iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** V **Month of publication:** May 2021

DOI: https://doi.org/10.22214/ijraset.2021.34753

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue V May 2021- Available at www.ijraset.com

MAC Unit Optimization for Area, Power and Timing Constraints

Karthik S Kulkarni[1], H V Ravish Aradhya[2]
[1, 2]*Dept. of Electronics and Communications, R V College of Engineering*

Abstract: ***A MAC unit is also known as the Multiply-Accumulate unit which is in the most demand of Digital Signal Processing (DSP). It is used to perform functions like addition and multiplications. MAC unit drives in two stages. Firstly, the multiplier computes a given number and the result are forwarded to the next stage. In the second stage, accumulation operation takes place. MAC unit was optimized for power, area and timing parameters using advanced adders and multiplying algorithms. For design, which was optimized for timing parameter, slack timing obtained was 3.058ns using Brent Kung adder and Modified booth algorithm. For area optimized design, area was 129 um2 implemented using Ladner Fischer adder and Modified booth multiplier. For power optimized MAC unit, total power drawn was 0.1797 mw implemented using Brent Kung adder and Karatsuba multiplier. Optimization of MAC unit with respect to one or more than one parameter facilitates designer to choose for appropriate applications. Designs can be further optimized using low power designing techniques at architectural level. Different behaviour can be observed for different size and technologies.***
Keywords: Brent Kung, Ladner Fischer, MAC, Karatsuba, Modified booth, Accumulate.

I. INTRODUCTION

Digital Signal Processors (DSPs) involve multiplication and/or accumulation operations as a major portion and MAC units play a crucial role in implementing it, especially in high performance DSPs. Previous successive products are added and stored in a register. The MAC inputs are acquired from the register location and given to the multiplier [1][2]. The design comprises of combination of advanced adders and multipliers. Efficient adders include Brent Kung adder and Ladner Fischer Adder. Advanced multipliers comprise of Modified booth multiplier and Karatsuba Multiplier. MAC unit operates in two stages. Firstly, the multiplier computes a given number and the result are forwarded to the second stage. In the second stage, accumulation operation takes place. Multiplier takes two 8-bit numbers as inputs. For modified booth multiplier, first 3-bit numbers are taken for evaluating partial product. Bits are then right shifted, and all partial products are obtained. Partial products are added by left shifting. Obtained sum is product of the numbers. The new product is added with the previous product. Product is stored in a register.

For Karatsuba Multiplication, multiplier and multiplicands are split into 2 nibbles each. Product of higher nibbles of multiplier and multiplicands are obtained, and product is left shifted by 4 times before storing. Similarly, product of lower nibbles of multiplier and multiplicands are obtained. Besides this, product of sum of higher and lower nibbles of multiplier and multiplicand are gained. Difference of the products obtained in these steps is then left shifted by 4 times. Product from the individual steps are then added to derive a final product.

In paper [3], MAC unit proposed supports one 16-bit MAC operation or addition of two 8-bit multiplications plus a 16-bit sum. To style the proposed MAC unit more versatile adaptable, the bit-width of exponent and mantissa can be malleably exchanged Proposed MAC unit provides more springiness with only 21.8 percent area overhead. It demands much less hardware cost.

In paper [4], a 32-bit MAC using 32-bit array multiplier and RCA (ripple carry adder) along with another MAC using Vedic multiplier and RCA has been proposed. It was observed that the MAC unit using vedic multiplier and RCA was 33 percentage efficient compared with other MAC units in terms of delay.

Vedic Multiplier is one of the efficient multipliers to decrease the delay and improve the performance. Reversible MAC Unit gives lesser delay and area when compared with the conventional MAC Unit. MAC unit with Carry Save adder gives a better performance when compared with that of MAC unit with Carry Look- Ahead Adder [5]. Author gives a comparison of the design constraints for 32 bits.

A comparison with the prevailing 8-bit vedic multiplier employing Square-Root (SQR) Carry-Select Adder (CSLA) is offered. The projected design satisfies with great improvement in delay and area. In addition, power reduction around 9.5% is achieved in [6].

Author proves that a Modified booth algorithm harvests less delay in contrast with a generic multiplication process and it also curbs the number of partial products in [7]. The Carry look-ahead adder is used for restricting the surplus delay of MAC unit.

The projected MAC shows better functioning compared to generic method and has benefits of lesser area overhead and critical path delay. This new high-speed hybrid carry look-ahead adders are simulated and synthesized using 90 nm Design Compiler.
The projected CSA tree employs 1's-complement-based radix-2 modified Booth's algorithm (MBA) and devours the modified array for the sign extension to amplify the bit density of the operands. Amalgamating multiplication with accumulation and developing a mixed type of carry save adder (CSA), the execution was improved. Since accumulator that has highest latency in MAC was combined into CSA, overall performance was improved. The detailed work and architecture is presented in [8].

II. MODIFIED BOOTH MULTIPLIER

Multiplication can be regarded as a flow of repeated additions. The number to be summed is the multiplicand, the number of times that it is summed is the multiplier, and the result is the product. Each step of summation synthesizes a partial product. A multiplier can be divided into three operational steps:

A. The rst is radix-2 encoding in which a partial product is generated from the multiplicand (X) and the multiplier(Y).
B. The second is adder array or partial product compression to add all partial products and convert them into the form of sum and carry.
C. The last is the addition in which the multiplication result is produced by adding the sum and the carry.

To Booth recode the multiplier term, consider the bits in chunks of three, so that each chunk overlaps the previous chunk by one bit. Grouping starts from the Least Significant Bit. Figure 1 shows the grouping of bits and chunk overlaps the previous chunk by one bit.

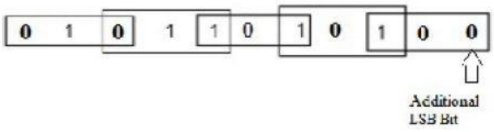


Fig. 1 Grouping of Bits

The Booth recoding procedure, then, is as follows:
1) Working from LSB to MSB, substitute each 0 digit of the given number with a 0 in the recoded number until a 1 is met.
2) When a 1 is met, put a 1 at that position in the recoded number, and omit over any succeeding is until a 0 is met. Table 2.1 shows the recoding of bits.
3) Swap that 0 with a 1 and continue.

TABLE 1
RECODING OF BITS

000	add 0
001	add multiplicand
010	add multiplicand
011	add 2*multiplicand
100	subtract 2*multiplicand
101	subtract multiplicand
110	subtract multiplicand
111	subtract 0

III.KARATSUBA MULTIPLIER

Recursive Karatsuba is based on incorporating Karatsuba algorithm repeatedly at every stage to improve speed when bit size is high. The algorithm works on separating the bits (N) into groups of half-the-number-of-bits (N/2) and then following the same Karatsuba procedure with the segmented bits recursively. For a 16-bit multiplication, e.g, it breaks the number down to 8-bit multiplication, which is again divided into 4-bit and finally reduced to 2-bit which is the last stage for normal multiplication to be performed. At every stage we have implemented adaptive Karatsuba for the 3rd product term.

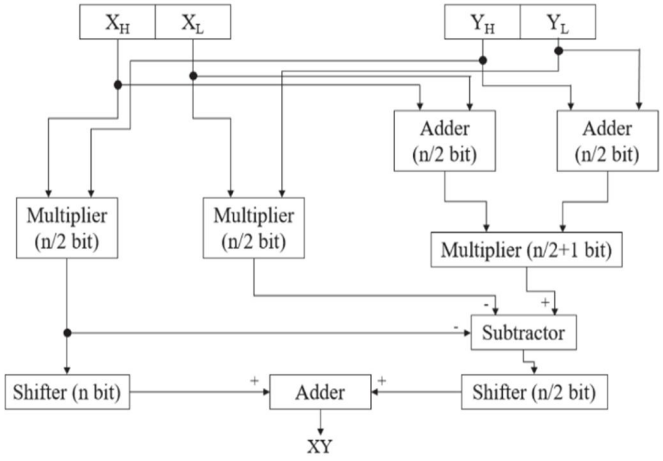


Fig. 2 Block diagram of Standard Karatsuba Multiplier

The block diagram of standard Karatsuba multiplier is shown in Figure 2. The Karatsuba algorithm section, the X and Y are decomposed into *XH XL* and *YH YL* then they are processed through separate multipliers and adders to get the result.

IV.BRENT KUNG ADDER

Brent-Kung prefix tree is a prominent structure, which has the very less fan-out and lateral wire tracks among the popular prefix adders. However, it is a complicated structure because it has the highest number of logic levels. A 16-bit Brent-Kung prefix tree structure is depicted below. Brent-Kung adder restricts the lateral fan-out of each node to unity, as in Kogge Stone adder but without the explosion of wires. The capacitive load is more due to the presence of wide span of wires. For instance, in a 16-bit adder shown, the assembly starts with prefix operators every 2 bits. The input span is 1 bit, and the output span is 2 bits. At logic level 2, the distance between each operator is 4 bits while it is 8 bits in logic level 3. At logic level 4, the only prefix operation is at the MSB with input span of 8 bits and the output span of 16 bits. At logic levels 5 through 7, the input bit spans are decremented, and they are 4, 2 and 1 bit respectively. The critical path for a 16-bit adder is from bit 0 in the pre-computation stage to bit 14 in logic stage 6. Hence, even with buffering, the Brent-Kung adders are among the slowest prefix adders. In terms of prefix operator count, for a n-bit adder, the total number of prefix operators is 2(n-1) - log2n. For n = 16, it is 26.

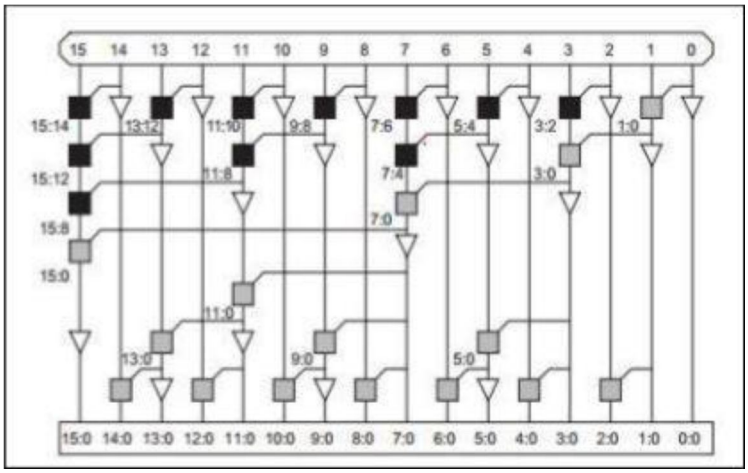


Fig. 3 16-bit Brent Kung adder flow diagram

V. LADNER FISCHER ADDER

Ladner-Fischer prefix tree structure is shown below. Sklansky's conditional sum adder can be included in this family of prefix structure. It exploits the associativity property of prefix operators extensively, but not the idempotency property, while constructing a binary tree of prefix operators. It is in some sense a basis for the other prefix tree structures. This structure has minimum logic depth but has large fan-out requirement up to n/2. The lengthiest lateral fanning wires go from a node to n/2 other nodes. Capacitive fanout loads are very large for later levels in the graph for wider input operands. In VLSI CMOS implementations, it involves increasing the drive strengths of the buffers and inverters to support larger loads. This increases the area, limits the performance by increasing the delay and burns more power due to larger drive cells. The number of logic levels of log2n is always the minimum in this scheme for an n-bit adder. Each logic level has n/2 cells. Therefore, the total number of prefix operators is nlog2n. It is indicated in Figure – 4.

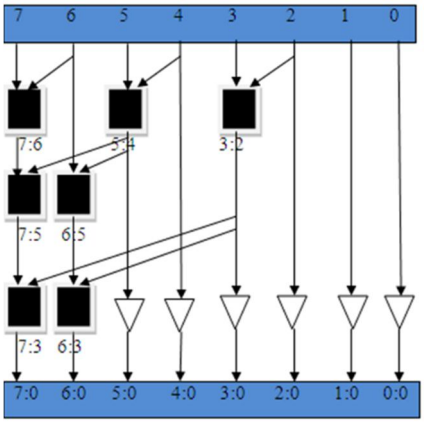


Fig. 4 Flow of Ladner Fischer adder

VI. IMPLEMENTATION OF MAC UNIT FOR DESIGN PARAMETERS

MAC unit is implemented using multiplier and adder blocks in feedback fashion. With an example of Modified booth algorithm, Multiplier takes two 6-bit numbers as inputs. First 3-bit numbers are taken for evaluating partial product. Bits are then right shifted, and all partial products are obtained. Partial products are added by left shifting the bits 2, 4 and 6 times successively. Obtained sum is product of the numbers. In the rst iteration, product is added with zero (0) and successive iterations, the new product is added with the previous product. Product is stored in a register. When multiplier product is obtained, product in the register is sent to adder unit and algorithm from above steps are repeated.

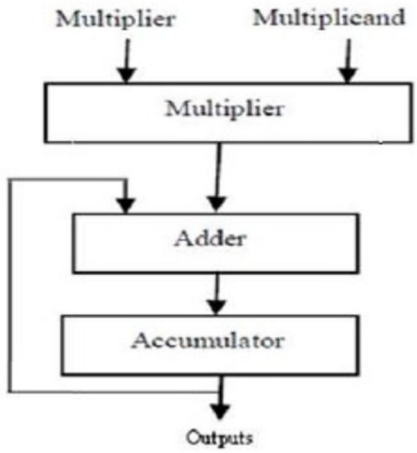


Fig. 5 Block diagram of MAC unit

Physical design is achieved after synthesis of the design. Before starting with Physical design, the required files to the PD folder should be copied. Required files include.sdc(Serial Design Constraints), netlist.v(which is generated after synthesis of design), slow.lib and fast.lib files. Then Innovus tool is invoked from command prompt. Following steps were followed for implementation:

A. According to ASIC Design flow, physical design starts with Floorplanning. In this step, H/W ratio is set to 0.8. Distance of core from boundary is set at 6.
B. VDD and ground (GND) rings are added. Metal 6 and Metal 5 are used with 15 strips each.
C. For the placement, standard cells are placed along with IO pins.
D. Clock is fed to the design.
E. Timing and SI driven NanoRoute is carried out. Routing mainly involves global routing and detailed routing.
F. After post-routing optimization, GDSII file is obtained.
G. Area, timing and power reports are obtained after each significant step to analyze the changes the design undergoes to satisfy the constraints.

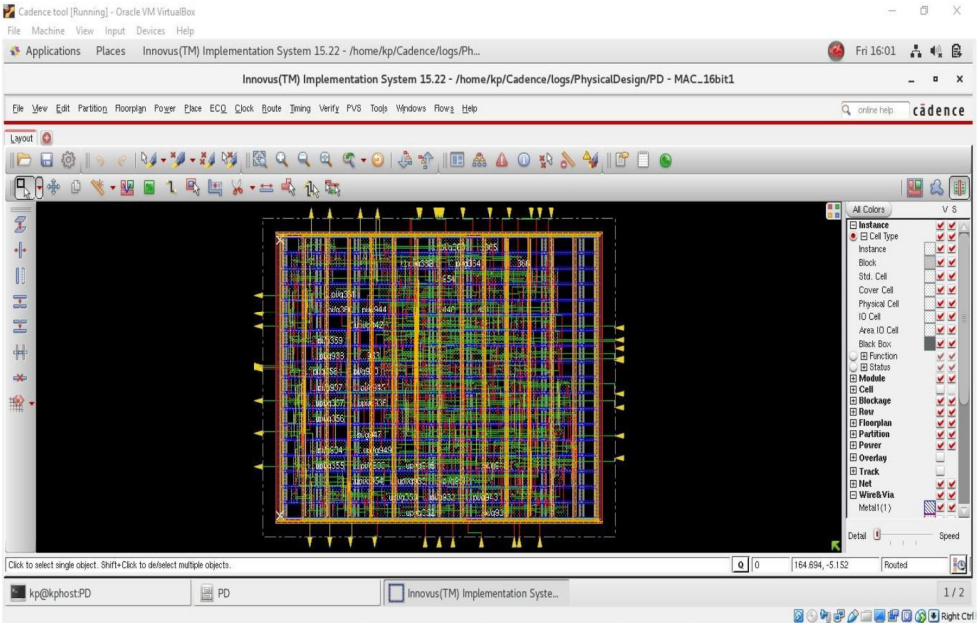


Fig. 6 Hardware implementation of MAC Unit in Innovus tool

VII. RESULTS ANALYSIS AND DISCUSSION

MAC unit was designed using multiplier, adder modules along with register acting as an accumulator block. Though the functionality of MAC unit remains same, their composition varies vividly. They can be designed to suit various applications and constraints. Hence, respective blocks have to be selected for the purpose. Out of 3 major blocks, register block serves like a memory unit and not much optimization can be performed using Verilog HDL. Nevertheless, Adder and Multiplier can be optimized using various logics from advanced designs. Such designs were implemented for major constraints like delay, power, and area.

To design a MAC unit with lesser delay, Modified booth algorithm for multiplication plays a major role. Lesser delay constitutes to higher slack. This is because of its property of lesser area overhead and faster computation. For addition, Brent Kung occupies a little overhead, but provides negligible delay to provide output. Hence, 16-bit MAC unit was designed with Modified booth multiplier and Brent Kung adder.

To design a MAC unit with lesser area, Modified booth algorithm for multiplication plays a major role. Lesser area constitutes to higher performance. This is because of its property of lesser area overhead and faster computation. For addition, Ladner Fischer occupies a little overhead, but provides negligible delay to provide output. Hence, 16-bit MAC unit was designed with Modified booth multiplier and Ladner Fischer adder.

To design a MAC unit with lesser power, Karatsuba algorithm for multiplication plays a major role. Lesser power constitutes to higher performance. This is because of its property of lesser area overhead and faster computation. For addition, Brent Kung Adder occupies a little overhead, but provides negligible delay to provide output. Hence, 16-bit MAC unit was designed with Karatsuba multiplier and Brent Kung adder.

Thus, MAC unit was implemented to extract the optimized result in terms of power, delay and area without hampering logical functionality. Table 2 displays summarized results of optimized MAC units indicating their respective power, delay and area.

Table II
Results Summary Of Optimized Mac Unit

Optimized MAC unit	Slack(ns)	Area(um^2)	Power(mW)
16bit- Brent Kung Adder- Modified Booth Multiplier	3.059	6133	2.43
16bit- Ladner Fischer Adder- Modified Booth Multiplier	0.181	129	5.76
16bit- Brent Kung Adder- Karatsuba Multiplier	1.887	6356	0.1797

VIII. CONCLUSION AND FUTURE SCOPE

An efficient multiplier block and adder block has been designed using Verilog and gate level netlist is procured. Physical design of the obtained MAC unit is performed. The ASIC flow is understood and its implementation in tools. The steps are followed in similar manner using advanced adders and multiplier that help in optimizing the design constraints like area, timing, and power. Gate level netlist was synthesized and resulting waveforms are verified with the design operating in single clock cycle.

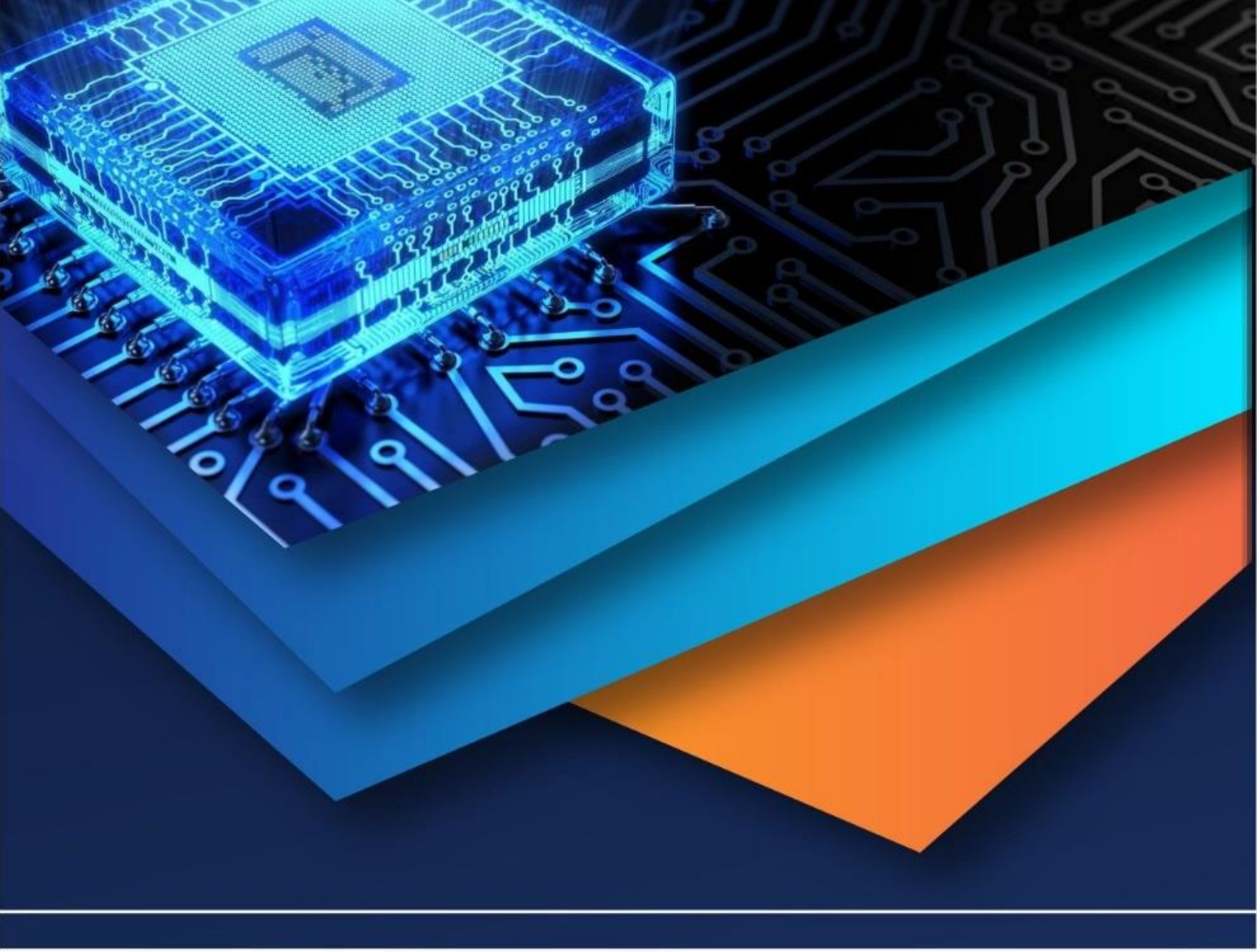
The MAC unit is designed with 180nm technology. i.e., length of the transistor is 180nm. Precision is up to 1ps. Maximum of 6 metal layers is present. A total of 15 sets of VDD and GND are present horizontally and vertically inside the boundary. The design does not take Electrostatic discharge into consideration. The work can be extended in many dimensions in the future. The technology that are market trending like 45nm or less can be employed. This is proficient in terms of area, power, and timing. Another approach is increasing the size of the design to higher number of bits to be used in DSP. Low power VLSI techniques can be also used in algorithmic level.

IX. ACKNOWLEDGEMENT

The work was carried out under guidance of Dr. H. V. Ravish Aradhya. I would acknowledge my guide, staff, and friends of my institution to have helped me complete the work successfully.

REFERENCES

[1] Aradhya H.V.R and Goudar S, “Development anSd Analysis of Parameters to Evaluate Design Partitioning of SoC,” Proceedings of the 2 nd International Conference on Inventive Research in Computing Applications, ICIRCA-2020, 2020, pp. 416–421.
[2] Aradhya H.V.R, Aktab M.L.U, Saberi F, “Development of a Random Test Generator for Multi-Core Processor Design Verification,” Proceedings of the lrd International Conference on Electronics and Communication and Aerospace Technology, ICECA-2019, 2019, pp. 1200–1204.
[3] H. Zhang, D. Chen and S. Ko, "New Flexible Multiple-Precision Multiply-Accumulate Unit for Deep Neural Network Training and Inference," in IEEE Transactions on Computers, vol. 69, no. 1, pp. 26-38, 1 Jan. 2020
[4] K. Lilly, S. Nagaraj, B. Manvitha and K. Lekhya, "Analysis of 32-Bit Multiply and Accumulate unit (MAC) using Vedic Multiplier," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 2020
[5] R. Pandimeena, A. Annapoorani, T. D. Kaviya, A. Rajalakshimi and R. Dhanagopal, "Design and Comparison of Power, Area and Delay of 32-bit Reversible MAC unit," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2020
[6] Akella Srinivasa Krishna Vamsi and Ramesh S R, “An Efficient Design of 16 Bit MAC Unit using Vedic Mathematics Conference on Computing, Communication and Networking Technologies (ICCCNT), Bangalore, 2018
[7] R. Balakumaran and E. Prabhu, "Design of high speed multiplier using modified booth Algorithm with hybrid carry look-ahead adder", in 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT),Nagercoil, 2016
[8] Young-Ho Seo and Dong-Wook Kim, "New VLSI Architecture of Parallel Multiplier-Accumulator Based on Radix-2 Modified Booth Algorithm," IEEE Transactions on very large scale integration (vlsi) systems, vol. 18, no. 2,february 2010



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY