IUST
Iran University of Science and Technology

Iranian Journal of Electrical and Electronic Engineering

Journal Homepage: ijeee.iust.ac.ir

Research Paper

Design and Analysis of an Iterative Carry Save Adder-based Power-Efficient Multiplier

T. Mendez* and S. G. Nayak*(C.A.)

Abstract: The need for low-power VLSI chips is ignited by the enhanced market requirement for battery-powered end-user electronics, high-performance computing systems, and environmental concerns. The continuous advancement of the computational units found in applications such as digital signal processing, image processing, and high-performance CPUs has led to an indispensable demand for power-efficient, high-speed and compact multipliers. To address those low-power computational aspects with improved performance, an approach to design the multiplier using the algorithms of Vedic math is developed in this research. In the proposed work, the pre-computation technique is incorporated that aided in estimation of the carries during the partial product calculation stage; that enhanced the speed of the multiplier. This design was carried out using Cadence NCSIM 90 nm technology. The comparative analysis between the proposed multiplier design and the multipliers from the literature resulted in a substantial improvement in power dissipation as well as delay. The research was extended to assess the designed architectures' performance statistically, applying the independent sample t-test hypothesis.

Keywords: Low-Power, Multiplier, Pre-Computation, VLSI Implementation.

1 Introduction

THERE is a growing demand in the complexity of functionalities essential for high-speed data processing and embedded devices used in multimedia, image processing, and other data control applications. This has risen due to the tremendous growth in the density of an integrated circuit (IC).

The rapid growth of interconnect and processor technology is the driving force for the steady surge in the number of gates for nearly three decades on a very large-scale integration (VLSI) chip. To maintain the application constraints endorsed by the IC, two aspects that are vital in an IC are the power efficiency and speed that is indirectly dependent on the multiplier. The dawn of modern process technologies alongside the accessibility of contemporary computer-aided design (CAD) tools has given birth to a broad assortment of complex applications.

An essential operation of the arithmetic unit is multiplication which is a power-consuming element. A typical CPU generally allocates a considerable quantity of time, processing the multiplication operation of the arithmetic unit. Multiplication considerably requires more hardware resources as well as processing time when compared with addition and subtraction operations. While considering delay, the traditional multiplication involving the shift and add method is generally not preferable.

VLSI implementation of multipliers involves three crucial considerations, namely power, delay, and area. The basic operations in multiplication can be arranged as 3 steps: The initial step includes the generation of partial products determined by multiplying the multiplicand and multiplier. The reduction of partial products can be considered as the next step. The addition with carry propagation will be the third and final step. A significant role is played by the reduction of partial products in accordance with the conditions of power, and delay.

Iranian Journal of Electrical and Electronic Engineering, 2022.
Paper first received 14 July 2021, revised 25 September 2021, and accepted 04 October 2021.
* The authors are with the Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Udupi, Karnataka, India.
E-mails: mendez.tanya@gmail.com and gs.nayak@manipal.edu.
Corresponding Author: S.G. Nayak.
https://doi.org/10.22068/IJEEE.18.1.2238

This work proposes to explore the trade-offs of high-performance multiplier designs using low power techniques to enhance the efficiency in power as well as delay. The paper is organized as: The previous work on low power multipliers is given in Section 2. The proposed design of the power-efficient multiplier and the methodology are given in section 3. Section 4 gives the data analysis with results and inferences. Finally, section 5 delivers the conclusion.

2 Previous Works

Multipliers, a vital component of the arithmetic unit, generally require a compact design in terms of less power and speed. The focus in the VLSI design of a multiplier would be to reduce the dynamic power, which forms a significant portion of the total power. This can be obtained by reducing the number of operations. The handling of data with comparatively minimal power and delay necessitates the parallel operation of several multipliers. Previous work on the implementation of multipliers using hardware includes various algorithms. The reduction in the computational complexity and the partial products summation is a decisive criterion in the performance determination of parallel multipliers. One such parallel multiplier to reduce the computational complexity was a 16×16 multiplier designed for low power consumption and superior speed by Chakravarthy *et al*. [1]. Reduction of data-path, rearrangement methods, and the addition of a minor element in the data path was used to curtail the processor static power and lag. The area occupied by the multiplier was a bit more than the existing architectures due to the increased number of logic elements.

The multipliers built on the idea of Vedic arithmetic are proving to be quick and low-power devouring models. A Vedic multiplier designed employing Urdhva Tiryagbhyam (UT) sutra and a hybrid adder structure was used to enhance processing speed and was tested using Verilog HDL by Sri Lakshmi and Vigneswaran [2]. The hybrid adder structure comprised two multiplexers and an XOR gate, which acted as the selection line for the multiplexers. The full adders used in the carry select adder of the Vedic multiplier were further replaced by the hybrid adder, which made it faster and power-efficient as the propagation delay was reduced. Improved results were observed utilizing the hybrid adder considering the power and delay compared to the traditional full adders and compressors.

The authors Eshack and Krishnakumar [3] utilized reversible gates and pipelining in UT sutra to design Vedic multipliers resulting in increased speed and reduced consumption of power. The multiplier was designed considering the drawbacks of delay in the partial product generation and the power consumption prevalent in multipliers. Pipelining was used for the summation of partial products reducing the computational delay. The inherent reversible ability of Toffoli gates utilized in the proposed pipelined multipliers further lowered the delay. Another 8-bit Vedic multiplier was designed by Chandrashekara and Rohith [4] with reduced computational delay to enhance the performance. For the summation of the partial products, a Modified Carry Save Adder (MCSA) was used. A 2×2 multiplier was first designed to construct a Vedic multiplier of size 4-bit that was eventually utilized in designing the Vedic multiplier of size 8-bits. A comparative analysis based on power, delay, and Look-Up Table (LUT) by designing 8 x 8 binary multipliers using 4 different algorithms of Vedic Mathematics was carried out to analyze the speed by Mehra *et al.* [5]. The four multiplication sutras of Vedic math's UT, Nikhilam, Anurupyena, and Karatsuba algorithm were used for the design of four different 8×8 binary multipliers. Based on the analysis, the least power consumption was 240 mW for the UT multiplier. The fastest algorithm was also the UT multiplier with a 59.5 ns delay. The most area-efficient algorithm was observed to be the Karatsuba algorithm consuming only 42 LUTs.

Chaitanya *et al*. [6] developed a novel design of a multiplier applying the Vedic sutra UT employing multiplexers instead of the existing half and fuller adders. A further reduction in the delay was observed by implementing pre-computation logic in which multiplexers were used in place of adders. Pradhan and Panda [7] proposed an improved multiplier architecture by employing the Nikhilam algorithm of Vedic arithmetic and the carry-save method of addition to enhance the speed further. The two's complement of the multiplicand and multiplier was taken, to reduce the repeated multiplication process. The most significant part of the product was achieved by multiplying the complements, and the least significant part was obtained utilizing a carry-save adder. The design of a multiplier for complex numbers utilizing Vedic mathematics centered on the Nikhilam sutra was developed by Saha *et al*. [8]. The algorithm was split into three steps: the selection of radix, the determination of exponent, and finally the multiplication. Multiplication of operands was accomplished in a single step applying parallel computation which resulted in superior performance as the carry propagation was lowered.

The techniques of UT sutra were incorporated in binary multipliers by Rashid *et al.* [9] thus enhancing the speed through parallel generation of partial products and exclusion of unnecessary multiplications. Even though the multiplier was faster than conventional multipliers, the power can be reduced further. The literature on different Vedic algorithm-based multiplier architectures signified that the delay of the designs using Vedic arithmetic was considerably less when compared with conventional techniques. The Vedic multipliers implemented utilizing the UT Sutra algorithm were found to be power-efficient and fast. The multiplier designs focused on distinct execution

techniques that incorporated carry select adders, Ripple Carry Adders (RCA), and compressors for the calculation of partial products [11-15]. To improve the performance of DSP processors, an 8×8 multiplier utilizing the UT sutra of Vedic arithmetic was formulated by Pushpangadan *et al.* [16]. The application of parallelism in the addition of partial products enhanced the speed and regularity of the circuit. The application of suitable low power techniques can further improve the performance of the multiplier from the perspective of power and speed to cater to the demands of DSP processors.

The design of a 4×4, 8×8, and 16×16 multiplier using UT sutra to target current DSP processors that require high speed and low power were discussed by Basha *et al.* [17]. Three multiplier models were designed utilizing RCA, Carry Save Adder (CSA), and the hybrid combination of RCA and CSA for the addition of partial products. The performance parameters analyzed using Monte Carlo (MC) simulation in Cadence indicated that the multiplier designed with modified RCA exhibited significant benefits in terms of power. However, the delay reported in all the designs was as high as 91 ns, making them unsuitable for high-performance applications.

Sethi and Panda [18] proposed an n-bit binary circuit to execute the squaring operation eliminating multipliers. The circuit designed calculates the square of n-bit numbers by utilizing the squaring circuit of size 2 bits recursively n-2 times. The proposed circuit comprises two adder/ subtractor blocks of size n-bits, an adder block of size 2n-bits, and a squaring circuit of n-1 bits. Even though the circuit was faster compared to conventional designs, the power consumption was as high as 0.253 watts making them unsuitable for low power applications. Shoba and Nakkeeran [19] designed a hierarchical multiplier with reduced computational delay by using a carry select adder and Binary to Excess 1 Converter (BEC). The proposed Vedic multiplier was used in the base multiplier of the hierarchy to reduce the area and delay. The implementation of the multiplier with Gate Diffusion Input logic resulted in power reduction.

Sahu *et al*. [20], using the Nikhilam algorithm of Vedic Maths techniques, designed a binary multiplier for signed numbers. The conversion of significant to small operand multiplication with addition led to substantial benefits in speed. The proposed multiplier dealt with more than a bit of the multiplier and multiplicand for each clock cycle, resulting in fewer cycles. The process of multiplication was used for both signed binary and decimal numbers. Deepa and Marimuthu [21] have designed a binary multiplier using the principles of Yavadunam algorithm of Vedic maths. There were substantial speed improvements, the layout had reduced complexity, and the multiplier could multiply any range of binary numbers. The multipliers were unsuitable for low-order bits, as they consumed more power but proved favorable for higher-order bits from 32 bit onwards.

The existing multipliers consume more power making them unsuitable for predominant use in portable battery-operated devices. The multipliers utilized in high-speed processors functioning at high clock frequencies take more delay for the execution of instructions. Power saving is thus an important focus of development. The processor performance can be enhanced by delay-power product reduction of the multipliers. To address low power computational aspects with enhanced performance, an attempt to design the multiplier utilizing the algorithms of Vedic math has been explored. The pace of the multiplier is strengthened by utilizing Vedic arithmetic as the operations performed are extremely fast and require reduced hardware. Further pre-computation [22] technique is employed to estimate the carries during the partial product calculation stage to improve the speed of the multiplier. This will result in a power-efficient multiplier with reduced delay addressing the main research gap. This multiplier design offers a low-power and fast approach that can be utilized in image processing applications like image multiplication, sharpening, smoothening, filtering, etc.

3 Proposed Design

The positivist paradigm of the research can facilitate the evaluation of the proposed multiplier in terms of its ability to reduce the power dissipated, as it can provide empirical proof through modeling and simulation. This method mainly adopts a quantitative approach to research as it requires the theory to be tested involving a systematic investigation utilizing computational techniques to draw inferences. Further, modeling the proposed multiplier using Verilog HDL has effectively measured the power and delay. The simulation was carried out using Cadence NCSIM, and finally, the 90 nm technology library was used during the synthesis that was accomplished with the help of the Cadence RTL Compiler. Random binary input data was applied to carry out the quantitative study in regular order and to test the correctness of the model that was designed.

Research on the low power design of multipliers makes a significant contribution to the field of VLSI as multipliers form an indispensable part of almost all high-speed processors and CPUs. Firstly, Moore's Law predicts that the number of devices on a chip increases twice every eighteen months. This in turn will enhance the power dissipation as the number of devices is growing. An obvious and imperative low power demand emerges from such advancement forces of electronic circuits. Secondly, there is a rapid surge in demand on the market for compact and lightweight electronic devices driven by batteries. This desire for smaller and more robust electronic devices ultimately transforms to low-power needs that enhance cost-effectiveness.

Thirdly, the computerization equipment furnishing modern-day offices cause immense ingestion of power that demands low power devices to counteract the environmental concerns. The low power multiplier proposed, reduces the power consumed, making it suitable for low power applications. This, in turn, reduces the product of power and delay of the multiplier.

Vedic arithmetic, gifted by the ancient gurus of India is effective in reducing tricky and complicated computations to a very straightforward one. It comprises 16 sutras that demonstrate how computations can be accomplished mentally and in a modest style. UT is a universal and effective technique that can be utilized to determine the product of any two numbers. This is often considered to be the crowning jewel of all sutras. The UT is a vertically and crosswise technique to achieve the final product of two numbers. The number of steps required to obtain the product is one less than the overall number of digits of both numbers. The formula is extremely terse and small, comprising of just a composite term meaning "vertically and crosswise", the applications of which are manifold. The multipliers built on the idea of Vedic arithmetic are proving to be quick and low-power devouring models. This results in a multiplier utilizing lower power compared to the traditional multipliers.

The ever-increasing usage of battery-powered portable electronic appliances is the steering need for integrated circuits that consume the least feasible amount of power. This craving towards compact, lightweight, and more robust electronic gadgets transforms implicitly to minimal power requirements. Ironically, high-performing processing systems characterized by significant power dissipation also drives the lower power demands. The computation time of the multiplier is reduced by utilizing a Vedic Multiplier (VM) of 8-bit size, designed with inputs as X, Y, and Z as the 16-bit result. The multiplier suggested is represented in Fig. 1.

In the suggested multiplier, inputs are split into upper and lower sections each sized four bits. Multiplication comprises of two operations: the creation of the partial products and their summation. Two promising approaches to speed up the multiplication process include reduction of partial products or pace up the summing process. The two approaches could be applied concurrently. The reduction of partial products also lowers the time necessary to execute their sum. Three basic stages are followed throughout the majority of high-performance and low power multipliers: generation of partial products in a parallel fashion, reduction of partial products, conversion of the sum into a nonredundant version employing a high-speed adder as illustrated in Fig. 2.

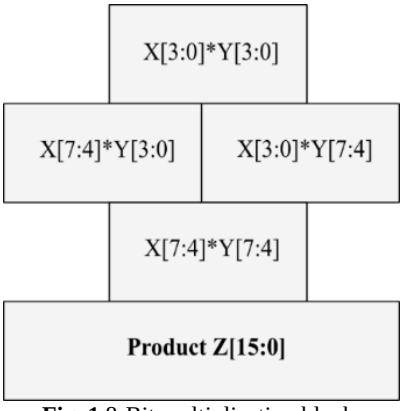

Fig. 1 8-Bit multiplication block.

The first phase in the architectural design of the 4-bit multiplier is the computation of partial products. The inputs to this phase are the multiplicand [A_3 A_2 A_1 A_0] and the multiplier [B_3 B_2 B_1 B_0] and the outputs are [P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0]. The generated partial products P_{00} to P_{33} which are obtained from the input operands are illustrated in Fig. 3. The partial products can be obtained by utilizing the logical AND operation $P_{ij} = A_j.B_i$, where i is the column index and j, is the row index. The basic design is based on an array of n^2 AND gates. The generation of carry is the second phase in architectural design. The UT algorithm of Vedic arithmetic is applied. The C_{ij} term relates to the carry generated from the i-th to the j-th sum stage.

$$C_{12} = P_{01}\ ?(1'b0) : (P_{10}) \tag{1}$$

$$C_{23} = P_{20}\ ?(P_{02}.P_{11}) : (P_{02}.P_{11} + P_{02} + P_{11}) \tag{2}$$

$$C_{23}^{1} = C_{12}\ ?(1'b0) : [(P_{02}.P_{11}' + P_{02}'P_{11}) \oplus P_{20}] \tag{3}$$

$$C_{34} = P_{21}\ ?(P_{03}.P_{12}) : (P_{03}.P_{12} + P_{03} + P_{12}) \tag{4}$$

$$C_{34}^{1} = C_{23}\ ?(C_{34}.P_{30}) : (C_{34}.P_{30} + C_{34} + P_{30}) \tag{5}$$

$$C_{45} = P_{13}\ ?(1'b0) : (P_{22}) \tag{6}$$

$$C_{45}^{1} = C_{34}\ ?(C_{45}.P_{31}) : (C_{45}.P_{31} + C_{45} + P_{31}) \tag{7}$$

$$C_{56} = C_{45}\ ?(P_{23}.P_{32}) : [P_{23}.P_{32} + (P_{23} \oplus P_{32})] \tag{8}$$

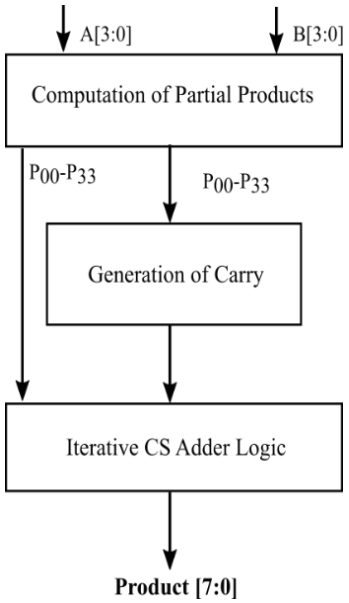

Fig. 2 4-Bit iterative carry save adder based multiplier.

7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		A_3	A_2	A_1	A_0		
	*	B_3	B_2	B_1	B_0		
		C_{45}^1	C_{34}^1	C_{23}^1			
C_{67}	C_{56}	C_{45}	C_{34}	C_{23}	C_{12}		
				P_{30}	P_{20}	P_{10}	P_{00}
			P_{31}	P_{21}	P_{11}	P_{01}	
		P_{32}	P_{22}	P_{12}	P_{02}		
	P_{33}	P_{23}	P_{13}	P_{03}			
P_7	**P_6**	**P_5**	**P_4**	**P_3**	**P_2**	**P_1**	**P_0**

Fig. 3 Multiplication process 4-bit.

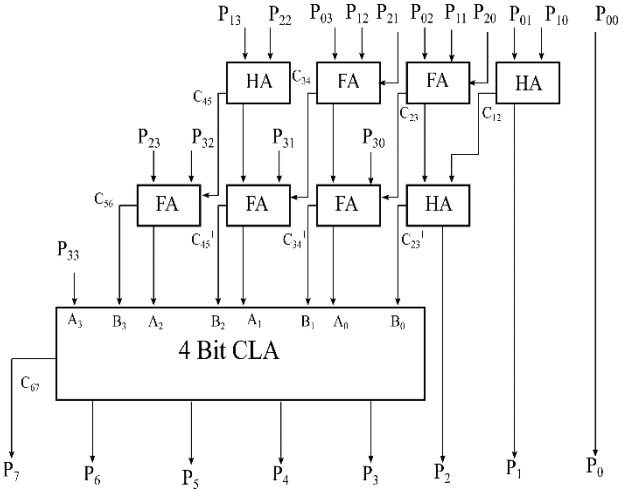

Fig. 4 4-Bit multiplier based on iterative carry save adder.

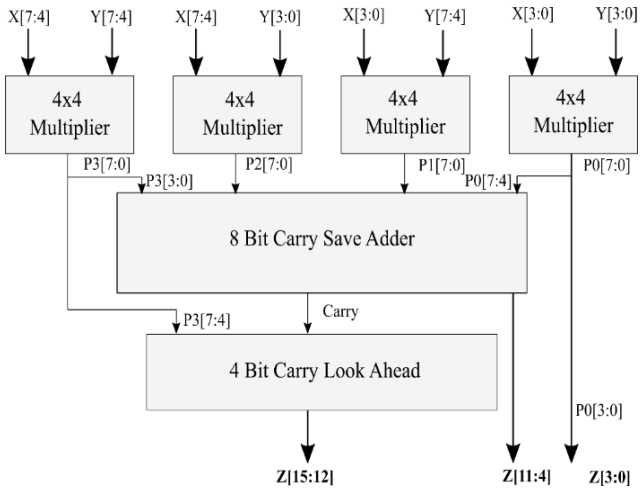

Fig. 5 8-Bit multiplier using iterative carry save adder.

The carry produced from column 1 is C_{12}, the carry produced from column 2 are C_{23} and C^1_{23}, the carry produced from column 3 are C_{34} and C^1_{34}, the carry produced from column 4 are C_{45} and C^1_{45}, the carry produced from columns 5 and 6 are C_{56} and C_{67}, respectively. The equations indicate the carry produced utilizing the technique of pre-computation in each column.

This optimization technique of generation of carries is based on selectively calculating the carries by pre-computing them prior to when they are necessary and then utilizing these pre-calculated carry values to decrease the internal switching activity. The third phase in the architectural design utilizes an iterative carry-save adder logic, as illustrated in Fig. 4 for the column-wise computation of the generated partial products and carries. The 4-bit iterative carry-save method expressed by the algorithm can be applied to execute the calculation among these partial products and the appropriate carries. This carry-save scheme is a type of array addition. The adder method applied for the final summation of the two operands is carry look ahead. The output provides the 8-bit product that is obtained parallelly.

The summation of the partial product generated and the required carry to obtain the final product bits of the multiplication operation is obtained using XOR logic. Multipliers usually comprise adders; thus, an iterative carry-save adder design with reduced delay is a considerable move to boost the multiplier efficiency. The iterative carry save multiplier benefits from the decreased computation time, as it is about twice as fast compared to the ripple carry method employed in typical multipliers. The usage of CLA in the final stage further reduces the speed by two. The structure of the multiplier of size 8 bits is illustrated in Fig. 5, where the products are obtained utilizing the above designed 4-bit iterative carry-save based multiplier.

The products are summed using the following addition tree shown in Fig. 6. The multiplier of size 16 bits is designed in a similar fashion utilizing the four 8-bit low power multipliers designed, a 16-bit carry-save, and an 8-bit carry look-ahead adder. The 32-bit multiplier is designed similarly using four 16-bit low power multipliers designed, a 32-bit carry-save and a 16-bit carry look-ahead adder. Multipliers of higher orders can be developed in a similar style.

4 Results and Discussion

The proposed multiplier and existing designs presented have been synthesized and implemented utilizing the 90nm digital standard cell library. The designs have been executed in the same framework and

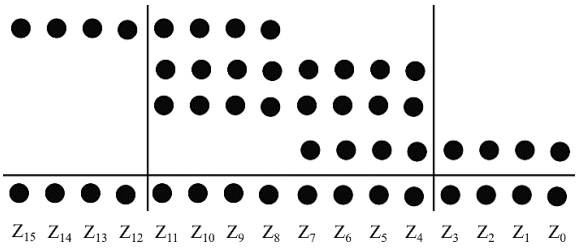

Fig. 6 Addition tree structure for 8-bit multiplication.

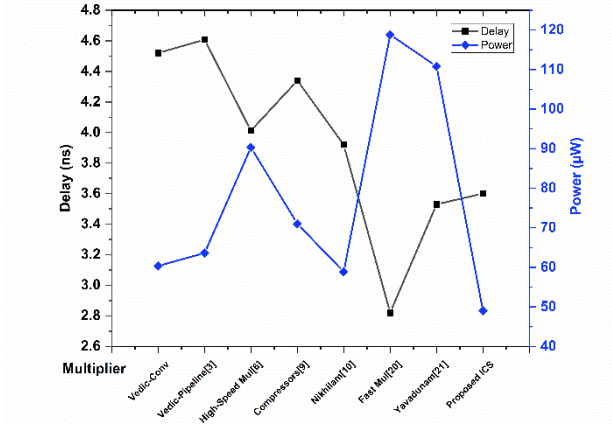

Fig. 7 Delay and power comparison of various 8-bit multipliers.

Table 1 Delay, dissipated power, PDP, and EDP of various 8-bit binary multipliers.

Architecture	Delay [ns]	Power-dynamic [µW]	Power-static [µW]	Power-total [µW]	PDP [fJ]	EDP [e^{-21}Js]
Vedic-conv	4.52	48.91	11.47	60.38	272.92	1.23
Vedic-pipeline [3]	4.607	54.47	9.14	63.61	293.05	1.350
High-speed multiplier [6]	4.012	84.278	6.011	90.29	362.24	1.45
Multiplier based on compressors [9]	4.34	60.54	10.45	70.99	308.097	1.337
Nikhilam algorithm multiplier[10]	3.92	49.56	9.3	58.86	230.731	0.904
Fast multiplier [20]	2.82	100.02	18.77	118.794	334.99	0.9446
Yavadunam Sutra based multiplier [21]	3.53	98.09	12.72	110.81	391.159	1.38
Proposed ICS	3.6	41.534	7.487	49.021	176.475	0.635

Table 2 Delay, dissipated power, PDP, and EDP of various 16-bit binary multipliers.

Architecture	Delay [ns]	Power-dynamic [µW]	Power-static [µW]	Power-total [µW]	PDP [fJ]	EDP [e^{-21}Js]
Vedic-conv	5.808	207.063	47.736	254.79	1479.87	8.59
Vedic-pipeline [3]	5.547	204.729	38.231	242.96	1347.699	7.475
High-speed multiplier [6]	5.24	327.69	42.46	370.69	1942.41	10.178
Multiplier based on compressors [9]	6.04	190.4	13.58	203.98	1232.04	7.44
Nikhilam algorithm multiplier[10]	4.96	180.65	38.1	218.75	1085	5.38
Fast multiplier [20]	3.61	598.86	126.31	725.176	2617.88	9.45
Yavadunam Sutra based multiplier [21]	6.01	645.93	83.71	729.65	4385.19	26.35
Proposed ICS	4.547	172.736	32.439	200.517	932.93	4.24

delay and power parameters are assessed. Inputs for experimental purposes were taken in a normal format. Verilog Hardware Description Language (HDL) is utilized for the Register Transfer Level (RTL) entry of the functional design by mapping it through a logic synthesis process into preassembled library cells. The design functionality of the proposed multiplier design and the multiplier designs from the literature review are verified with random binary inputs by assessing the predicted product with the true product. The predicted and true products were observed to match for all designs. The RTL verification was conducted utilizing the Cadence Incisive tool. Power analysis is performed using different toggle rates for the inputs.

The process of transforming the technology-independent RTL into a technology-dependent netlist is carried out during the logic synthesis process by utilizing the process library. The toggling information of the output ports and internal signals generated during the design functional verification is captured by the value change dump (VCD) file that is given as one of the inputs to the logic synthesis. The RTL describes the functional design. The process library describes the logical, physical, and electrical information of the logic gates. The conversion of RTL into an optimized netlist is obtained by the Electronic Design Automation (EDA) tool. The logic synthesis of the proposed low power Vedic multiplier and existing multiplier designs is achieved using the Genus tool of Cadence with 90 nm technology library. The power consumption calculation is obtained using a 1GHz clock frequency.

The prevailing 8-bit, 16-bit, and 32-bit multipliers Eshack *et al*., [3], Chaitanya *et al*. [6], Rashid *et al*., [9], Saha *et al*., [10], Sahu *et al*. [20], Deepa and Marimuthu [21] presented in the literature review are designed employing Verilog HDL and along with the application of random inputs, the functionality was verified and implemented using 90 nm library. The performance of the 8-bit multipliers with respect to delay, power, power-delay product (PDP), and energy-delay product (EDP) is presented in Table 1 and plotted in Fig. 7. The proposed 8-digit, 16-digit, and 32-digit multiplier using the ICS adder were designed and implemented.

The following inferences were drawn from Table 1.

1. The delay of the 8-bit multiplier proposed was lesser

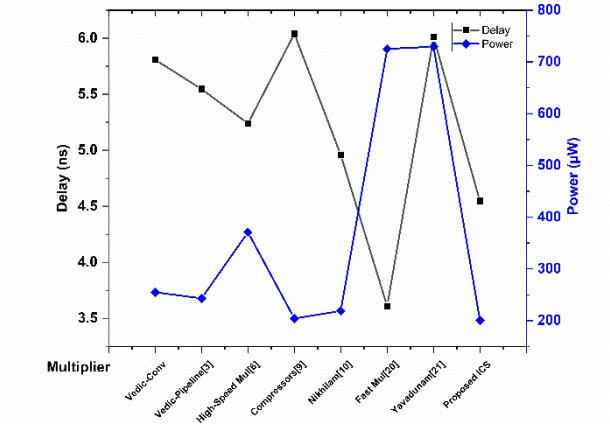

Fig. 8 Delay and power comparison of various 16-bit multipliers.

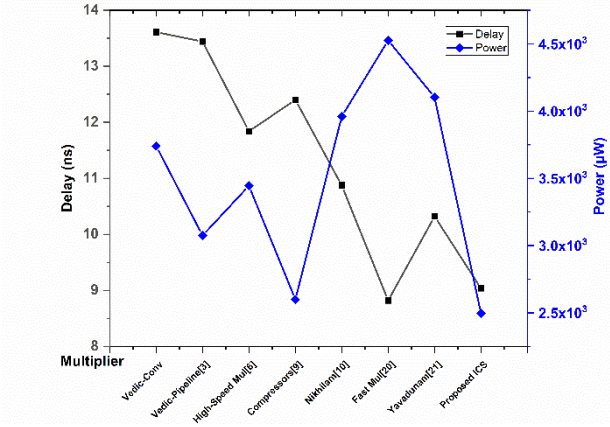

Fig. 9 Delay and power comparison of various 32-bit multipliers.

Table 3 Delay, dissipated power, PDP, and EDP of various 32-bit binary multipliers.

Architecture	Delay [ns]	Power-dynamic [µW]	Power-static [µW]	Power-total [µW]	PDP [fJ]	EDP [e^{-21}Js]
Vedic-conv	13.610	3424.386	316.107	3740.494	50.9	692.86
Vedic-pipeline [3]	13.44	2878.192	196.890	3075.082	41.329	555.46
High-speed multiplier [6]	11.84	3045.06	399.57	3444.64	40.78	482.88
Multiplier based on compressors [9]	12.40	558.94	39.86	2598.81	32.22	399.13
Nikhilam algorithm multiplier[10]	10.875	3270.20	689.703	3959.910	43.06	468.32
Fast multiplier [20]	8.82	3737.35	788.311	4525.67	39.916	352.06
Yavadunam Sutra based multiplier [21]	10.32	3631.68	470.707	4102.39	42.336	436.914
Proposed ICS	9.0385	212.028	2284.288	2496.316	22.56	203.935

in comparison to other designs having a value of 3.6 ns.

2. In contrast to the mentioned designs, the proposed 8-bit multiplier design exhibits minimum total power, PDP and EDP of 49.021 µW, 176.475 fJ, and 0.635 e^{-21} Js, respectively.

The performance of the 16-bit multipliers with respect to delay, power, PDP, and EDP is presented in Table 2 and plotted in Fig. 8.

The following inferences were drawn from Table 2.

1. The delay of the proposed 16-bit multiplier design was lesser than other designs with a value of 4.547 ns.
2. In contrast to the mentioned designs, the proposed 16-bit multiplier design exhibits minimum total power, PDP and EDP of 200.517 µW, 932.93 fJ, and 4.24 e^{-21} Js, respectively.

The performance of the 32-bit multipliers with respect to delay, power, PDP, and EDP is presented in Table 3 and plotted in Fig. 9.

The following inferences were drawn from Table 3.

1. The delay of the proposed 32-bit multiplier design was lesser than other designs with a value of 9.0385 ns.
2. In contrast to the mentioned designs, the proposed 32-bit multiplier design exhibits minimum total power, PDP and EDP of 2496.316 µW, 22.56 pJ, and 203.935 e^{-21} Js respectively.

The PDP indicates the design's energy efficiency and sometimes designs with low PDP tend to have slow performance. To evaluate the performance of the proposed design, another performance metric, EDP, is calculated. The PDP and EDP can be easily estimated using (9) and (10).

$$PDP = Power \times Delay \tag{9}$$

$$EDP = PDP \times Delay \tag{10}$$

It is evident from Tables 1, 2, and 3 that the proposed multiplier using ICS consumes lesser power and has an improved PDP and EDP in comparison to existing multipliers mentioned in the literature review. The plot in Fig. 10 illustrates the PDP-EDP plot for various 8-bit multipliers.

To ensure that the incorporation of pre-computation in conjunction with ICS adder in the proposed low power Vedic multiplier indicates a substantial improvement in power the null and alternative hypothesis have been framed in the context of this research as follows:

Null Hypothesis (H_0): There is no significant increase in multiplier performance with the application of pre-computation techniques in comparison to conventional techniques. (H_0: $\mu_1 = \mu_2$).

Alternative Hypothesis (H_a): There is a significant increase in multiplier performance with the application of pre-computation techniques in comparison to conventional techniques. (H_0: $\mu_1 > \mu_2$).

The statistics of the power of the multipliers with conventional techniques and pre-computation techniques are shown in Table 4.

The hypothesis was tested using an independent one-tailed sample t-test with a 0.05 significance level exploiting SPSS software. The significance ≤ 0.001 was obtained for 2-tailed test and for 1-tailed test, the significance ≤ 0.001/2 = 0.0005 < 0.05. Therefore, the null hypothesis was rejected. The alternate

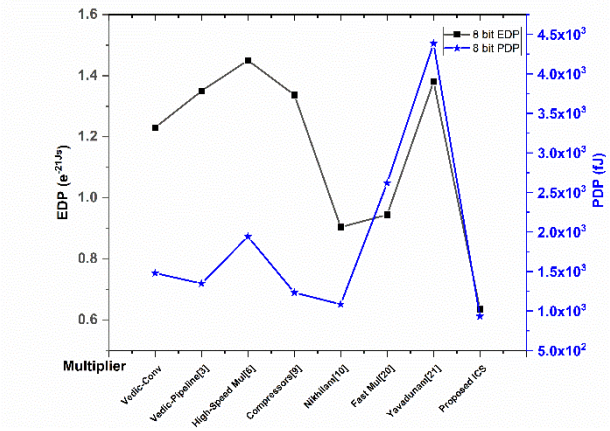

Fig. 10 PDP-EDP comparison of various 8-bit multipliers.

Table 4 8-Bit multiplier power with conventional and pre-computation techniques.

SI. No.	Power [µW] (Conventional)	SI. No.	Power [µW] (Pre-computation)
1	61	1	49
2	63	2	48
3	63	3	50
4	58	4	47
5	60	-	-
6	65	-	-
Sample size, $n_1 = 6$		Sample size, $n_2 = 4$	

Group Statistics

	Type	N	Mean	Std. Deviation	Std. Error Mean
Power	1	6	61.67	2.503	1.022
	2	4	48.50	1.291	.645

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Power	Equal variances assumed	2.304	.168	9.571	8	<.001	13.167	1.376	9.994	16.339
	Equal variances not assumed			10.893	7.734	<.001	13.167	1.209	10.362	15.971

Independent Samples Effect Sizes

		Standardizer[a]	Point Estimate	95% Confidence Interval	
				Lower	Upper
Power	Cohen's d	2.131	6.178	2.931	9.379
	Hedges' correction	2.361	5.577	2.646	8.467
	Glass's delta	1.291	10.199	2.600	18.088

a. The denominator used in estimating the effect sizes.
Cohen's d uses the pooled standard deviation.
Hedges' correction uses the pooled standard deviation, plus a correction factor.
Glass's delta uses the sample standard deviation of the control group.

Fig. 11 Independent sample *t*-test result.

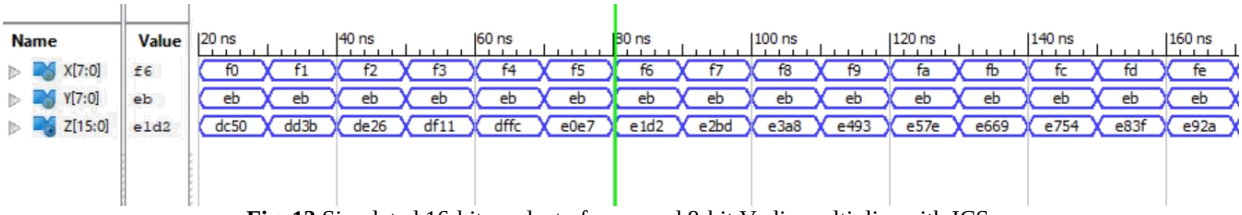

Fig. 12 Simulated 16-bit product of proposed 8-bit Vedic multiplier with ICS.

hypothesis (H_a): There is a significant increase in multiplier performance with the application of pre-computation techniques in comparison to conventional techniques (H_0: $\mu_1 > \mu_2$) was accepted. The results of the hypothesis test conducted are as indicated in Fig. 11. The introduction of the pre-computation technique along with the application of Iterative Carry Save Adder improved the multiplier performance concerning power and delay. There was a significant improvement in the product of power and delay of the multipliers in comparison with the conventional multipliers mentioned.

The simulated products of the 8-bit, 16-bit, and 32-bit proposed Vedic multiplier designs are illustrated in Figs. 12, 13, and 14, and are in accordance with the expected values.

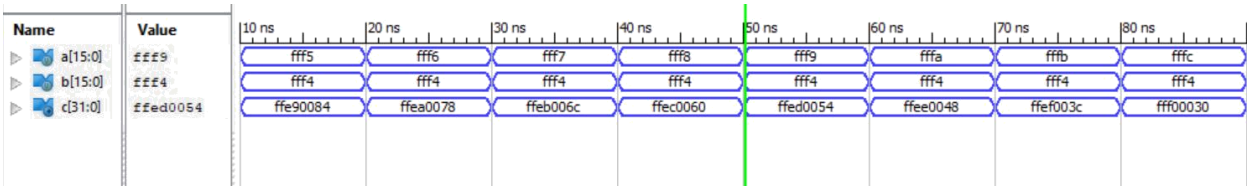

Fig. 13 Simulated 32-bit product of proposed 16-bit Vedic multiplier with ICS.

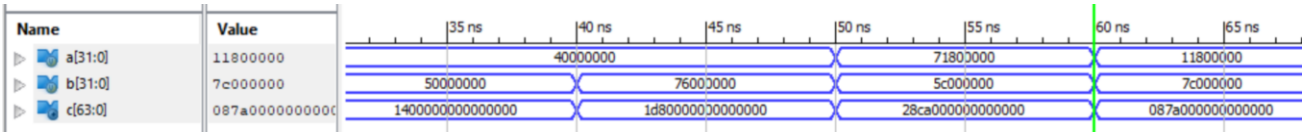

Fig. 14 Simulated 64-bit product of proposed 32-bit Vedic multiplier with ICS.

5 Conclusion

A multiplier based on Vedic arithmetic utilizing iterative carry-save adder and pre-computation of carries is designed. Three phases are followed in the multiplier design, computation of partial products in a parallel fashion, generation of carry, and conversion of the sum into a nonredundant version employing a high-speed adder. The computation of partial products was designed utilizing the logical AND operation that required an array of n^2 AND gates for an n-bit multiplier. The generation of carry was designed by applying the technique of pre-computation. An iterative carry-save adder logic is used for the column-wise computation of the generated partial products and carries. The usage of CLA in the final stage further reduced the speed by a factor of two. The simulation results achieved in the Genus tool of Cadence with 90 nm technology library approve the effectiveness of the proposed multiplier from the perspective of speed, power dissipation, PDP, and EDP when compared with conventional multipliers.

The proposed design exhibited a reduction of 18.81% in power dissipation and a decrease of 20.3% in the delay whilst the PDP and EDP improved by 35.34% and 48.37% respectively. These performance figures signify that the proposed design is beneficial for applications that necessitate fast operations with minimal power consumption. The paper also studied the statistical analyses of the designed architecture by testing the hypothesis (using an independent sample t-test). The proposed multiplier promises its power-efficient nature as an integral part of ALU for signal and image processing applications with stringent demands of power and delay. In future work, the limitations of the proposed power-efficient multiplier are to be addressed by improvising the hybrid architecture.

Intellectual Property

The authors confirm that they have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property.

Acknowledgment

The first author would like to thank Manipal Academy of Higher Education, Manipal Institute of Technology, Manipal, for granting Dr. T.M.A Pai Fellowship to carry out this research work. The authors are also thankful to the Department of Electronics and Communication Engineering, Manipal Institute of Technology for providing laboratory facilities. The authors would also like to thank Dr. Chaitanya CVS who is currently working as Software Verification Engineer at Intel Corporation for his valuable support.

CRediT Authorship Contribution Statement

T. Mendez: Conceptualization, Methodology, Software, Formal analysis, Writing - Original draft. **S. G. Nayak:** Supervision.

Declaration of Competing Interest

The authors hereby confirm that the submitted manuscript is an original work and has not been published so far, is not under consideration for publication by any other journal and will not be submitted to any other journal until the decision will be made by this journal. All authors have approved the manuscript and agree with its submission to "Iranian Journal of Electrical and Electronic Engineering".

References

[1] N. S. Kalyan Chakravarthy, O. Vignesh, and J. N. Swaminathan, "High speed and low power buffer based parallel multiplier for computer arithmetic," *Lecture Notes in Networks and Systems*, Vol. 145, pp. 407–416, 2021.

[2] Y. Sri Lakshmi and T. Vigneswaran, "Design and implementation of high-speed Vedic multiplier using hybrid full adder," *International Journal of Advanced Science & Technology*, Vol. 29, No. 3, pp. 10663–10669, 2020.

[3] Ansiya Eshack and S. Krishnakumar, "Reversible logic in pipelined low power Vedic multiplier," *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 16, No. 3, pp.1265–1272, 2019.

[4] M. N. Chandrashekara and S. Rohith, "Design of 8 bit Vedic multiplier using Urdhva Tiryagbhyam sutra with modified carry save adder," in *IEEE 4th International Conference on Recent Trends on Electronics, Information, Communication, and Technology*, pp. 116–120, 2019.

[5] A. Mehra, V. Verma, R. Majumdar, N. Gaur, A. Kumar, A. Awasthi, and C. Pandey, "Comparative analysis of different Vedic algorithms for 8×8 binary multipliers," *International Journal of Industrial and Systems Engineering*, Vol. 33, No. 2, pp. 129–140, 2019.

[6] C. V. S. Chaitanya, C. Sundaresan, P. R. Venkateswaran, K. Prasad, and V. S. Ramakrishna, "Design of high-speed multiplier architecture based on Vedic mathematics," *International Journal of Engineering and Technology*, Vol. 7, pp.105–108, 2018.

[7] M. Pradhan and R. Panda, "High speed multiplier using Nikhilam Sutra algorithm of Vedic mathematics," *International Journal of Electronics*, Vol. 101, No. 3, pp. 300–307, Mar. 2014.

[8] P. Saha, A. Banerjee, P. Bhattacharyya, and A. Dandapat, "High speed ASIC design of complex multiplier using Vedic mathematics," in *Proceedings of the IEEE Students' Technology Symposium*, pp. 237–241, Jan. 2011.

[9] M. D. Belal Rashid, B. S. Balaji, and M. B. Anandaraju, "VLSI design and implementation of binary multiplier based on Urdhva Tiryagbhyam Sutra with reduced delay and area," *International Journal of Engineering Research and Technology*, Vol. 6, No. 2, pp. 269–278, 2013.

[10] P. Saha, A. Banerjee, and Dandapat, and P. Bhattacharyya, "ASIC design of a high-speed low power circuit for factorial calculation using ancient Vedic mathematics," *Microelectronics Journal*, Vol. 42, No. 12, pp. 1343–1352, Dec. 2011.

[11] S. B. K. Tirthaji and V. S Agrawala, *Vedic mathematics*. 13th Edition, Motilal Banarsidass, 2010.

[12] A. Savji and S. Oza, "Design and implementation of Vedic multiplier," *International Journal of Recent Technology and Engineering*, Vol. 8, No. 6, Mar. 2020.

[13] K. Pichhode, M. D. Patil, D. Shah, and B. C. Rohit, "FPGA implementation of efficient Vedic multiplier," in *IEEE International Conference on Information Processing*, pp. 565–570, 2015.

[14] S. K. Panda, R. Das, and T. R. Sahoo, "VLSI implementation of Vedic multiplier using Urdhva-Tiryagbhyam sutra in VHDL environment: A novelty," *IOSR Journal of VLSI and Signal Processing*, Vol. 5, No: 1, pp. 17–24, 2015.

[15] Y. Bansal and C. Madhu, "A novel high-speed approach for 16×16 Vedic multiplication with compressor adders," *Computers and Electrical Engineering*, Vol. 49, No. C, pp 39–49, Jan. 2016.

[16] R. Pushpangadan, V. Sukumaran, R. Innocent, D. Sasikumar and V. Sundar, "High speed Vedic multiplier for digital signal processors," *IETE Journal of Research*, Vol. 55, No. 6, pp. 282–286, Nov. 2009.

[17] D K. Basha, P. Prakash, D. M. K. Chaitanya, and K. A. Manjusha, "RCA-CSA adder based Vedic multiplier," *International Journal of Applied Engineering Research*, Vol. 12, No. 18, pp. 7603–7613, 2017.

[18] K. Sethia and R. Panda, "Multiplier less high-speed squaring circuit for binary numbers," *International Journal of Electronics*, Vol. 102, No. 3, pp. 433–443, Mar. 2015.

[19] M. Shoba and R. Nakkeeran, "Energy and area efficient hierarchy multiplier architecture based on Vedic mathematics and GDI logic," *Engineering Science and Technology, an International Journal*, Vol. 20, No.1, pp. 321–331, 2017.

[20] S. R. Sahu, B. K. Bhoi, and M. Pradhan, "Fast signed multiplier using Vedic Nikhilam algorithm," *IET Circuits, Devices & Systems*, Vol. 14, No. 8, pp. 1160–1166, 2020.

[21] A. Deepa and C. N. Marimuthu, "Design of a high speed Vedic multiplier and square architecture based on Yavadunam Sutra," *Sadhana - Academy Proceedings in Engineering Sciences*, Vol. 44, No. 9, pp. 1–10, 2019.

[22] J. Monteiro, S. Devadas, and A. Ghosh, "Sequential logic optimization for low power using input-disabling precomputation architectures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 17, No. 3, Mar. 1998.

T. Mendez received her B.Tech. degree in Electronics and Instrumentation Engineering in 2007 from Mahatma Gandhi University, Kerala, and her M.Tech. degree in VLSI and Embedded Systems from CUSAT, Kerala in 2011. She has 7 years of teaching and 2 years of industry experience. She is currently working as a Research Scholar with the Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal. Her research interests include digital system design, low-power VLSI design, electronic design automation, and VLSI architecture design.

S. G. Nayak received his Bachelor's degree in Electronics and Communication Engineering, Master's degree in Biomedical Engineering, and Ph.D. degree in Electrical and Electronics Engineering. He is presently employed as Head and Professor of the Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal. His research interests include processor architecture, digital systems design and applications. Additional information is accessible on his homepage: https://manipal.edu/mit/department-faculty/faculty-list/g-subramanya-nayak.html.