor) transistors are used to implement logic gates.If both Input A and Input B are 0 or both are 1, the output is 0 (logical XOR operation). If either Input A or Input B is 1 (but not both), the output is 1. The output is 0 only when both Input A and Input B are at the same logic level.

4) Two input HALF ADDER:

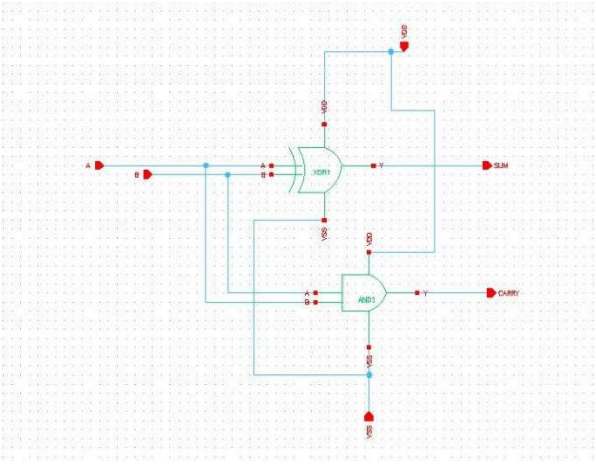


Fig 4: Schematicof two input HALF ADDER.

A CMOS 2-input half adder is a digital logic circuit that performs the addition of two binary inputs, usually denoted as A and B, and generates the sum (S) and carry-out (Cout) outputs. Half adder uses both AND gate and XOR gate. The XOR gate performs the sum operation: S=A⊕B.

The output of the XOR gate represents the sum of the two inputs. The AND gate performs the carry operation: Cout=A·B. The output of the AND gate represents the carry-out. A CMOS 2-input half adder is a digital logic circuit that performs the addition of two binary inputs, usually denoted as A and B, and generates the sum (S) and carry-out (Cout) outputs.

5) Two input combinational circuit:

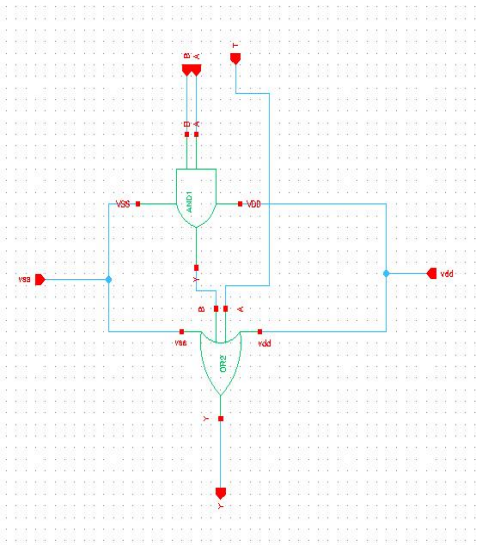


Fig 5: Schematicof two input Combinational circuit.

A combination circuit using both AND gate and OR gate is a common arrangement in digital logic design. Where two inputs, A and B, are processed through an AND gate and the result is then fed into an OR gate along with another input, C. The output of this combination circuit will be based on the logical operations performed by the AND gate and OR gate. AND gates are used to generate the G signals by performing logical AND operations on the input bits. OR gates are used to combine the G and P signals to determine the final carry-out.These gates are used for additional logic functions, such as generating the final output carry. The output of the AND gate is given as one input to the OR gate along with input T. The overall operation is to check if either the result of the AND gate (A AND B) or T is high (1). If any of these conditions is true, the output Y will be high (1); otherwise, it will be low (0).

5) Existing 4-bit parallel prefix kogge stone adder:

An equivalent prefix T KOGGE stone adder makes use of four AND gates, four half-adders, five circuits with combinational AND and OR gates, one OR gate, and four XOR gates. The inputs for a half adder serve as the circuit's inputs. And the outputs are linked to the XOR outputs. Here, generation and propagation are carried out in parallel, with the inputs being A0, A1, A2, A3, B0, B1, B2, B3, and Cin. So it's beneficial to minimize the design's area usage. Standard logic gates, such as AND, OR, and XOR, are used to perform the design.

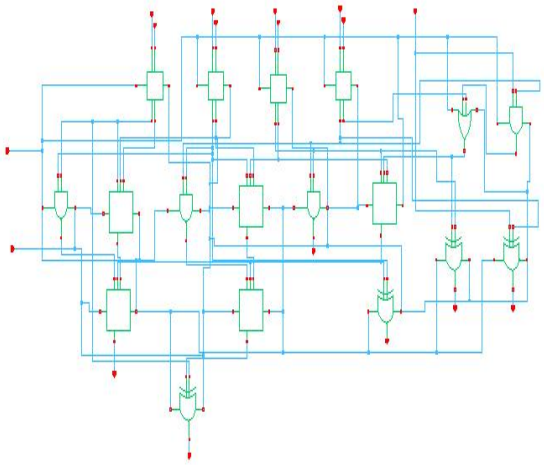


Fig 6: Schematic of Existing 4-bit parallel prefix KOGGE stone adder

The KOGGE Stone adder is a parallel prefix adder known for its area efficiency and high-speed operation. It can add multi-bit numbers in a highly parallelized manner, making it suitable for applications requiring fast arithmetic operations. Here's how the area-efficient KOGGE Stone adder works using the specified components:

a) Half Adders: Half adders are used to compute the sum and carry of two bits.

b) AND Gates: AND gates are used to compute the AND operation between input bits.

c) OR Gate: OR gates are used to compute the OR operation between input bits.

d) XOR Gates: XOR gates are used to compute the XOR operation between input bits.

e) Combination of AND + OR Gates: These gates are used for additional logic functions, such as generating the final output carry.

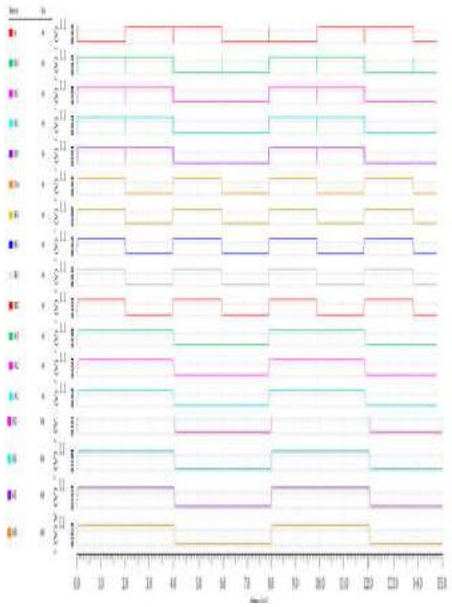


Fig 7: Output waveforms of 4-bit parallel prefix KSA.

There are four main steps in the Parallel Prefix Adder:

1. Generate Partial Sum and Carry:

a) To find the sum of each pair of input bits, utilize XOR gates.

b) To ascertain whether a carry exists between corresponding input bits, use AND gates.

c) Calculate the ultimate carry-out using OR gates.

2. Compute Propagate and Generate Signals:

a) Ascertain whether a carryover from stage one (P) has been carried over to stage two.

b) Determine if a carry (G) is produced in the present stage.

3. Ripple Carry Chain:

a) Distribute signals via a chain of ripples.

b) Each stage considers the propagate (P) and generate (G) signals from the stages before it when generating the final carry.

4. Calculating the Final Sum:

a) To determine the final sum, utilize XOR gates.

III. PROPOSEDMETHODS

Proposed Parallel Prefix KOGGE Stone Adder:

1) Two input and gate using pass transistor logic:

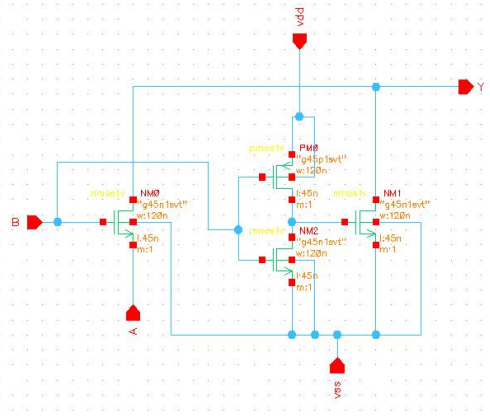


Fig 8: Schematicof two input AND gate using PTL.

An AND gate can be created with pass transistors and inverters in pass transistor logic. Transmission gates, or pass transistors, are used in pass transistor logic, a type of digital circuit design, to carry out logic operations. This is an example of a simple pass transistor logic two-input AND gate. There are two input signals, A and B. The input signals A and B control the pass transistors, which function as switches. The output of the pass transistors is connected to an inverter to implement the AND gate logic. The pass transistors are activated when both A and B are high, allowing the signal to reach the inverter. If either A or B is low, the output will be low and one of the pass transistors will be OFF, blocking the signal from flowing through.

2) Two input or gate using pass transistor logic:

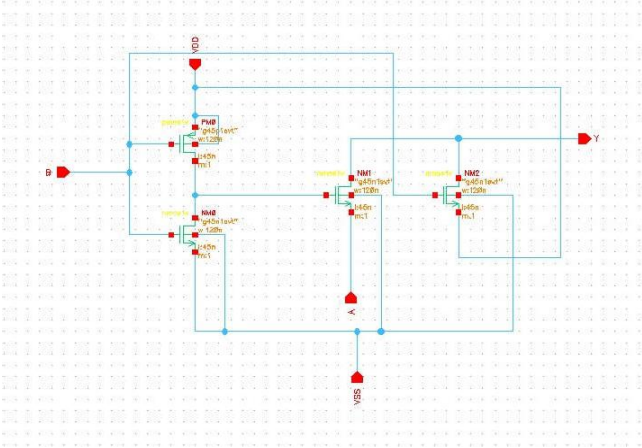


Fig 9: Schematicof two input OR gate using PTL.

An OR gate using pass transistor logic can be implemented by combining NMOS (n-channel metal-oxide-semiconductor) and PMOS (p-channel metal-oxide-semiconductor) transistors. Here's a simple circuit for a two-input OR gate using pass transistor logic. For an OR gate, the output is high when at least one input is high, and the output is low only when both inputs are low.

ISSN NO: 2249-3034

3) Two input XOR gate using pass transistor logic:

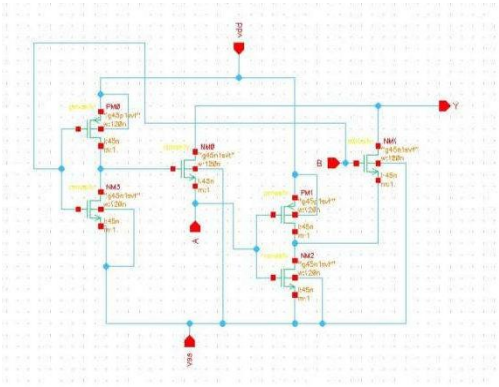


Fig 10: Schematicof two input XOR gate using PTL.

A pair of complementary pass transistors (nmos and pmos) connected in parallel make up the transmission gate, the fundamental building block of pass transistor logic (PTL). With PTL, you can utilize a transmission gate and an inverter to create a two-input XOR gate. These two signals are the inputs: A and B. Combining transmission gates and inverters to achieve XOR capability is represented by the XOR Gates block. When both input signals (A and B) are active, the transmission gates let the signals to pass through. The signal can pass through to the common node while A is at logic '0' because the NMOS transistor connected to A is turned ON. The signal can pass through to the common node when B is at logic '0' because the NMOS transistor connected to B is turned ON. On the other hand, complementary (inverted) signals are also introduced for XOR. The PMOS transistor connected to A turns ON when it reaches logic '1'.

4) Two input HALF ADDER using pass transistor logic:

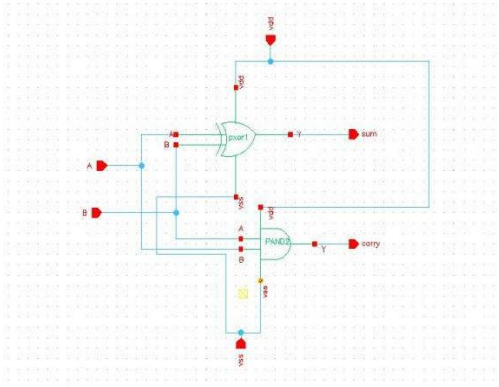


Fig 11: Schematicof two input HALF ADDER using Pass Transistor Logic.

A half adder is a digital circuit that adds two binary digits (bits) and produces a sum bit (S) and a carry bit (C). In pass transistor logic, we can implement a half adder using pass transistors. The logic for a half adder consists of an XOR gate and an AND gate. Combining both XOR gate and AND gate, we get the pass transistor logic implementation of a half adder:

Sum (S): Output from the XOR gate.

Carry (C): Output from the AND gate.

The pass transistor logic for a half adder considers the behavior of XOR gate and AND gate to perform binary addition. The XOR gate generates the sum bit, and the AND gate generates the carry bit.

5) Two input combinational circuit using pass transistor logic:

This combination creates a more complex logic function where the OR gate considers both the AND gate output and an additional input. AND Operation (A AND B): Use a transmission gate for each input pair (A and B). The output of the AND gate A AND B is taken to the input of the OR gate. The AND gate is implemented using transmission gates. These transmission gates are controlled by the complementary signals of A and B to achieve the logical AND operation.

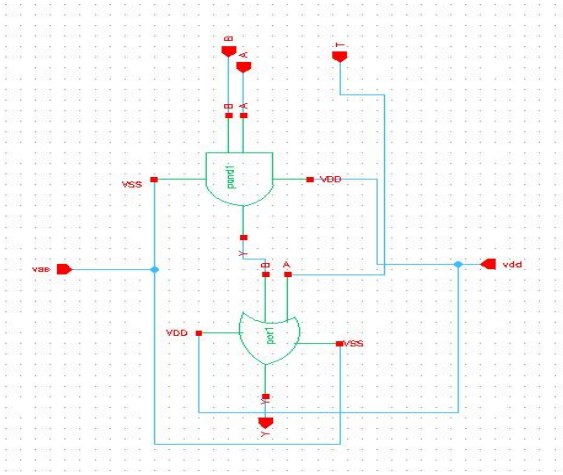


Fig 12: Schematicof two input combinational circuit using Pass Transistor Logic.

Use a transmission gate for the output of the AND gate and the input T to form an OR operation. If the result of A AND B is high or if T is high, the transmission gate allows the signal to pass. The output of the OR gate T is the final output of the combinational circuit.

ISSN NO: 2249-3034

6) Proposed 4-bit parallel prefix KOGGE stone adder using pass transistor logic:

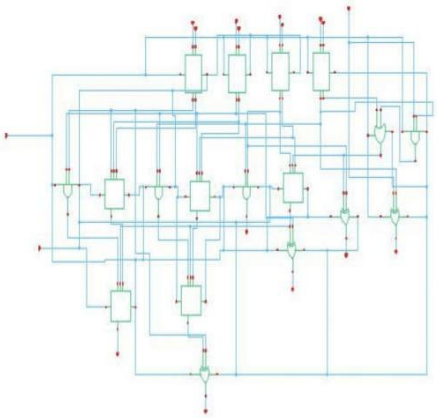


Fig 13: Proposed 4-bit parallel prefix KOGGE stone adder using pass transistor logic.

When compared to parallel prefix KOGGE stone adder, the circuit in Fig 12: using pass transistor logic provides less area architecture and low delay, it also provides better performance. It also reduces the number of transistors. The number of inverters used in design is automatically reduced. For each bit position, implement XOR gates using pass transistors. Pass transistors can be configured to selectively pass, or block signals based on the control input. XOR gates will generate the sum (S) for each bit position. Implement an OR gate using pass transistors. Pass transistor logic is configured to create the OR logic function. The OR gate combines the final carry-out values from each bit position.

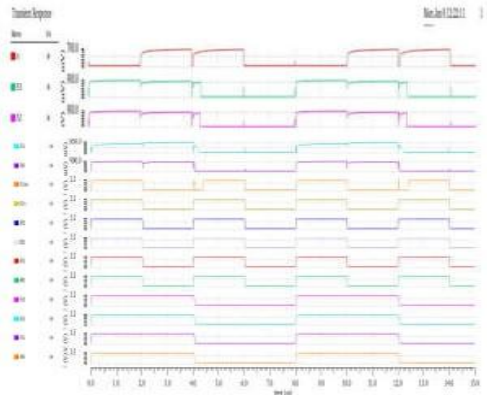


Fig 14: output waveform of Proposed 4-bit parallel prefix KOGGE stone adder using pass transistor logic.

For example, here applied input values for 4-bit to proceed execution of proposed KOGGE stone adder. Now put the given values as A = 1 0 1 0, B = 0 1 0 1, and C_{in} = 1, then S_0 =0, S_1 = 0, S_2 =0, S_3 =0, C_{OUT} = 1. In addition to that, the results have verified by waveform results for PP-KSA which are shown in Fig: 12.

IV. RESULTSANDDISCUSSION

ComparisonTableforResults

Name of theADDER	Number of Transistors used	Avg Power (µW)	Time Delay (ns)
4-Bit PP-KSA	210	176.61 **µW**	1.959 **ns**
4-Bit PP-KSA using PTL.	124	81.62 **µW**	0.663 **ns**

V. CONCLUSION

A 4-bit KOGGE stone adder using normal CMOS logic consumes more area and gives more delay. The number of transistors used is also more, 210 PMOS and NMOS transistors are used. Whereas 4-bit KOGGE stone adder using pass transistor logic consumes less area and less delay. And number of transistors used is less, 124 PMOS and NMOS transistors are used. The proposed PP-KSA using pass transistor is better than PP-KSA using CMOS logic due to a smaller number of MOS devices used. Here two designs of PP-KSA are discussed and compared the analyzed performance on aspects of area, power, delay, and number of transistors used.

VI. REFRENCES

[1] N. Poornima, V. S. Kancahana Bhsaskaran, “Area Efficient Hybrid Parallel Prefix Adders,” in Procedia Materials Science, Vol.10, PP: 371-380 (2015).

[2] S. Daphni, K. S. Vijula Grace, “Design and Analysis of 32-bit Parallel Prefix Adders for Low Power VLSI Applications” in Advances in Science, Technology and Engineering Systems Journal Vol. 4, No. 2, PP: 102-106 (2019).

[3] M. Hatem, El - Boghdadi., “A class of almost optimize size independent parallel prefix circuits”, in Journal of Parallel and Distributed computing, Vol.73, No.6, PP:888-894 (2013).

[4] Shilpa C. N, Kunjan D. Shinde, Nithin H. V, “Design, Implementation and Comparative Analysis of Kogge Stone Adder using CMOS and GDI design: A VLSI Based

Approach", in 8th International Conference on Computational Intelligence and Communication Networks IEEE conference, PP: 570-574 (2016).

[5] S. Daphni, K. S. Vijula Grace, "A review analysis of Parallel Prefix Adders for better performance in VLSI applications", in Proceedings of 2017 IEEE International Conference on Circuits and Systems, PP:103-106, (ICCS 2017).

[6] Amy Mariam George, G. Jyothish Chandran, "Design and analysis of 8-bit parallel prefix comparators using constant delay logic", in Procedia Technology, Vol.24, PP: 1178-1185 (2016).

[7] S. Rakesh, K. S. Vijula Grace, "A comprehensive review on the VLSI design performance of different Parallel Prefix Adders", in Materials today proceedings, Vol.11, Part 3, PP:1001-1009, (2019).

[8] A.P [8] Li – Wen Chang, Juan Gómez - Luna, "Chapter 8 - Parallel patterns: prefix sum: An introduction to work efficiency in parallel algorithms", in Programming Massively Parallel Processors (Third Edition), PP: 175-197 (2017).

[9] Marc Snir, "Depth-size trade-offs for parallel prefix computation", in Journal of Algorithms, Vol. 7, no. 2, (1986).

[10] Reshma Mary James, Ajith Ravindran, "Review of Full Adder Performance Analysis Using Kogge Stone Adder and Magnetic Tunnel Junction", in Fourth International Conference on Devices, Circuits and Systems (ICDCS'18), PP: 84-90 (2018).

[11] S. Daphni, K. S. Vijula Grace, "A review analysis of Parallel Prefix Adders for better performance in VLSI Applications" in Proceedings of 2017 IEEE International Conference on Circuits and Systems PP:103-106 (2018).

[12] Anjana. R, Abishna. B, Harshitha. M.S, Abhishek. E, Ravichandra.V, Dr. Suma M.S, "Implementation of Vedic Multiplier using Kogge-Stone Adder", International Conference on Embedded Systems - (ICES), PP:28-31 (2014).