

Efficient Implementation of Fixed-Point MAC and Multimode MAC Blocks Based on Vedic Mathematic

Fatima Tariq Hussein*, Fatemah K. AL-Assfor

Computer Engineering Department, College of Engineering, University of Basrah, Basrah, Iraq

Correspondance

*Fatima Tariq Hussein

Department of Computer Engineering, University of Basrah, Basrah, Iraq

Email: pgs.fatima.tariq@uobasrah.edu.iq

Abstract

Recently, the need for high speed multiply-accumulate (MAC) operations is crucial in numerous systems like 5G, deep learning, in addition to many digital signal processing (DSP) applications. This work offers an improved MAC (I-MAC) block of different bit-size based on Vedic Mathematic and employing a hybrid adder consists of an enhanced Brent-Kung with a carry-select adder (HBK-CSLA) to achieve the sum of products for the MAC. The work is then, developed to design a new multimode fixed-point (FX-Pt) MAC block by exploiting the proposed design of the I-MAC architecture. The proposed multimode MAC block supports three modes of operation; single 64-bit MAC operation, dual 32-bit multiplication with 32-bit single addition, and single 32-bit MAC operation. The design has utilized an adjusted architecture for the Vedic-multiplier (Adjusted-VM), a 64-bit HBK-CSLA, and a control circuit to select the desired mode of operation. The performance of the multi-mode MAC is then optimized by exploiting pipelining concept. The proposed architectures are synthesized in various FPGA families utilizing VHDL language in Xilinx ISE14.7 tool. The performance results have exposed that the proposed 64-bit I-MAC block have attained observable lessen 9.767% in delay and area usage of 47.49% compared with the most existing MAC block designs.

Keywords

Improved Multiply Accumulator (I-MAC), Adjusted-Vedic Multiplier (Adjusted-VM), Hybrid Brent Kung Carry-Select Adder (EBK-CSLA), Multimode MAC, FPGA.

I. INTRODUCTION

Digital signal processing (DSP) and deep learning (DL) systems, like convolutional neural networks (CNN)s necessitate high levels of precision and timing accuracy to perform their data stream processing tasks effectively [1–3]. MAC block is a basic component in these systems which used to perform convolution operation and filtering, and inference operations in deep learning [4, 5]. MAC block architecture is basically composed of an $(n \times n)$ multiplier and a $2n$ -bit accumulated adder, and it can be operated on both fixed-point (FX-Pt) and floating-point (FL-Pt) numbers. To meet the conditions of cutting-edge DSP and DL applications, two approaches can be considered to improve the speed of MAC unit, first; improve the multiplier's speed and its area, second, employing high speed adders to generate the final product and/or accumulate

the generated product with the pervious products [6, 7].

Numerous multiplier and adder designs have been considered to achieve high MAC performance, one of the multipliers is the Vedic-multiplier (VM) [8–12]. Consequently, employing an improved version of VM or fast adder design will enhance the MAC block performance which in turns positively enhance the DSP architecture in terms of the speed and area resources. Numerous designs and algorithms have been proposed for designing MAC unit. In 2016, S. Nighat et al. [13] had designed 8 and 16-bit MAC blocks employing adapted Wallace-tree Multiplier with and without carry save additions and a carry-increment adder (CIA) together with the carry-select adder for the accumulated addition. Their design had incurred high delay and area due to the carry propagation delay of CIA. In 2017, M. Yuvaraj et al [6] had been offered



This is an open-access article under the terms of the Creative Commons Attribution License, which permits use, distribution, and reproduction in any medium, provided the original work is properly cited.
©2025 The Authors.

Published by Iraqi Journal for Electrical and Electronic Engineering | College of Engineering, University of Basrah.

absolute Vedic based MAC with different bit sizes. Their design had included an integrated VM structure comprised of all multiplier blocks that based on Vedic algorithms (namely; the six Vedic algorithms), and a specialized control circuit to determine which multiplier block be utilized based on the input operands type in order to produce better results. Their design had achieved lower delay in comparison with the traditional MAC but consumed high area resources due to the use a set of six VM blocks in addition to the control circuit.

Authors in [14] introduced different approach for designing a 4-bit MAC block utilizing a Sirisha-Purushottam-Tilak (SPT) reversible gates but they have used the parallel adder to accumulate the products which has led to high delay and area. Their designs provide better performance factors in terms of gate count and constant inputs. Authors in [15, 16] had proposed MAC blocks employing diverse types of multipliers, such as Booth, Wallace Tree (WT), Array, and VM, and different types of adders like Ripple-Carry Adder (RC-A), Carry-Save Adder (CS-A), and Kogge Stone Adder (KS-A) to examine their effects on the MAC performance. They found that the MAC circuit designed by the WT multiplier and CR-A with CS-A had introduced less delay than other MAC designs. Ramadevi et al in 2022 [17], have presented high-speed MAC circuits through employing UT-Sutra and carry select adder (CSLA). In [18], authors analyze the effect of utilizing a VM that includes three CS-As on the performance of 64-bit MAC unit. According to their analysis, the MAC block had introduced less energy and occupied less space than traditional MAC unit. Then, a 2-stage pipelined 64-bit MAC block is developed by Bhadra et al. [19] utilizing VM and two different adders to perform sum of products: the RCA with CLA and RCA with CSLA. However, their designs incur high delay and area resources utilization due to the use of the RCAs. In [20], authors have presented two designs for a 64-bit Vedic based MAC block utilizing first: DKJ-adders with CR-As, and then, utilizing DKJ-adders with CSLAs to perform partial product reduction, final product generation of the VM as well as the accumulated addition of the products for the MAC block. Both designs have given better performance in terms of power consumption, delay, and area compared to the traditional 64-bit MAC. In 2023, Bharathi has designed a diverse bit-width MAC block that relied on the offset binary-coded (OBC) and distributed arithmetic (DA) techniques and has examined the effect of using diverse adders on 64-bit MAC block performance. Nevertheless, the area of his designs is low for high bit-width when utilizing the OBC, but the delay of the design is very high [21].

In this article, an adjusted structure for the VM is offered and called Adjusted-VM. The offered multiplier depends on the utilization of a hybrid form adder to speed-up the generation of the product result step in the multiplier design. The pro-

posed hybrid-adder used is comprised of an enhanced design of the Brent-Kung adder (BK-Adder) and a carry-select adder (CSL-Adder), it is called "HBK-CSLA" H stands for hybrid. Then, the article has developed an improved design of a FX-Pt MAC named I-MAC block with different bit widths (namely: 8, 16, 32, and 64-bit). The proposed I-MAC exploits the Adjusted-VM to perform multiplication and the proposed HBK-CSLA to be perform the accumulated addition of the products for the proposed I-MAC, in order to boost the performance of MAC block. Further, a new architecture for a multimode FX-Pt MAC block is developed from the I-MAC to make the MAC block flexible in selecting the input-data precision that operated on in order to exploit the beneficial of having the same hardware design for supporting multi-precision input data to operate on. In other words, achieving maximum hardware usage all the time. Finally, to optimize the performance of the multimode MAC block, the design is pipelined. All of the proposed architectures are VHDL coded, simulated by Xilinx 14.7 and implemented utilizing numerous FPGA families. This article is organized as follows: Sections II introduces the basic (2*2)-bit binary VM and the developed (4*4) and (32*32) Adjusted-VMs. Section III focused on designing different bit-width of an improved architecture of the MAC block (called I-MAC) based on the Adjusted-VM. Section IV introduces a new MAC block architecture with multimode of precisions (without/with pipeline). Section V, summarizes the results and performance evaluation of the I-MAC and multimode MAC blocks that process data in FX-Pt format. Section VII concludes the work and its results.

II. ADJUSTED VEDIC MULTIPLIER (ADJUSTED-VM)

Vedic multiplier (VM) depends on an algorithm called "vertical and crosswise" or Urdhva Tiryakbhyam (UT)-Sutra [22, 23]. The most important feature of the VM is its ability to generate only four partial product and requires less hardware than other multipliers. Thus, it can perform multiplication at high speed and consume lower area [24, 25]. Generally, multiplication of two operands is accomplished by three steps: Step-1: partial-products (PPr)s generation .

Step-2: PPrs reduction using a set of adders, or a single adder such as a (3:2) carry-save adder (CS-A) to produce the product result as two vectors form (namely; sum (S) and carry (C) vectors) which is called redundant form.

Step-3: producing the final product (i.e. multiplication result) through utilizing a fast adder to add the S and C vectors generated in step 2.

A (2*2) VM encompasses four AND-gates and two modules of half-adder (HA) [26–28]. A (2*2) VM accepts two (2-bit) operands ($A \equiv A[1:0]$ and $B \equiv B[1:0]$) to provide 4-bit prod-

uct result (namely $P[3:0]$) according to the UT-Sutra approach in Vedic mathematic [29–31]

$$P[0] = A[0] * B[0] \quad (1a)$$

$$C_{01}P[1] = A[1] * B[0] + A[0] * B[1] \quad (1b)$$

$$P[3] = A[3] * B[3] + C_{01} \quad (1c)$$

where $P[i]$: is the product i^{th} bit.

In this article, an $(4*4)$ Adjusted-VM circuit is designed first utilizing $(2*2)$ VM with a 4-bit CS-A, and a 3-bit enhanced HBK-CSLA [22] to perform the midst-(second) stage additions, as shown in Fig. 1. A 2-bit increment-by-1 (Ib-1) converter is employed to update the most significant bits (MSB)s of multiplication.

The proposed $(4*4)$ Adjusted-VM can be easily expanded to build an $(8*8)$, a $(16*16)$, and a $(32*32)$ Adjusted-VMs. Fig. 2 demonstrates the $(32*32)$ Adjusted-VM module.

It comprises of four $(16*16)$ Adjusted-VM to generate four PPrs, each of 16-bit. Then, a 32-bit CS-A is employed to reduce the PPrs from four to two vectors: sum and carry vectors ($S[31:0]$ and $C[31:0]$), followed by a 32-bit HBK-CSLA to generate the midst 32-bit of the final product. The most significant bits of the product are generated by two (8-bit) HBK-CSLA.

III. PROPOSED FX-PT MAC BLOCK ARCHITECTURE

This section presents an $2n$ -bit fixed-point efficient MAC block based on utilizing an $(n*n)$ Adjusted-VM, with numerous sizes of n .

To improve the speed of the MAC more, the HBK-CSLA is again used to be the MAC block adder which is used to add the generated product result from the multiplier (see Fig. 3).

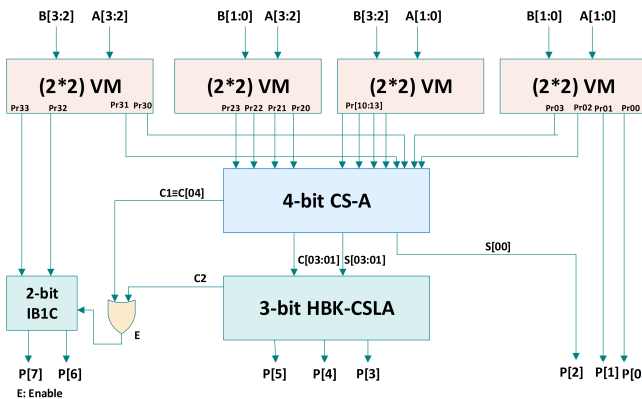
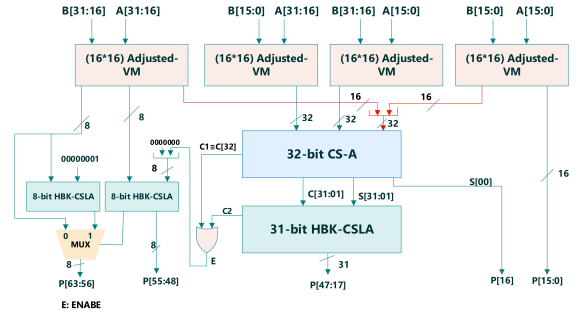
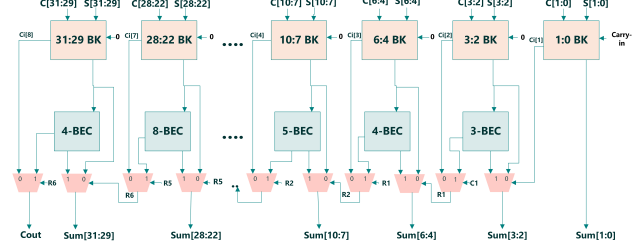


Fig. 1. Proposed $(4*4)$ Adjusted-VM.



(a) Proposed $(32*32)$ Adjusted-VM



(b) Proposed 32-bit HBK-CSLA

Fig. 2. Architecture of a $(32*32)$ Adjusted-VM and its HBK-CSLA.

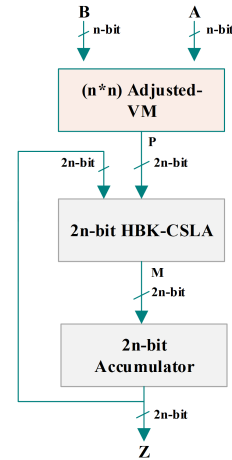


Fig. 3. Block diagram of $2n$ -bit I-MAC module.

Fig. 4 demonstrates the proposed 64-bit FX-Pt I-MAC block. It comprises of $(32*32)$ -bit Adjusted-VM, a 64-bit HBK-CSLA, and accumulator.

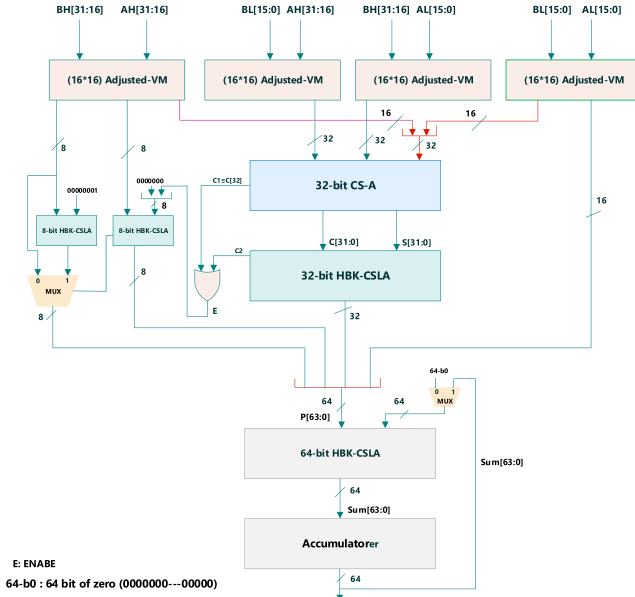


Fig. 4. The Proposed 64-bit FX-Pt I- MAC Block.

IV. PROPOSED MULTI-MODE MAC ARCHITECTURE

This section introduces a new design for a multimode MAC block that based on Vedic mathematic. To best of knowledge, there is no such elastic multimode precision MAC block design that based on Vedic mathematic, manifesting in the literature. The proposed multi-mode MAC unit supports three modes of operation, namely; single 64-bit MAC, sum of (16*16) multiplication and a 32-bit addend, or single 32-bit MAC (as can see in Table I).

TABLE I.
MODES OF OPERATION OF THE MULTIMODE
FX-PT MAC BLOCK

Selection mode $S_1 S_0$	Operation mode for multimode MAC
00	Single 64-bit MAC (Single (32*32)) multiplication with 64-bit accumulation)
01 or 10	Accumulation the sum of dual (16*16)-bit multiplications
11	Single 32-bit MAC (Single (16*16) multiplication with 32-bit accumulation)

The input operands of multimode MAC block consist of either single-precision (32-bit) or two sets of half-precision (16-bit) input operands (see Fig. 5).

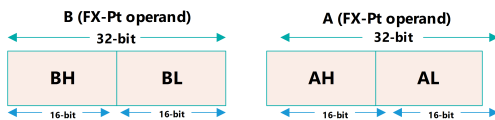


Fig. 5. 64-bit FX-Pt input operands.

The multimode MAC is developed from the proposed 64-bit I-MAC block.

Fig. 6 demonstrates the proposed multimode-precision FX-Pt MAC block.

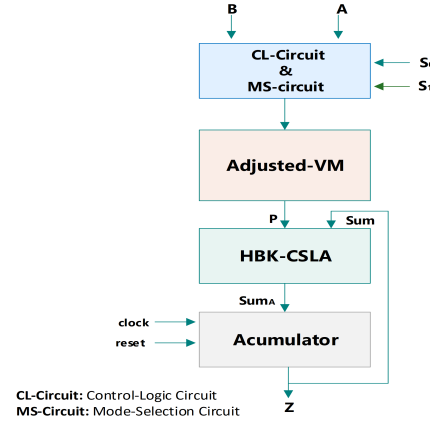


Fig. 6. Proposed multimode-precision FX-Pt MAC block.

As can see in Fig. 7, it comprises of four (16*16) Adjusted-VM, control-logic circuit (CL-Circuit) to select the precision of the input operands, a 32-bit CS-A, together with diverse sizes of HBK-CSLA to accumulate the results. The multiplier input operands are controlled by tri-state buffers (TS-Buffers). Two selection signals: S_0 and S_1 are utilized to select the mode of operation, and two enable signals are used to control the MAC operation during the three operated modes. These signals are:

$$E1 = \overline{S_0} \cdot \overline{S_1} \quad (2a)$$

$$E2 = \overline{S_0} \cdot S_1 \quad (2b)$$

The enabled signals E1 and E2 enable the multiplier inputs only when a new multiplication operation is taking place, this leads to avoid any undesirable action to keep energy. The multimode MAC starts its operation when the signal “start” is active high and one of the modes of operation is selected. The three modes are operated as follows:

In mode 1, signals $E_1 E_2 = 00$; in this case the two 32-bit inputs A and B (namely; $A \equiv AH:AL$ and $B \equiv BH:BL$) are passed to the four (16*16) Adjusted-VM modules to perform (32*32)-bit to generate four (32-bit) PPrs. Then, a 32-bit CS-A is used to add the midst-stage partial-products, followed by a 32-bit HBK-CSLA to find out the final product of the midst stage. The signal E1 is used to control these adders as depicted in Fig. 7. The MSBs of both: the CS-A and the

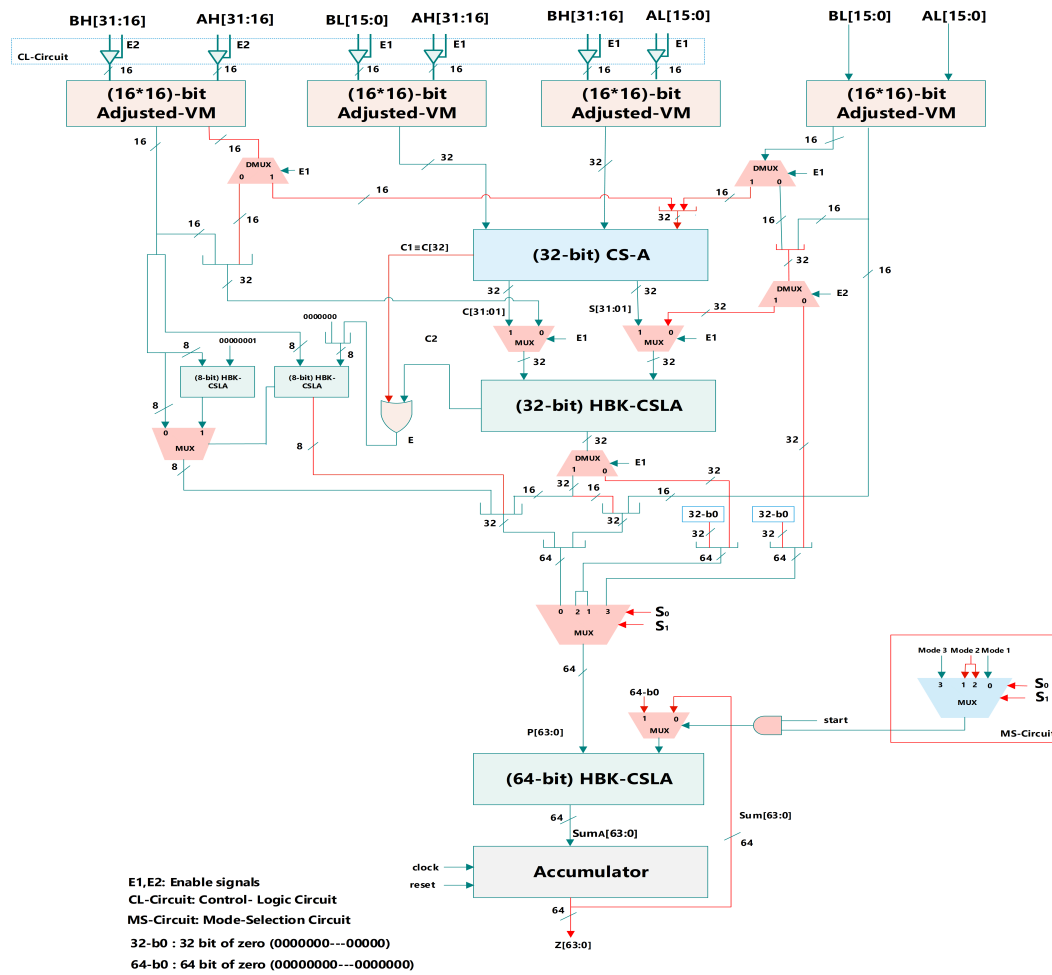


Fig. 7. Proposed multimode-precision FX-Pt MAC block.

HBK-CSLA (namely; C1 and C2) are OR-ed together and then fed to a multiplexer which is used to control the first 8-bit HBK-CSLA block. The 64-bit product is then accumulated using 64-bit HBK-CSLA.

In mode 2, the enabled signal E1 is disabled to prevent the two middle (16*16) Adjusted-VMs performing multiplication. Consequently, only the first and the fourth (16*16) Adjusted-VMs have fed with input data and thus, performing dual (16*16) bit multiplication ($AL*BL$) and ($AH*BH$), respectively. In this mode, the 32-bit CS-A and the 32-bit HBK-CSLA are disabled as well. This mode performs the

sum of dual (16*16)-bit products with a 32-bit addend. In mode 3, the signals E1 and E2 are disabled to make the first (16*16) Adjusted-VM accepts the 16-bits operands AL and BL, respectively and accomplish single (16*16)-bit multiplication. The generated 32-bit product is then moved to perform a single 32-bit MAC operation. In this mode, the most significant 32-bit of the accumulation adder are not used. The multimode FX-Pt MAC block design is then, converted to a 2-stage pipelined structure to carryout higher speed (less delay), as explained in Fig. 8.

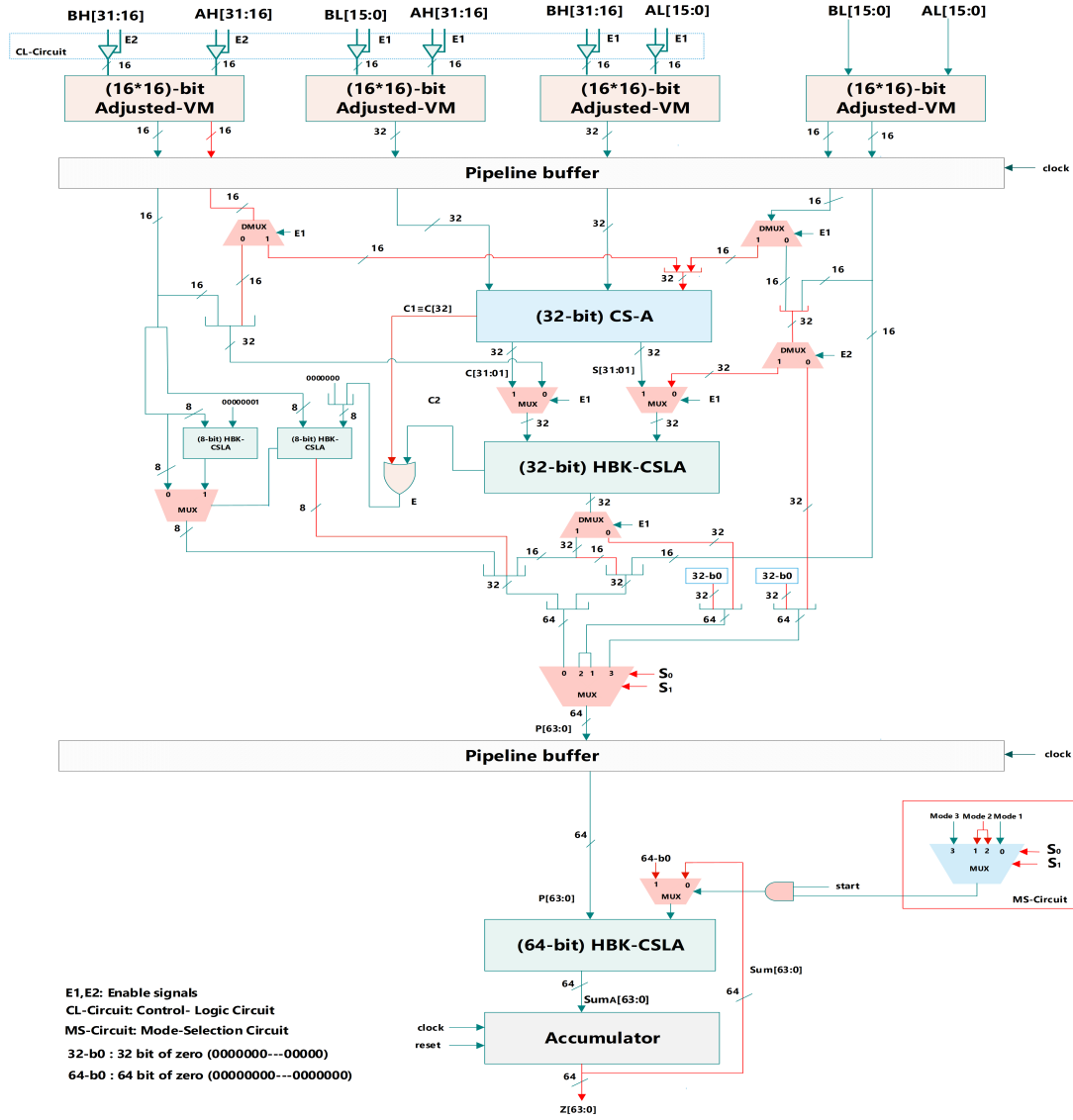


Fig. 8. Pipelined multimode-precision FX-Pt MAC block.

V. RESULTS OF SIMULATION AND THE EVALUATION

The proposed Adjusted-VM, FX-Pt I-MAC with different bit sizes, and multimode MAC blocks are VHDL coded and executed by Xilinx tool.

A. Results of Simulation

The FPGA implementation results of the Adjusted-VM, I-MAC, and multimode MAC blocks in terms of area usage (namely; in Lookup table (LUT)) and the MAC delay (nsec) are evaluated in Xilinx ISE-14.7 tool utilizing six FPGA fami-

lies: Spartan-3E, Spartan-6, Virtex-5, Virtex-6, Virtex-7, and Zynq, as indicated in Tables II-IV.

TABLE II.

PERFORMANCE OF NUMEROUS BIT-SIZE ADJUSTED-VM MODULE

Metric	FPGA Family	Adjusted-VM (bit)			
		(4*4)	(8*8)	(16*16)	(32*32)
Delay(nsec)	Spartan-3E	11.590	19.459	31.830	52.193
	Spartan-6	7.891	11.805	17.225	26.253
	Artix-7	3.693	6.152	10.633	17.799
	Virtex-5	6.793	9.346	14.601	22.368
	Virtex-6	3.116	5.195	9.336	15.597
	Virtex-7	2.902	4.918	8.669	14.447
	Zynq	2.902	4.918	8.669	14.447
Area (LUT No.)	Spartan-3E	24	156	717	3151
	Spartan-6	23	118	533	2296
	Artix-7	22	118	522	2296
	Virtex-5	22	118	528	2300
	Virtex-6	22	118	522	2296
	Virtex-7	22	118	522	2296
	Zynq	22	118	522	2296

TABLE III.

DELAY AND AREA OCCUPANCY OF NUMEROUS BIT-SIZE MAC BLOCK

Metric	FPGA Family	Unpipelined FX-P I-MAC			
		8-bit	16-bit	32-bit	64-bit
Delay(nsec)	Spartan-3E	14.673	25.283	34.654	55.416
	Spartan-6	11.371	16.866	21.063	35.553
	Virtex-5	7.256	10.600	12.363	21.757
	Virtex-6	4.262	7.028	10.168	16.838
	Virtex-7	3.979	6.535	9.469	16.035
	Zynq	3.979	6.535	9.469	16.035
Area (LUT No.)	Spartan-3E	57	229	805	3694
	Spartan-6	40	182	666	2610
	Virtex-5	39	176	658	2570
	Virtex-6	36	174	646	2552
	Virtex-7	36	173	646	2548
	Zynq	36	173	646	2548

TABLE IV.

DELAY AND AREA OCCUPANCY OF FX-PT MULTIMODE MAC BLOCK

Metric	FPGA Family	Multimode MAC Block	
		Unpipelined-MAC	Pipelined-MAC
Delay(nsec)	Spartan-3E	56.546	33.351
	Spartan-6	37.470	19.854
	Virtex-5	24.731	14.047
	Virtex-6	17.402	9.829
	Virtex-7	16.131	9.04
	Zynq	16.131	9.04
Area (LUT No.)	Spartan-3E	3856	2921
	Spartan-6	2747	2243
	Virtex-5	2718	2152
	Virtex-6	2702	2143
	Virtex-7	2701	2143
	Zynq	2701	2143

All proposed design have been examined in terms of area and delay using various FPGA families, as illustrated in Fig. 9-14.

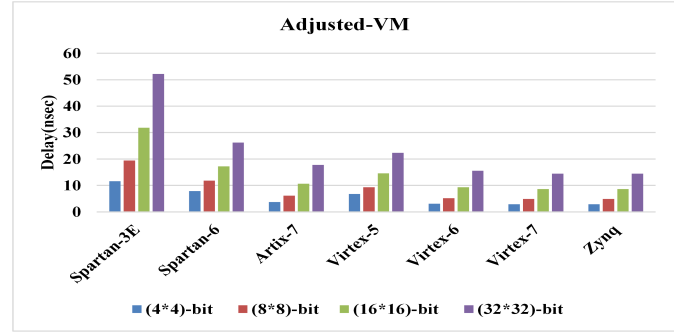


Fig. 9. Delay analysis of diverse bit-size Adjusted-VM circuits.

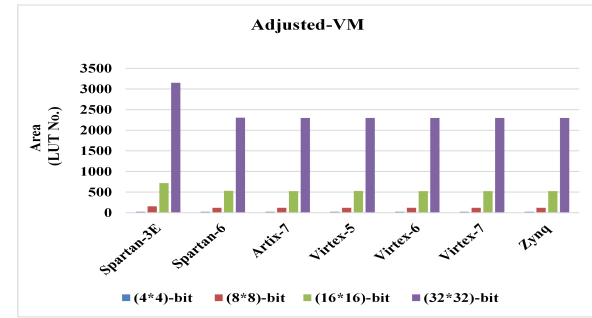


Fig. 10. Area analysis of diverse bit-size Adjusted-VM circuits.

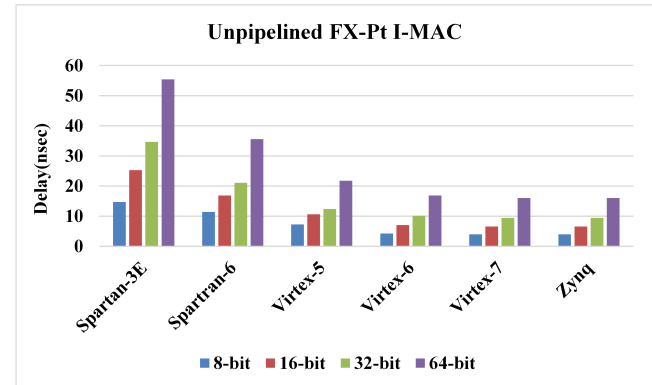


Fig. 11. Delay analysis of several bit-size for unpipelined I-MAC block.

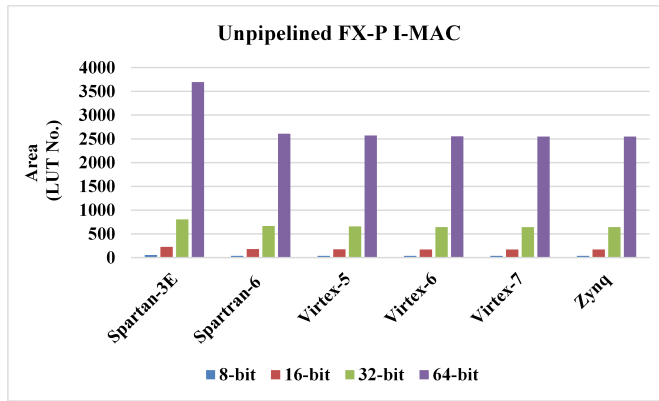


Fig. 12. Area analysis of several bit-size for unpipelined I-MAC block.

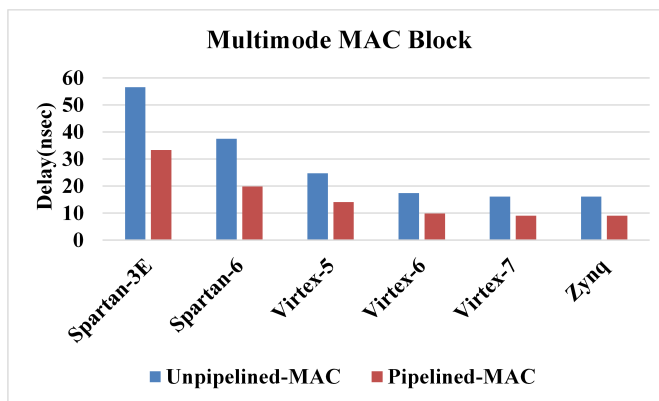


Fig. 13. Delay analysis for multimode FX-Pt MAC block.

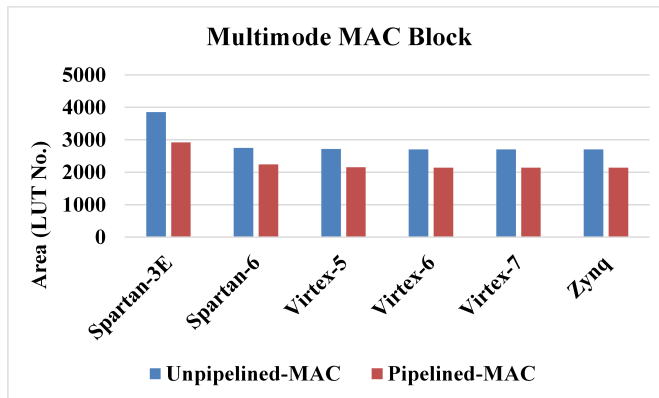


Fig. 14. Area analysis for multimode FX-Pt MAC block.

The I-MAC is simulated to functionally verifying its ability in carrying out the sum of product for the FX-Pt operands. Figure 15 states the 64-bit I-MAC simulation result for a few of input operands (five sets of input operands (A and B) represented in decimal) after setting the start signal to 1 for

enabling the MAC block to carry out the sum of product for the input data:

$$Sum = Sum + A * B \equiv \sum A * B \quad (3)$$

Operation performed is: $Sum = 0 + (20 * 35) + (325 * 77) + (2345 * 1111) + (10000 * 20000) + (98 * 2340) = 202862040$.

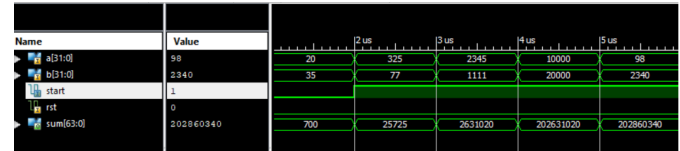


Fig. 15. Input/output waveforms of the 64-bit I-MAC block.

The proposed multimode FX-Pt MAC is simulated as well to prove their functionality in performing the sum of product for some FX-Pt operands (the operands are represented in decimal) to verify the corresponding outputs.

For example, to select mode 1 of operation, the selection lines $S_1 S_0 \equiv S[1:0]$ are set to 00, and the start signal is set to 1 to enable the multimode MAC block to perform the sum of product for the following five sets of input operands (A and B) as stated in Fig. 16:

$Sum = 0 + 100 * 33 + 250 * 25 + 1111 * 98 + 233 * 21 + 100 * 5000 = 623321$.

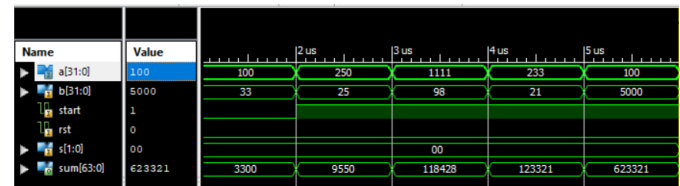


Fig. 16. Input/output waveforms of multimode Vedic based MAC block.

Fig. 17 intimates the register-transfer-level (RTL)- diagrammatic of the synthesized 64-bit MAC block.

It can be perceived that the 64-bit MAC design has exploit the following number of hardware modules: four modules of (16*16)-bit Adjusted-VM, a 32-bit CS-A, four HBK-CSLAs of numerous bit sizes: one (32-bit), two (8-bit) and one (64-bit) HBK-CSLAs, an OR-gate, a 64-bit accumulator, and a (2:1) MUX.

The internal design scheme in RTL for the multimode MAC block has exploited 128-bit TS-buffer, four (16*16)-bit Adjusted-VM, four HBK-CSLAs of size 8 (two modules), 32, and 64-bit, two OR gates, a couple (4-to-1) multiplexers, five (2-to-1) multiplexers, a mode-selection logic circuit (Fig. 18).

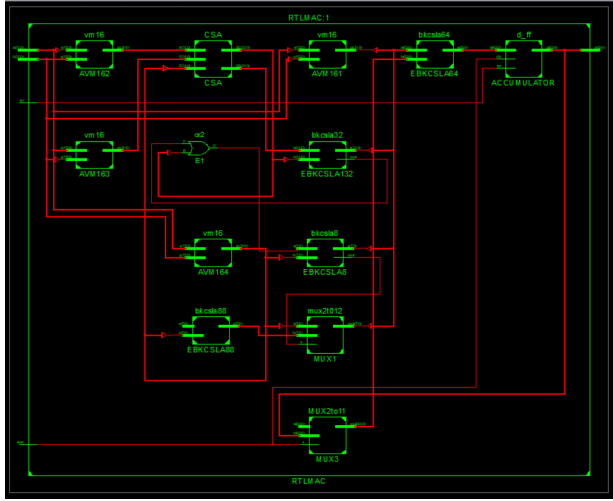


Fig. 17. The RTL of the proposed 64-bit FX-Pt I-MAC block.

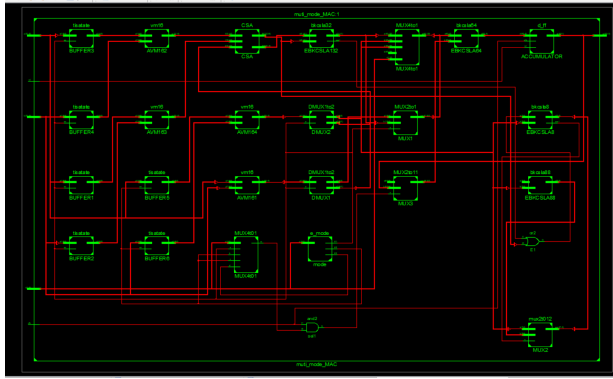


Fig. 18. The RTL diagrammatic of the multimode FX-Pt MAC block based on Vedic mathematic.

B. Evaluation

Table V lists the delay and area of the proposed Adjusted-VM module implemented with the preceding multipliers in literature. It can be exposed that the proposed Adjusted-VM achieves the best speed up and area usage with respect to other designs in literature. As an example, the Adjusted-VM achieves highest speedup of 44.18% and lowest area usage of 29.459% than the (16*16)-bit multiplier circuit introduced in [10] for the same FPGA family (Virtex-7).

The comparison in Table VI plainly shows that the offered I-MAC architecture is speedier (less delay) and occupies less area than the other existent proposed designs for the same FPGA families. For example, the 32-bit and 64-bit I-MAC blocks have attained lower delay and area usage of 9.767% and 47.49%, respectively than the designs proposed in [31] when using the FPGA Zynq family. Also, the (64-bit) bit I-MAC introduce lower delay of 75.117% and 58.835%, respectively

when compared to the designs in [21] when using Spartan 3E FPGA family. However, their designs had exploited lower area than the I-MAC, since they had not designed a multiplier to perform multiplication and they had utilized LUTs for the PPrs.

As illustrated in Table IV, the proposed I-MAC yields lower execution time (delay) and attains noticeable reduction in area usage compared with the most existent MAC blocks. The delay reduction comes from the utilization of the efficient Adjusted-VM along with a HBK-CSLA which is used as the accumulated adder for the MAC block. Thus, the proposed FX-Pt MAC can be employed to satisfy the Applications demands of cutting-edge systems like the Deep learning and DSP systems. For the case of multimode precisions FX-Pt MAC block with/without pipelining there is no existent design to compare with Due to the difference in the tools used for measurement and evaluation.

TABLE V.
DELAY AND AREA USAGE COMPARISON FOR THE
MULTIPLIERS WITH VARIOUS BIT-WIDTH

Ref.	FPGA family	Multiplier size	Delay(nsec)	Area Utilization (LUT No.)
[12]	Spartan-3E	(16*16)	30	-
[26]	Spartan-3E	(4*4)	15.42	30
		(8*8)	28.73	153
		(16*16)	10.43	24
[8]	Spartan-6 (Design 1)	(8*8)	18.46	125
		(4*4)	8.936	34
		(8*8)	13.594	188
		(16*16)	21.94	858
		(32*32)	34.778	3680
	Spartan-6 (Design 2)	(4*4)	8.417	33
		(8*8)	16.015	173
		(16*16)	25.044	776
		(32*32)	43.183	3234
		(16*16)	32.175	656
Proposed	Spartan-3E	(16*16)	24.903	884
		(8*8)	17.776	628
		(4*4)	15.531	740
		(16*16)	11.590	24
		(8*8)	19.459	156
	Spartan-6	(16*16)	31.830	717
		(32*32)	52.193	3151
		(4*4)	7.891	23
		(8*8)	11.805	118
		(16*16)	17.225	533
	Artix-7	(32*32)	26.253	2296
		(4*4)	3.693	22
		(8*8)	6.152	118
		(16*16)	10.633	533
		(32*32)	17.799	2296
	Virtex-5	(4*4)	6.793	22
		(8*8)	9.346	118
		(16*16)	14.601	538
		(32*32)	22.368	2300
	Virtex-6	(4*4)	3.116	22
		(8*8)	5.195	118
		(16*16)	9.336	533
		(32*32)	15.597	2296
	Virtex-7	(4*4)	2.902	22
		(8*8)	4.918	118
		(16*16)	8.669	533
		(32*32)	14.447	2296
	Zynq	(4*4)	2.902	22
		(8*8)	4.918	118
		(16*16)	8.669	533
		(32*32)	14.447	2296

TABLE VI.
PERFORMANCE COMPARISON OF NEMOURS BIT-WIDTH
I-MAC BLOCK.

Ref.	FPGA family	MAC size	Delay(nsec)	Area Utilization (LUT No.)
[31]	Zynq	16-bit	9.385	-
		32-bit	10.494	-
		64-bit	30.54	-
[1]	NR	32-bit	82.911	3637
[16]	NR	32-bit	17.199	694
[21]	Spartan-3E (Design1)	64-bit	222.712	612
	Spartan-3E (Design2)	64-bit	134.621	185*
Proposed	Spartan-3E	8-bit	14.673	57
		16-bit	25.283	229
		32-bit	34.654	805
		64-bit	55.416	3694
	Zynq	8-bit	3.979	36
		16-bit	6.535	173
		32-bit	9.469	646
		64-bit	16.035	2548

VI. CONCLUSION

In this article, an improved MAC (I-MAC) block with numerous bit-width is introduced to perform FX-Pt calculations which is important to accomplish the sum of products, convolution, and inference operations. The offered I-MAC was designed based on utilizing an Adjusted-VM and a hybrid adder design that involves a Brent-Kung adder and a carry-selected adder called HBK-CSLA. Then, a new architecture for a multimode MAC block was developed to make the operation of MAC block more flexible in selecting the input data precisions. The multimode MAC is then pipelined to optimize its work and reduce the delay. The proposed (16*16) Adjusted-VM achieved a decline of 38.89% and 99.738% for the delay and FPGA area usage, respectively than the most current multiplier design. Also, the proposed (64-bit) I-MAC block offers high-speed (i.e. lessen delay), it attains considerable reduction in delay of 75.117% and 58.835%, over the most existent designs published in 2023. Since the multimode was developed and designed from the efficient I-MAC, thus it can be assuring that the multimode MAC is efficient as well.

CONFLICT OF INTEREST

The author have no conflict of relevant interest to this article.

REFERENCES

- [1] K. Lilly, S. Nagaraj, B. Manvitha, and K. Lekhya, "Analysis of 32-bit multiply and accumulate unit (mac) using vedic multiplier," in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pp. 1–4, IEEE, 2020.
- [2] T. Shanmugaraja and N. Kathikeyan, "Power effective multiply accumulation configuration for low power applications using modified parallel prefix adders," in *2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC)*, pp. 1–7, IEEE, 2023.
- [3] P. C. Mule and S. Mande, "Design and performance analysis of fpga based dpu using mac unit," in *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*, pp. 1–6, IEEE, 2023.
- [4] P. Chintapoodi, R. Thirumuru, A. B. Mohammad, G. Sudhagar, *et al.*, "Efficient re-configurable multiply and accumulate unit for convolutional neural network," in *2022 International Conference on Recent Trends in Microelectronics, Automation, Computing and Communications Systems (ICMACC)*, pp. 77–81, IEEE, 2022.
- [5] H. O. Ahmed, M. Ghoneima, and M. Dessouky, "Concurrent mac unit design using vhdl for deep learning networks on fpga," in *2018 IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE)*, pp. 31–36, IEEE, 2018.
- [6] M. Yuvaraj, B. J. Kailath, and N. Bhaskhar, "Design of optimized mac unit using integrated vedic multiplier," in *2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*, pp. 1–6, IEEE, 2017.
- [7] K. J. Ramesh *et al.*, "A review: Multiply and accumulate architectures for digital signal processing and digital image processing," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 12, pp. 3797–3804, 2021.
- [8] H. Rakesh and G. Sunitha, "Design and implementation of novel 32-bit mac unit for dsp applications," in *2020 International Conference for Emerging Technology (INCET)*, pp. 1–6, IEEE, 2020.
- [9] A. Chhabra and J. Dhanoa, "A design approach for mac unit using vedic multiplier," in *2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, pp. 1–5, IEEE, 2020.
- [10] P. Sairam, K. Manikumar, Y. S. Reddy, B. U. Narayana, and K. Gowreesrinivas, "Fpga implementation of area efficient 16-bit vedic multiplier using higher order compressors," in *2023 IEEE Devices for Integrated Circuit (DevIC)*, pp. 404–407, IEEE, 2023.
- [11] A. Verma, A. Khan, and S. Wairya, "Design and analysis of efficient vedic multiplier for fast computing applications," *International Journal of Computing and Digital Systems*, vol. 13, no. 1, pp. 190–201, 2023.
- [12] Y. Bansal and C. Madhu, "A novel high-speed approach for 16×16 vedic multiplication with compressor adders," *Computers and Electrical Engineering*, vol. 49, pp. 39–49, 2016.

- [13] V. Kapse, A. Jain, and M. Pattanaik, "Design of an area efficient and low power mac unit," in *Smart Trends in Information Technology and Computer Communications: First International Conference, SmartCom 2016, Jaipur, India, August 6–7, 2016, Revised Selected Papers 1*, pp. 276–284, Springer, 2016.
- [14] B. V. R. M. Sirisha, Dr. K. V. Rama Rao, "A novel fpga based mac unit design using reversible logic gate approach," *World Journal of Engineering Research and Technology WJERT*, vol. 6, pp. 276–283, 2020.
- [15] N. K. V, "Performance analysis of mac unit using booth, wallace tree, array and vedic multipliers," *International Journal of Engineering Research and Technology (IJERT)*, vol. 9, pp. 497–504, 2020.
- [16] M. Ramprasad and P. Kodavanti, "Design of high speed and area efficient 16 bit mac architecture using hybrid adders for sustainable application," *Journal of Green Engineering*, vol. 10, pp. 11809–11818, 2020.
- [17] D. S. Manikanta, K. S. S. Ramakrishna, M. Giridhar, N. Avinash, T. Srujan, *et al.*, "Hardware realization of low power and area efficient vedic mac in dsp filters," in *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 46–50, IEEE, 2021.
- [18] M. N. M. S. N. Suma Nair, K. Sai Naveen, "Design of vedic mathematics based on mac unit for power optimization," in *International Journal of Research in Engineering and Science (IJRES)*, pp. 284–289, IEEE, 2022.
- [19] A. Bhadra and S. Samui, "Design and analysis of high-throughput two-cycle multiply-accumulate (mac) architectures for fixed-point arithmetic," in *2022 IEEE Calcutta Conference (CALCON)*, pp. 267–272, IEEE, 2022.
- [20] b. Mangapathi Vinitha, "Fpga implementation of multiplier-accumulator unit using vedic multipliers and reversible gates," in *Journal of Xi'an Shiyu University, Natural Science Edition*, pp. 1869–1877, IEEE, 2023.
- [21] Y. J. M. Shirur, "Distributed arithmetic mechanization of multiply and accumulate core for dsp applications," *Journal of Survey in Fisheries Sciences*, vol. 10, no. 2S, pp. 595–603, 2023.
- [22] F. K. Al Assfor, I. S. Al-Furati, and A. T. Rashed, "Vedic-based squarers with high performance," *Indonesian Journal of Electrical Engineering and Informatics (IJEI)*, vol. 9, no. 1, pp. 163–172, 2021.
- [23] V. Bianchi and I. De Munari, "A modular vedic multiplier architecture for model-based design and deployment on fpga platforms," *Microprocessors and Microsystems*, vol. 76, p. 103106, 2020.
- [24] R. Dr.S.K.Oza, "8x8 vedic multiplier using reversible logic gate," *Ann. For. Res*, vol. 66, no. 1, pp. 3267–3277, 2023.
- [25] S. N. Gadakh and A. Khade, "Design and optimization of 16×16 bit multiplier using vedic mathematics," in *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, pp. 460–464, IEEE, 2016.
- [26] S. Akhter and S. Chaturvedi, "Modified binary multiplier circuit based on vedic mathematics," in *2019 6th international conference on signal processing and integrated networks (SPIN)*, pp. 234–237, IEEE, 2019.
- [27] H. G. A. M. A. V. V. Pradeepa S C, Gowri G Bennur, "Design and vlsi implementation of vedic multiplier using 45nm technology," in *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*, pp. 964–968, IEEE, 2023.
- [28] M. A. I. Korupoju Vikas, "Vlsi implementation of high-performance partial product reduction vedic mac," in *INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)*, pp. 571–576, IEEE, 2022.
- [29] D. Yaswanth, S. Nagaraj, and R. V. Vijeth, "Design and analysis of high speed and low area vedic multiplier using carry select adder," in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pp. 1–5, IEEE, 2020.
- [30] M. B. Murugesh, S. Nagaraj, J. Jayasree, and G. V. K. Reddy, "Modified high speed 32-bit vedic multiplier design and implementation," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 929–932, IEEE, 2020.
- [31] K. Rajesh and G. U. Reddy, "Fpga implementation of multiplier-accumulator unit using vedic multiplier and reversible gates," in *2019 Third International Conference on Inventive Systems and Control (ICISC)*, pp. 467–471, IEEE, 2019.