



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Issue 10, October 2022

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.118

 9940 572 462

 6381 907 438

 ijirset@gmail.com

 www.ijirset.com



Design of High-Speed, Power-Efficient MAC Unit by Using Wallace Multiplier and PPA

Vanumu Mohan Venkata Santosh Kumar¹, Dr.A.M.Prasad²

Student, M.Tech, VLSI & Embedded Systems, Dept. of ECE, UCEK(A), JNTUK, Kakinada, India.¹

Proffesor, Dept. of ECE, UCEK (A), JNTUK, Kakinada, India.²

ABSTRACT: This project consists of we suggest a pipeline multiply-accumulate (MAC) architecture with high speed and low power consumption. Dadda multiplication was used to develop a multiply-accumulate unit in a standard MAC, and a small adder was created to accumulate all of the carriers in order to properly handle the overflow bit in the PPR process. The most significant bits will not be added to or accumulated in this design until the PPR process of the subsequent multiplication. Earlier we have an array multiplier in that there is more area and high power consumption, we came for a Dadda multiplier, so we can reduce the area. If we use the Mac unit with the Dadda multiplier, we will gain greater delay and power. Wallace tree multiplier is utilized in the suggested method to create the multiply-accumulate unit. Experimental findings demonstrate that the suggested MAC architecture can significantly reduce both power consumption and circuit latency when compared to prior efforts. This architecture is synthesized and simulated using Xilinx vivado 2018.3 version tool.

KEYWORDS:-Digital Circuits, Dadda multiplier, Wallace tree multiplier, Logic Circuits, Multiplying Circuits, Pipeline Processing, and Power Dissipation Delay.

I. INTRODUCTION

MAC means multiply-accumulate unit.. It is a low-power high-speed pipeline multiply-accumulate architect ture.An accumulator, multiplier, and adder make up a basic MAC unit. As speed is of the biggest significance in DSP, Carry-Select or Carry-Save adders are typically implemented. However, if an on-chip area is taken into account, Carry Increment Adder can be utilized with the added benefit of high speed. The Vedic Multiplier is one possible multiplier implementation.

A crucial building component for digital signal processing (DSP) applications is the multiply-accumulate (MAC) unit. An ordinary procedure is known as a multiply-accumulate (MAC) operation, which computes the product of two numbers and adds that product to an accumulator. The operation is frequently referred to as a MAC or a MAC operation, and the hardware component that performs it is referred to as a multiplier-accumulator (MAC unit). Additionally, it provides the microcontroller with signal processing capabilities for a variety of uses, including servo/audio control, etc.

The 2-stage pipelined arithmetic architecture used by MAC, an execution unit of the processor, improves multipliers. 8-bit and 16-bit operands are both supported by this design. Additionally, it accepts signed, fixed-point fractional input operands in addition to signed and unsigned integers.

The inputs for the MAC unit are read from memory cells and given to the multiplier block of the MAC, which multiplies the inputs and passes the result to an adder, which then stores the result in memory. The goal is to finish this operation in only one clock cycle. The suggested architecture includes a 65-bit accumulator that makes use of a carry choose adder, as well as a 32-bit floating point multiplier based on the Vedic multiplication technique. Instead of using a conventional multiplier to multiply the values of A and B, a floating point Vedic multiplier is used instead since it can speed up the MAC unit. In this architecture, the Carry Save Adder serves as an accumulator. Apparently, Carry chose an adder as the accumulator in addition to using the Vedic multiplier technique.

II. EARLIER WORK

For multiplying binary values, many people use array multipliers. A digital combinational circuit known as an array



multiplier is used to multiply two binary values by using an array of full adders and half adders. The numerous product keywords involved are added almost simultaneously using this array. Before the Adder array, an array of AND gates is utilized to create the various product terms.

By seeing the example we can get clarity on the array multiplier.

Ex: 1 0 1 1 11
 *1 1 0 1 13

1	0	1	1					
0	0	0	0		(+)1 st sum			
0	1	0	1	1				
1	0	1	1		2 nd sum			
1	1	0	1	1	1			
1	0	1	1					
1	0	0	0	1	1	1	1	Ans

Fig (1) : example

Here 11 is the multiplier and 13 is the multiplicand. In this LSB bit “1” of 13 is multiplied by 1011 .then 0 by 1011 then we will add these and we will get a 1 st sum. Like this multiplication takes place.

Disadvantages: 1)More delay.

2)High power consumption.

So, we came for a conventional mac unit that is dadda multiplier.

In the array multiplier we have a large delay and high power consumption so that is reduced by the dadda multiplier. The dadda multiplier follows the different structure that is inverted triangle so we can greatly reduce the delay and power.

II.CONVENTINAL MAC

A multiplier and an accumulator are the two separate building components that make up a traditional MAC unit (i.e., an accumulate adder). A (2N+1)-bit accumulator (adder) and an N-bit multiplier are both components of an N-bit MAC unit, where is the number of guard bits used to prevent overflow

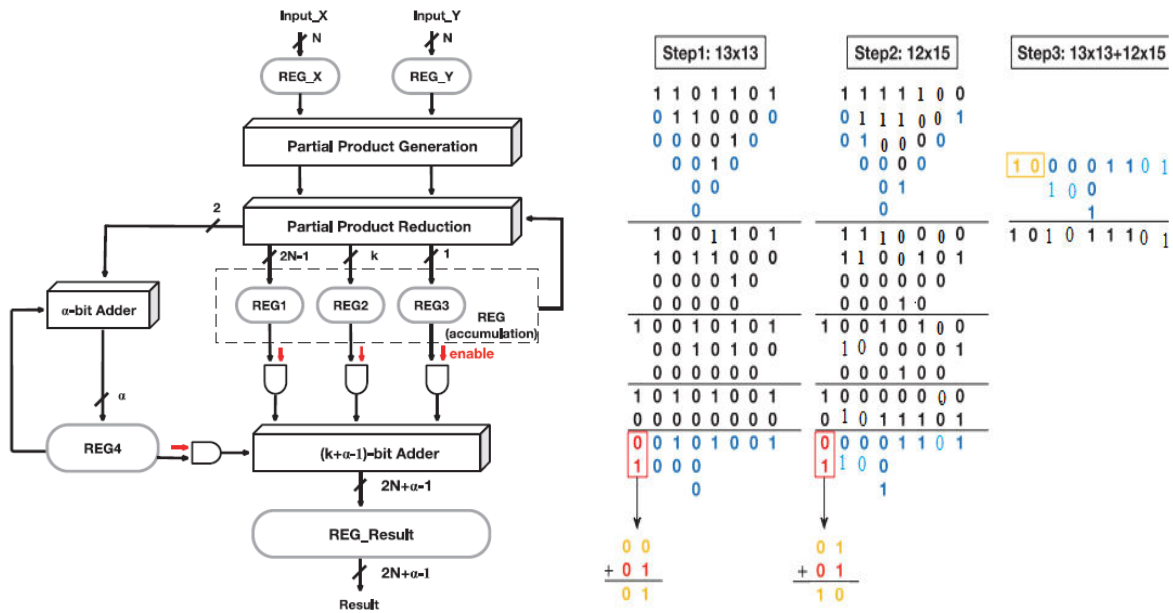
A multiplier usually has three steps:

1. partial product generation (PPG) process.
2. partial product reduction (PPR) process
3. final addition

In our general multiplication 15*13 , initially y,ou will multiply 3 with both 5 and 1 after that 1 with 5 and 1. So there isa total of 4 partial products are formed.Like this, while multiplying 4*4 bit multiplier it will generate 16 partial products. while multiplying 8*8 bit multiplier will generate 64 partial products are generated. Once we complete your partial product generation we have to reduce one row of final output which is the main aim of our multiplication.

III. MAC ARCHITECTURE

This is an out MAC architecture, we have to provide one multiplayer and one multiplicand that is input x and y is stored in our registers. From that, we have generated partial product generation.in our next step we are adding zeros to the partial products. In REG1 we are adding (2N-1) i,e 7 zeros .in REG2 we are adding k i,e is 4 zeros . In the REG3 we are adding 1 zero to the partial products that are clearly shown below example. In partial products are reduced by using half adders and full adderswe reduce to rows of output.



Fig

The excess bits are generated that given to the a α - bit adder. That is stored in REG 4. The 1 st result is stored in the $(k+\alpha-1)$ bit adder. In the next step we gave 2 input numbers again in X and Y. they generate partial products and in the next step, they add by the 1 st result of 2 input numbers. The final addition takes place in the REG_result. Then we get the final result that can be shown in the example below.

The third step is the final addition. An adder (called the final adder) is used to perform the summation of the final two rows. Formultiplicandmultiplier, a $(2N-1)$ -bit adder is required for the final addition.

In a conventional MAC unit, it is necessary to perform two carry propagations: additions in multiplications and additions in accumulations. Note that the carry propagation is time-consuming. Therefore, in, the multiplier output is fed back to the input of the PPR process. Since the accumulation is handled by the final adder, only one carry propagation is required. Moreover, based on this architecture, the area of the 16-bit MAC unit can be further reduced by 3.2% by fully utilizing the compression.

Drawbacks:

- large power consumption and
- large path delay

we came for Wallace multiple.

IV. PROPOSED MAC

The pipelined MAC unit, Wallace tree multiplier, and accumulation unit are all used in the suggested technique. An enhanced variation of tree-based multiplier architecture is the Wallace tree multiplier. To cut back on latency, it employs the carry save addition method. To limit our PPM to two rows for the PPR process, we use the Wallace tree technique. The fundamental cause is that our PPM differs theDadda tree strategy can employ less counters for our PPM than the Wallace tree approach.We conduct the $(2N-k-1)$ -bit final addition after reducing our PPM to two rows. A three-row PPM is as a result obtained, as shown in Fig. The three-row PPM will be saved in register accumulation, so take note of it. The subsequent cycle's PPM for the PPR process will then be formed by combining the register accumulation result with the PPG result.

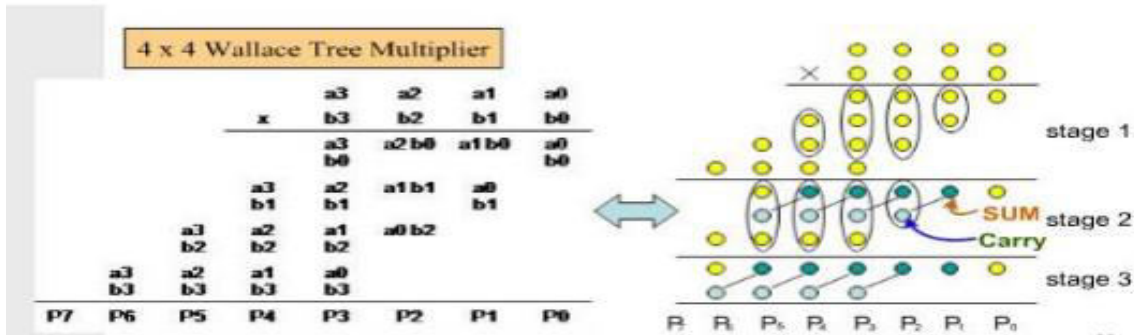


Fig.4. 4x4 Wallace tree multiplier with partial product and various stages

We are utilizing a translation circuit in this case to handle the overflow bits, and the translated output will be provided to an alpha bit adder. The MAC unit's translation circuit. The output is equal to the sum of the two input values since the two input values match the least significant bit of the -bit adder. When compared to previously used approaches for MAC unit implementation, the standard method, which uses a Dadda multiplier, results in more delay and power consumption.

The Wallace tree multiplier is our suggested solution to the Dadda multiplier's disadvantage. We will experience better delay and power consumption performed.

VI. EXPERIMENTAL RESULTS

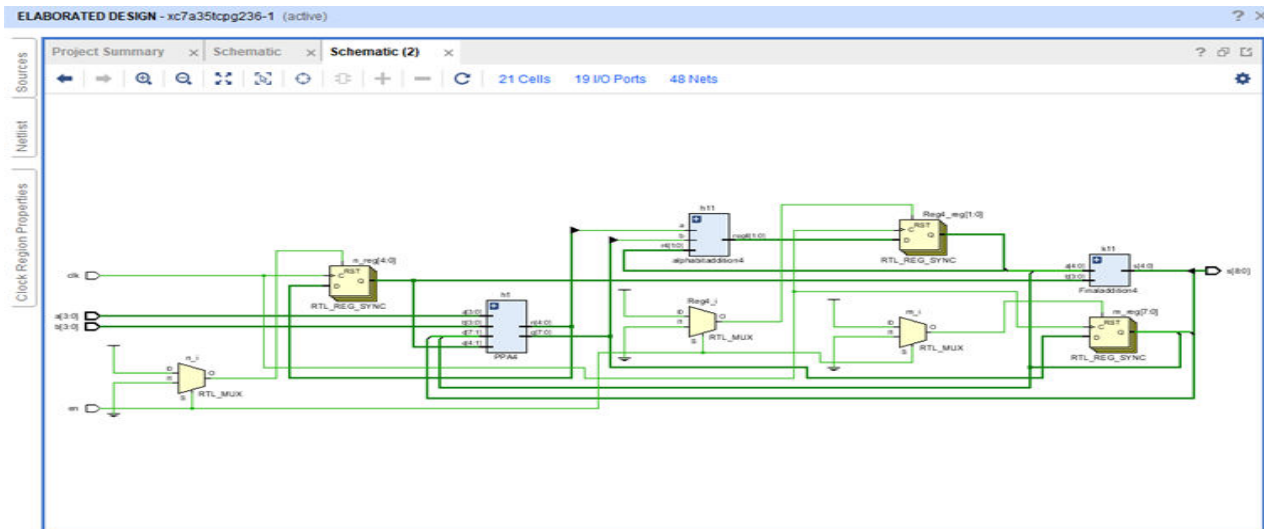


Fig.5:RTL Schematic

Similar to RTL schematic, technology schematic can also be generated after synthesizing the code. It consists of more number of block which is not possible to show that's why top block is only shown in the below fig.

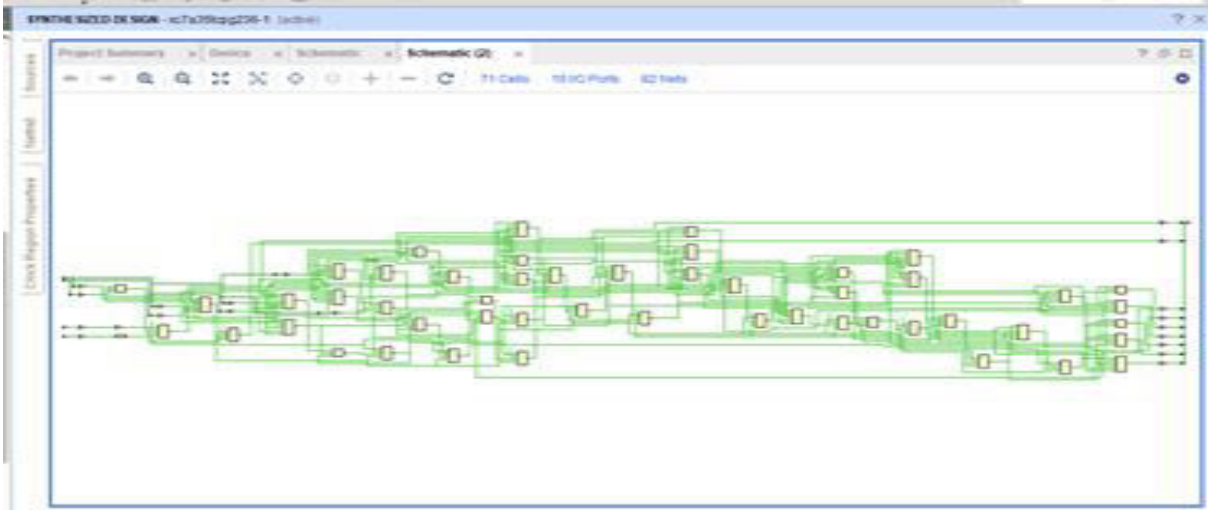
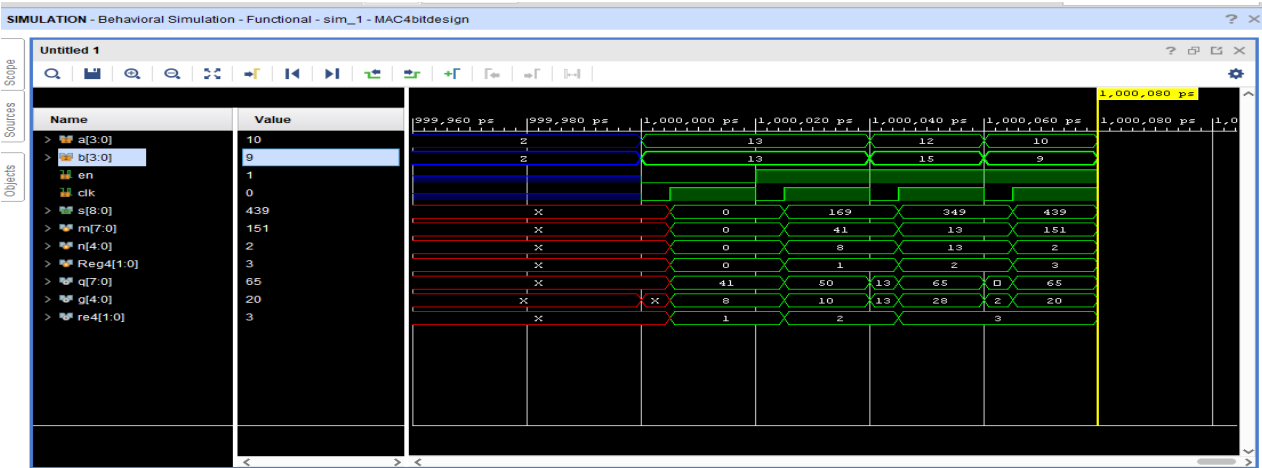
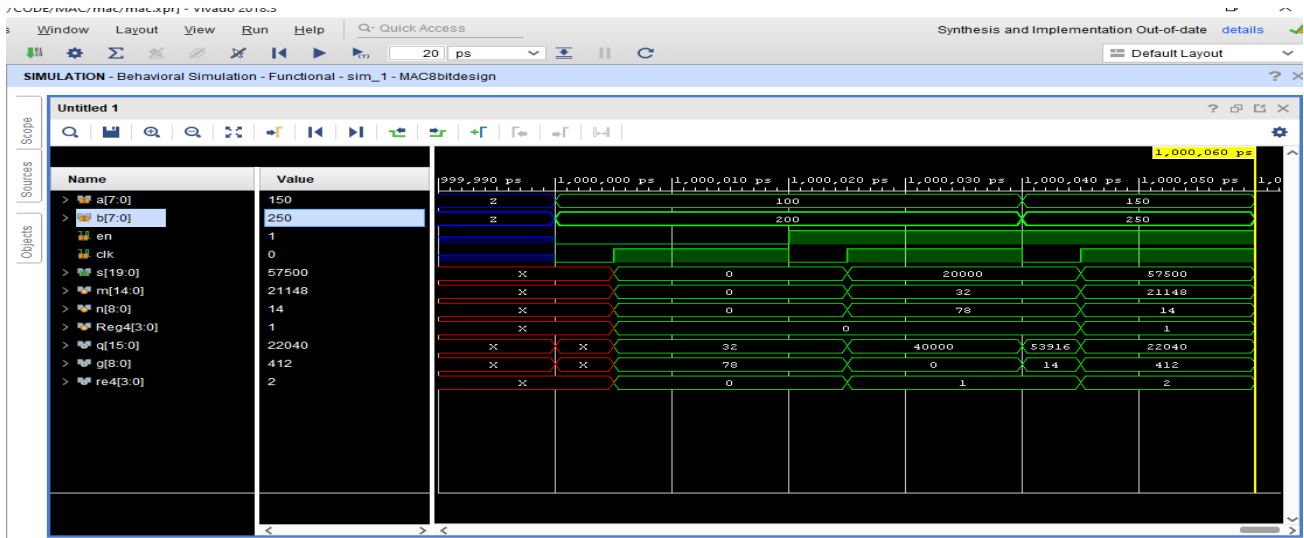


Fig.6:Technology Schematic

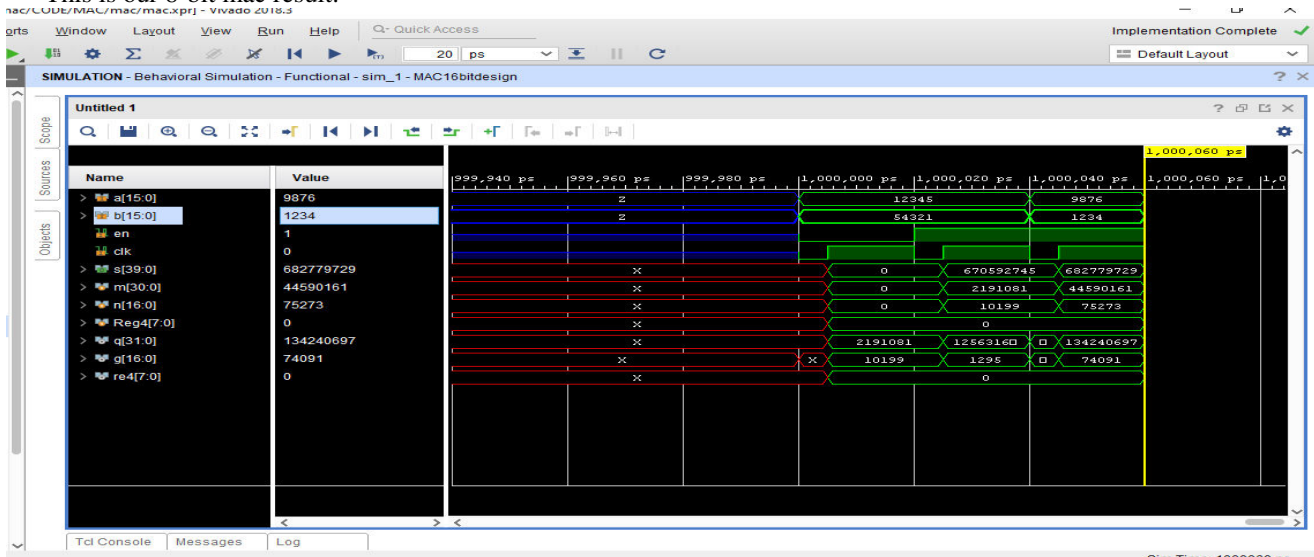
To check the correctness of the output and to observe the output waveform we need to do simulation process after completing the Verilog code. The output waveform is shown in Fig. 7



4 – bit mac result in this $13 * 13 = 169$, $12 * 15 = 180$, which is added to $169 + 180 = 349$ we are getting the result .169 is accumulate to 180 and finally, we get 349.



This is our 8-bit mac result.



VI. EVALUATION TABLE FOR AREA, DELAY, POWER:

MAC ByDadda Multiplier results

	4- BIT	8-BIT	16-BIT
(Report utilization)			
Lut			
FF	31	127	413
IO	13	27	56
	19	28	74
POWER	8.965 mw	19.95mw	44.65mw
DELAY	7.208 ns	9.791ns	15.056ns



MAC By Wallace multiplier results:

	4- BIT	8-BIT	16-BIT
(Report utilization)			
Lut			
FF	35	127	442
IO	13	31	56
	19	38	74
POWER	8.425 mw	19.18mw	44.65mw
DELAY	5.715 ns	9.522ns	15.056ns

In this mainly power and delay is reduced on comparison between the dadda multiplier and Wallace multiplier. We observe that area is increased in the Wallace multiplier whybecause the structure is different. The dadda multiplier follows the inverted triangle while the Wallace multiplier follows the parallelogram structure.

VII. CONCLUSION

We present a low-power, high-speed two-stage pipelined Multiply Accumulate Unit (MAC) for DSP applications in this project. The basic idea is to incorporate some additions into the PPR process. Critical path delays caused by carry propagations can thus be reduced. To handle overflow correctly during the PPR process, a -bit adder is used to count the total number of carries. In the proposed method, we implement a MAC unit with a Wallace tree multiplier to achieve better performance in terms of delay and power consumption when compared to a MAC unit based on a Dadda multiplier. . We observe that area is increased in the Wallace multiplier whybecause the structure is different. The dadda multiplier follows the inverted triangle while the Wallace multiplier follows the parallelogram structure.

REFERENCES

- [1] A. Abdelgawad, "Low Power Multiply Accumulate Unit (MAC) for Future Wireless Sensor Networks", in *Proc. IEEE Int. Symp. Sensors Applications*, Galveston, TX, 2013.
- [2] H.O. Ahmed, M. Ghoneima and M. Dessouky, "Concurrent MAC Unit Design using VHDL for Deep Learning Networks on FPGA", in *Proc. IEEE Int. Conf. Computer Applications and Industrial Electronics*, Penang, Malaysia, 2018.
- [3] V. Camus, C. Enz and M. Verhelst, "Survey of Precision-Scalable Multiply-Accumulate Units for Neural-Network Processing", in *Proc. IEEE Int. Conf. Artificial Intelligence Circuits and Systems*, Hsinchu, Taiwan, 2019.
- [4] W.J. Townsend, E.E. Swartzlander, J.A. Abraham, "A Comparison of Dadda and Wallace Multiplier Delays", in *Proc. SPIE Annual Meeting Optical Science and Technology*, San Diego, CA, 2003.
- [5] K.L.S. Swee and L.H. Hiung, "Performance Comparison Review of 32-Bit Multiplier Designs", in *Proc. IEEE Intelligent and Advanced Systems*, Kuala Lumpur, Malaysia, 2012, pp. 836-841.
- [6] S. Asif and Y. Kong, "Design of an Algorithmic Wallace Multiplier using High Speed Counters", in *Proc. IEEE Int. Conf. Computer Engineering & Systems*, Cairo, Egypt, 2015, pp. 133-138.
- [7] C.W. Tung and S.H. Huang, "Low-Power High-Accuracy Approximate Multiplier Using Approximate High-Order Compressors", in *Proc. IEEE Int. Conf. Communication Engineering and Technology*, Nagoya, Japan, 2019.
- [8] C. Nagendra, M.J. Irwin and R.M. Owens, "Area-Time-Power Tradeoffs in Parallel Adders", *IEEE Trans. Circuits and Systems -- II: Analog and Digital Signal Processing*, vol. 43, no. 10, pp. 689-702, Oct. 1996.
- [9] J. Saini, S. Agarwal and Aditi Kansal, "Performance, Analysis and Comparison of Digital Adders", in *Proc. IEEE Int. Conf. Advances in Computer Engineering and Applications*, Ghaziabad, India, 2015, pp. 81-83.
- [10] L. Pilato, S. Saponara and L. Fanucci, "Performance of Digital Adder Architectures in 180nm CMOS Standard-Cell Technology", in *Proc. IEEE Int. Conf. Applied Electronics*, Pilsen, Czech Republic, 2016.



INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.118



ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



www.ijirset.com

Scan to save the contact details