

Vedic-Based Squarers with High Performance

Fatemah K. Al-Assfor¹, Israa S. Al-Furati², Abdulmuttalib T. Rashid³

^{1,2}Department of Computer Engineering, University of Basrah, Iraq

³Department of Electrical Engineering, University of Basrah, Iraq

Article Info

Article history:

Received Aug 23, 2020

Revised Feb 24, 2021

Accepted Feb 26, 2021

Keywords:

Vedic Based Squarer (VBS)

Vedic multiplier (VM)

IBK-CSLA

Pipelining

FPGA

ABSTRACT

Squaring operation represents a vital operation in various applications involving image processing, rectangular to polar coordinate conversion, and many other applications. For its importance, a novel design for a 6-bit squarer basing on the Vedic multiplier (VM) is offered in this work. The squarer design utilizes dedicated 3-bit squarer modules, a (3*3) VM, and an improved Brent-Kung Carry-Select Adder (IBK-CSLA) with the amended design of XOR gate to perform fast partial-products addition. The 6-bit squarer circuit can readily be expanded for larger sizes such as 12-bit and 24-bit numbers which are useful for squaring the mantissa part of 32-bit floating-point numbers. The paper also offers three architectures for 24-bit squarer using pipelining concept used in various stages. All these squaring circuits are designed in VHDL and implemented by Xilinx ISE13.2 and FPGA. The synthesis results reveal that the offered 6-bit, 12-bit, and 24-bit squarer circuits introduce eminent outcomes in terms of delay and area when utilizing IBK-CSLA with amended XOR gate. Also, it is found that the three architectures of 24-bit squarer present dissimilar delay and area, and the architecture design based on 3-bit squarer modules with (3*3) VM introduces the lowest area and delay.

Copyright © 2021 Institute of Advanced Engineering and Science.

All rights reserved.

Corresponding Author:

Fatemah K. Al-Assfor,
Department of Computer Engineering,
The University of Basrah,
Qarmat Ali District, Basrah City, Iraq.
Email: fatimah.k.alassfor@gmail.com

1. INTRODUCTION

The squaring operation represents one of the most arithmetic operations in high-speed applications involved cryptography, animation, image compression, fast-Fourier transform (FTT), pattern recognition, adaptive filtering, and others [1-3]. Although squaring function is executed by utilizing multipliers such as Booth, Wallace tree, Dadda, or Vedic multiplier, commonly, performing squaring function by multiplication needs recurring addition operations that incur consuming the computing-time and unnecessarily increasing the area of overall design [4-6]. Thus, utilizing dedicated squarer circuits can help improve the speed of various applications and enhance the general area of the structure [7].

Carrying-out of squaring operation is based on using the property that there is only one operand to be multiplied by itself [8]. The required partial products can be minimized by excluding redundant bits, thus resulting in less area and higher speed [9, 10].

Many designs and techniques were planned to implement binary squarer. P. S. Kasliwal et al. [11] had proposed a vertical and cross-wise algorithm for squaring n-bit data. The delay of the design had been reduced as the generation of partial products and the addition of the partial derivatives are computed simultaneously. Nevertheless, the design area increased due to four Vedic multipliers (VMs) for squaring an n-bit binary operand. In [12, 13], authors have offered high-speed binary squaring circuits employing Urdhva Tiryagbhyam' Sutra (UTS) techniques. Although the courses' delay has been reduced, thereof the implementation utilized two squarer circuits, which led to an increase the device usage. The squarer circuit

planned in [14] is based on the Peasant Multiplication technique, namely, mediation and duplication, to divide one number by two and multiply the other by 2, but this design conquered more area.

P. Ramanamma [15] has created low-power architectures with less area and delay for squaring and cubing the numbers utilizing the Duplex property of Vedic mathematics to be employed in arithmetic/ logic units. After that, A. Deepa et al. [16] used the Yavadunam algorithm and a technique to reduce the bits for developing a Vedic squaring team to improve the speed, area, and power consumption of the squaring unit. To further enhance the speed and location, authors in [17] had developed a squaring circuit for a 2-bit number using Nikhila Sutra rules to accomplish an 8-bit squarer using the repetitive procedure. The implementation of this squarer involved comparators, adders subtractors, shifters, and other components, and it was improved by about 27% in terms of speed. Additionally, authors in [18] presented two designs for squaring circuits of size 8-bit and 16-bit, respectively. The first one is based on Antyayordashakepi Sutra, while the other uses the Duplex method. Both designs have given enhanced performance in terms of delay, area, and power consumption compared to the regular Wallace tree multiplier.

In 2020, Shamim Akhter et al. [7], have used a different approach to implement base-2 squaring units by employing an improved VM to enhance the computation speed and reduce the design area. Whereas the authors in [19] have presented efficient squarer to be used for high-speed digital applications. This squarer's architecture is based on utilizing the Yavadunam algorithm that states of squaring a number are equivalent to the addition of the product of the number to be squared by the difference deficiency and the squaring of deficiency.

All previously works in literature had designed squaring circuits for numbers of size 2-bit and its multiples; namely, 4, 8, 16, and 32-bit and had used adders like ripple-carry adder (RCA), fast RCA, and carry-select adder (CSLA) to compute the final squaring result [7, 20]. This work explores a new architecture for a 6-bit Vedic-based squarer (VBS) by employing a dedicated 3-bit squarer module as a primitive block and a 6-bit improved square root Brent-Kung carry select adder (IBK-CSLA), which will use an amended design for the XOR gate with a motive of generating the final result of squaring in minimal delay and lower area. This 6-bit squarer is then used to implement 12-bit and 24-bit VBS circuits which are very important for squaring the Mantissa part (namely, 24-bit) in single-precision binary floating-point (BFP) numbers. Further, the work offers three designs for the 24-bit squarer employing pipelining concept at three various adders complexities to examine the effect of reducing the bit size of addends to reduce the complexity of VBS related adders and thus, improve the computation speed of the VBS and decrease the area of the design. All the proposed squares are designed by VHDL using Xilinx13.2 and implemented using the FPGA Virtex family.

The rest of the paper is systematized as Section 2 introduces the research method and the previously published 3-bit binary VM functionality to derive a 3-bit VBS. Section 3 highlighted the 6-bit, 12-bit designs and the three pipelined 24-bit VBS architectures. Section 4 illustrates the results and analysis of various VBS. Eventually, section 6 concluded.

2. RESEARCH METHOD

Delay and area are the two prime factors considered when designing most digital systems. These systems heavily rely on adders, multipliers, and squarers [21-23]. The current multipliers and squarers (precisely their related adders), which form the central part of these systems, affect their speed and area [18]. The more complex squarers and multipliers or their related adders are, the more they influence the rate and size [2, 24].

The prime reason for the delay in squarers and multipliers is the propagation of carrying along the road to the squarer's most significant bit. When a Vedic technique such as UTS is utilized, the pipelined style can add the generated partial products [24].

Thus, it is observed that the VBS, which utilizing the UTS technique for squaring binary numbers, can produce outputs faster than other squarers by decreasing the delay to make the final result. An n -bit VBS circuit comprising UTS permits a higher-order bit squaring to be calculated by breaking-down them to lower-order bits. Functionally, an n -bit VBS is designed by two $(n/2)$ -bit VBS and an $(n/2 * n/2)$ -bit VM with product output shifted to left one position to multiply the output by 2 instead of using two $(n/2 * n/2)$ -bit VM that having identical product outputs. Each of the $(n/2)$ -bit VBS can again be comprised by two $(n/4)$ -bit VBS and an $(n/4 * n/4)$ VM in which its output is shifted to left one position. Hence, each higher bit VBS can be formed using two lower bits VBS circuits and a single lower VM that owning half-size (in bits) of the higher one. This break-down is continued till 2-bit (or a 3-bit) VBS modules are reached as portrayed in Figure 1.

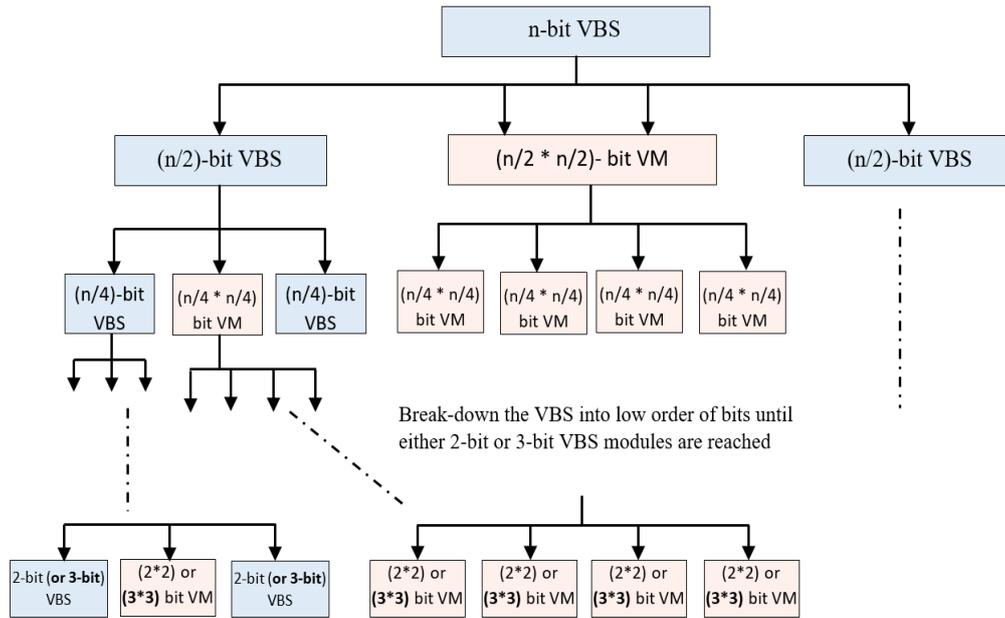


Figure 1. Breaking-down of an n-bit VBS into VBS modules with lower-order bits

2.1. A 3-bit Vedic Based Squaring Operation

A conventional 3-bit VM comprises nine AND gates, three half adders (HA)s, and three full adders (FA)s as depicted in Figure 2 [25]. The multiplier has two (3-bit) inputs $X = X_2X_1X_0$ and $Y = Y_2Y_1Y_0$, respectively and provides a 6-bit product result as $P = P_5P_4P_3P_2P_1P_0$.

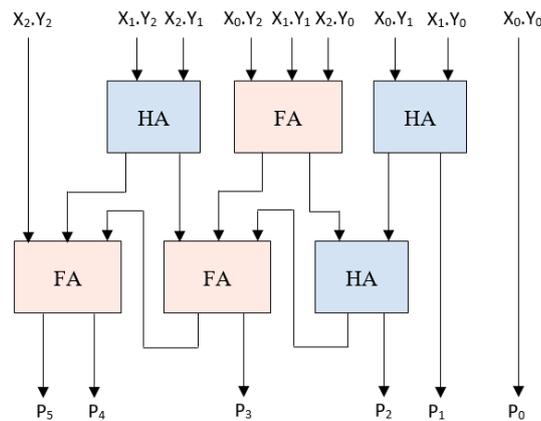


Figure 2. A conventional (3*3)-bit binary VM

This multiplier is used to square a 3-bit input operand utilizing the same input as two input operands ($X=Y$), such that $P = (X.X)$. In this case, the squarer output bits can be computed from the following expressions

- 1-bit output: $P_0 = X_0.Y_0$
- 1-bit output: $P_1 = X_0.X_1 \oplus X_1.X_0 = '0'$
- 1-bit output: $P_2 = X_0.X_1 \oplus X_1$
- 1-bit output: $P_3 = X_0.(X_1 \oplus X_2)$
- 2-bit output: $P_5P_4 = FA(\text{carry, sum})$

The above (3*3) VM can be modified to perform a 3-bit binary squarer using only two HAs and one FA to generate the squaring result as depicted in Figure 3.

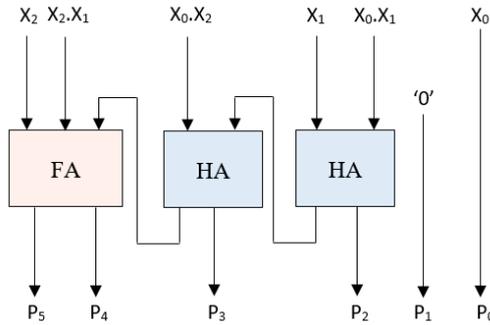


Figure 3. A conventional 3-bit Vedic based squarer (VBS)

3. PROPOSED VEDIC BASED SQUARER CIRCUITS

3.1. Dedicated 3- bit Squarer

To enhance the squarer performance in terms of speed and area, a dedicated 3- bit squarer is offered as illustrated in Figure 4. The dedicated 3- bit squarer eliminates the HA and the FA used in conventional 3- bit VBS explained in Figure 3 and thus, improves the squarer performance. From this, it can be concluded that output P_2 of the HA of conventional 3-bit squarer is eliminated to NOT plus AND gates and the output P_3 is exchanged to XOR gate followed by AND gate instead of two HAs connected serially. The output P_4 is replaced with NOT, OR, and AND gates, and the final squaring bit P_5 is minimized to AND gate only.

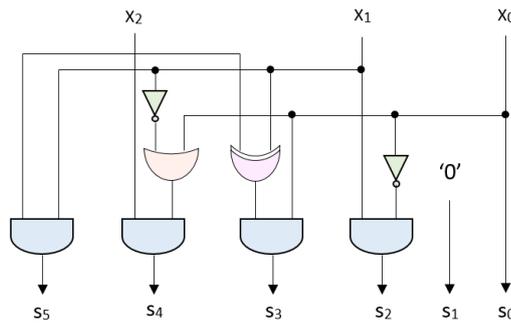


Figure 4. Dedicate 3- bit squarer module

3. 2 Proposed 6- bit Vedic based squarer

The 6-bit VBS circuits can be designed using the dedicated 3-bit squarer and a (3*3) VM. Figure 5 reveals the structure of the offered 6-bit squarer based on the Vedic mechanism. This offered squarer can be straightforwardly extended for squaring input operands having a larger number of bits.

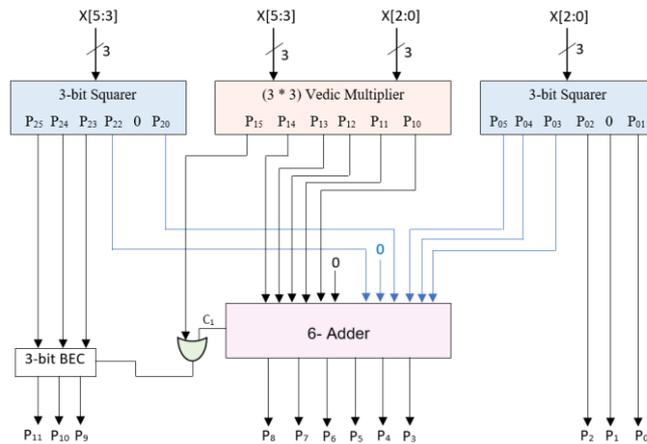


Figure 5. A 6-bit Vedic based squarer

The calculation steps of the proposed 6-bit squarer for an input data $X=(X_5 X_4 X_3 X_2 X_1 X_0)$ can be derived based on the conceptions of (3*3) VM discussed in [25], as follows

$$\begin{array}{r}
 X_2 X_1 X_0 \\
 * X_5 X_4 X_3 \\
 \hline
 (X_5 X_4 X_3) (X_5 X_4 X_3) \qquad (X_5 X_4 X_3) (X_2 X_1 X_0) \qquad (X_2 X_1 X_0) (X_2 X_1 X_0)
 \end{array}$$

3rd Stage
2nd Stage
1st Stage

The terms of the product can be generated by utilizing two 3-bit squarer modules (one for the 1st and the other for the 3rd stages of partial products), and a (3*3) VM for the 2nd stage. Thus, the output of each stage is explained below

- 1st Stage: $(X_2 X_1 X_0)*(X_2 X_1 X_0)$, presents the following 6-bit output: $P_{05} P_{04} P_{03} P_{02} 0 P_{00}$
- 2nd Stage: $(X_2 X_1 X_0)*(X_5 X_4 X_3)$, presents the following 6-bit output: $P_{15} P_{14} P_{13} P_{12} P_{11} P_{10}$
- 3rd Stage: $(X_5 X_4 X_3)*(X_5 X_4 X_3)$, presents the following 6-bit output: $P_{25} P_{24} P_{23} P_{22} 0 P_{20}$

The outputs of the three stages can be organized as follows through using the design principle of the (3*3) VM [26].

| 3 rd Stage | 2 nd Stage | 1 st stage |
|------------------------|---|-----------------------|
| $P_{25} P_{24} P_{23}$ | $P_{15} P_{14} P_{13} P_{12} P_{11} P_{10}$ | $P_{02} 0 P_{00}$ |
| $P_{25} P_{24} P_{23}$ | $P_{22} 0 P_{20} P_{05} P_{04} P_{03}$ | |

As explained in the previous section, instead of using two VM with identical outputs and adding their outputs together, a single VM can be utilized with product output shifted from one position to the left (i.e. multiply the product output by 2 to generate the same result if using two VM. Thus, the output of the 2nd stage can be obtained as

$$\begin{aligned}
 2^{nd} \text{ stage output } P[8:3] &= (P_{15} P_{14} P_{13} P_{12} P_{11} P_{10}) * 2 + (P_{22} 0 P_{20} P_{05} P_{04} P_{03}) \\
 &= (P_{15} P_{14} P_{13} P_{12} P_{11} P_{10} 0) + (P_{22} 0 P_{20} P_{05} P_{04} P_{03})
 \end{aligned}$$

Thus, a 6-bit adder can be used to perform the addition in the 2nd stage. Different types of adders can be employed to perform the addition in the 2nd stage, such as the RCA, regular CSLA, modified CSLA, and others [26]. The 3-bit increment by one (IB1) circuit [27] shown in Figure 5 is used to update the last three output bits of the squarer result. The carry C_1 and the partial product bit P_{15} are OR-ed together to control the IB1 module.

To refine the performance of the proposed 6-bit squarer in terms of area and delay, this work offers the use of a 6-bit improved Brent-Kung adder IBK-CSLA [9, 22] with a 4-bit binary to access-1 convertor (4-bit BEC) to perform the addition of 2nd stage as demonstrated in Figure. 6. The BEC employs less area and generates less delay than the RCA or any other adder. Further, since the two inputs XOR gate represents a significant component in different combinational circuits, especially in binary adders, to generate the sum output, this work achieves the XOR gate by an amended design way to improve the area and speed of squarer architecture. The 6-bit squarer gives better outcomes when using the amended XOR gate precisely when utilized in squarer's related adders (here IBK-CSLA). Figure 7 reveals the amended design of the 2-input XOR gate; it consists of only three basic gates instead of five gates and generates output with the least delay.

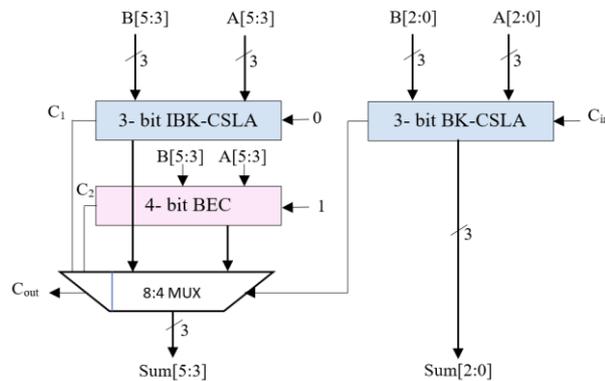


Figure 6. A 6-bit Improved Brent-Kung carry select adder (IBK-CSLA)

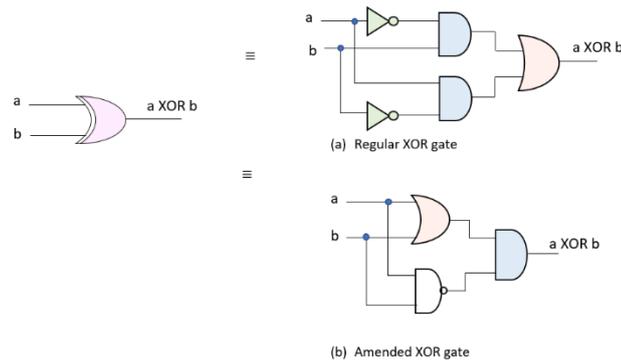


Figure 7 (a) Regular 2- input XOR gate (b) Amended 2- input XOR gate

The proposed 6- bit VBS architecture can be expanded easily for squaring 12- bit and 24- bit numbers which are useful for squaring the mantissa of single-precision binary floating-point numbers.

3.3. Design of 24- bit pipelined VBS circuit

In this section, three architectures with different bit-sized squarers are introduced for 24- bit pipelined VBS as follows:

- a. 24- bit VBS with two 12-bit pipelined VBS and a (12* 12) pipelined VM,
- b. 24-bit VBS with four 6- bit pipelined VBS and six (6*6) pipelined VMs,
- c. 24- bit VBS with eight 3- bit pipelined VBS and twenty-four (3*3) pipelined VMs.

The diagrammatic exemplification of these architectures is given in Figure 8 (a - to - c).

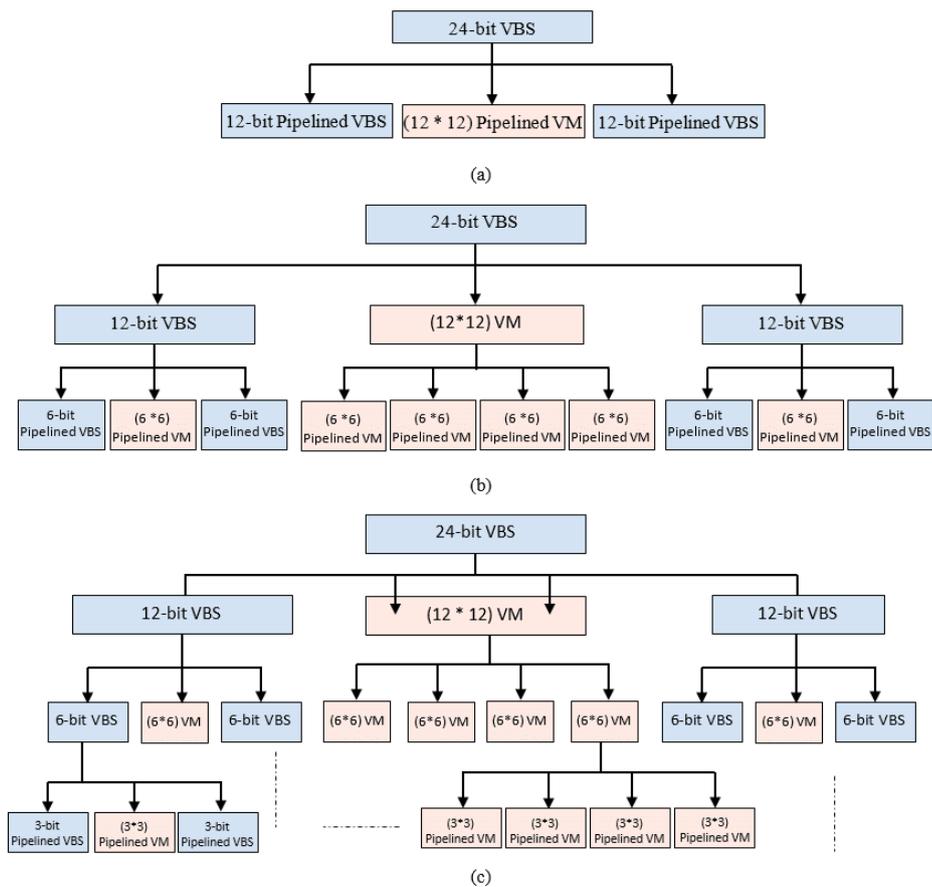


Figure 8. Architectures of 24- bit VBS using (a) Two 12-bit pipelined VBS and one (12* 12) pipelined VM, (b) Four 6- bit pipelined VBS and six (6*6) pipelined VM, (c) Eight 3- bit pipelined VBS and twenty-four (3*3) pipelined VM.

These offered 24-bit squarer architectures utilize the UTS technique in combination with the method of pipelining. The architectures are formed using the pipelining methodology to yield pipelined squarers at different squarer stages. This process is followed to give faster-squaring results for squarers than non-pipelined squarers. It is noticed that as the size of bits of the pipelined squarer becomes low, the complexity of related adders becomes minimal. Further, it can detect that a 24-bit pipelined VBS designed by utilizing the 3-bit pipelined VBS modules and (3*3) VMs gives better results when compared with the other two architectures.

This means that as the size (in bits) of the VBS modules and VM are decreased, their related adders' complexity becomes lesser, ultimately leading to a faster squarer circuit with a lower area.

4. RESULTS AND ANALYSIS

The 3-bit, 6-bit, 12-bit, and 24-bit VBS architectures are designed using VHDL and implemented by Xilinx FPGA Virtex-4 family/ device: XC4VLX15. The 6-bit, 12-bit and 24-bit VBS structures are designed and implemented using the dedicated 3-bit VBS modules and (3*3) VMs with three different adder architectures, namely RCA, conventional CSLA, and the IBK-CSLA using the proposed amended XOR gate, respectively. It is found that VBS circuits implemented utilizing IBK-CSLA with amended XOR gates give the finest outcomes in terms of delay and area in comparison with the other VBS circuits carried out with other adder designs, as depicted in Figures 9 and 10.

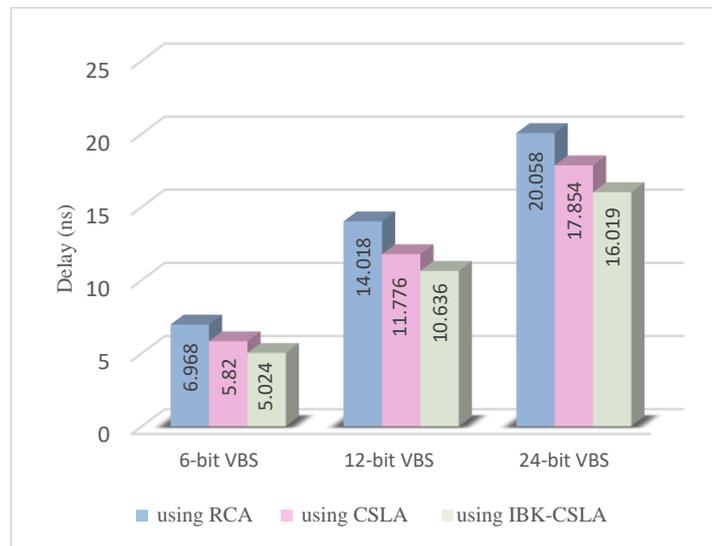


Figure 9. Delay comparative analysis for three different bit-size VBS circuits and by utilizing different adder architectures

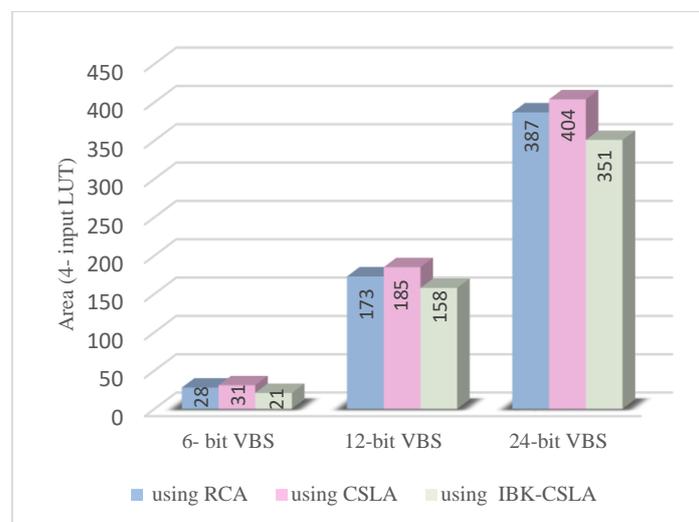


Figure 10. Area comparative analysis for three different bit-size VBS circuits and by utilizing different adder architectures

Further, the three proposed architectures of 24-bit VBS with differing sizes in pipelined VBS bits have been implemented. These architectures include VBS and pipelined VBS at various adder complexity. It is observed that there is a direct relationship between the size in bits for the pipelined VBS and the system throughput in designing the 24-bit VBS and it is seen that as the VBS and the complexity of its related adder are reduced, the area of the VBS reduces, and the delay for computing the squaring result reduces, as well. Figure 11 reveals the uncertainty and location for the three distinct designs of the 24-bit pipelined VBS architectures.

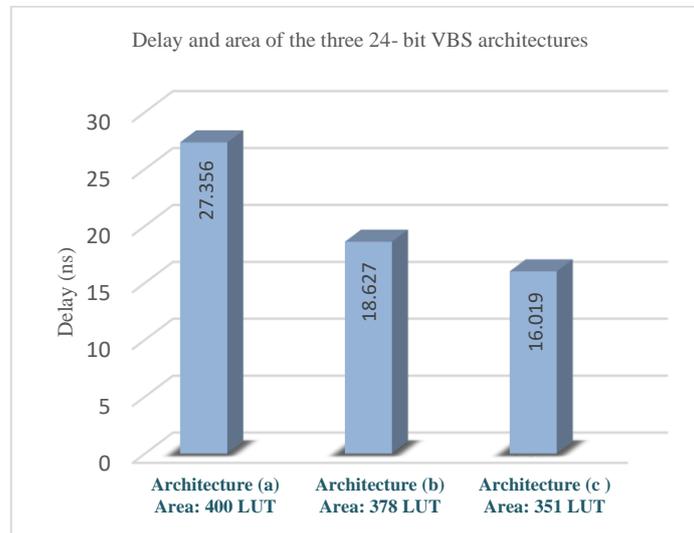


Figure 11. Delay and area for the three architectures of the pipelined 24-bit VBS

It can be observed that the architecture (a) with 12-bit pipelined VBS and a (12*12) VM employs the most significant number of 4-input LUTs (namely 400 LUTs). The use of LUTs is decreased with the level of pipelining reduction, and the squarer circuit utilizing 3-bit pipelined VBS modules and (3*3) pipelined VM has the lowest number of LUTs. The reason behind the increased throughput of the squarer architectures with the lower in bit size of pipelined VBS is that as bit size decreases, the related adder complexity is also reduced. The length of additions and addends decreases, which steers to a reduction in the time to yield the squarer output. It is found that the delay of the design 24-bit squarer utilizing 3-bit pipelined VBS and (3*3) pipelined VM produces the best outcome in terms of speed and hardware employment of FPGA.

All previous works presented in the literature had designed square meters of bit size multiplying 2, such as 4, 8, and 16-bit. Still, none of the pieces has designed squarers of multiple three such as 6, 12, and 24-bit, although they are essential for developing 32-bit floating point squarer units. Table 1 lists the area and delay of the proposed squarers and the previously squarers implemented by the FPGA Virtex family.

Table 1. Performance comparison for different VBS circuits

| Input Size | [7] | | [19] | | [18] | Proposed | | |
|------------|-------|-------|-------|-------|--------|----------|--------|--------|
| | 4-bit | 8-bit | 4-bit | 8-bit | 16-bit | 6-bit | 12bit | 24-bit |
| Delay (ns) | 4.99 | 10.32 | 15.67 | NR | 12.778 | 5.024 | 10.636 | 16.019 |
| Area (LUT) | 6 | 56 | 7 | NR | 261 | 21 | 158 | 351 |

4. CONCLUSION

The work offered high-performance VBS circuits with various sizes in bits and three different adder architectures. The implementation results reveal that the implemented VBS circuits present the finest results in terms of delay and area when using dedicated 3-bit VBS modules and (3*3) VMs with and utilizing IBK-CSLA with amended XOR gate. Further, three architectures for the 24-bit VBS are proposed and implemented. It is found that the implementation of the 24-bit VBS architecture utilizing the pipelined 3-bit VBS and (3*3) VMs provides the finest results in terms of area and delay. This evidences that as the bit size of pipelined VBS reduces, the VBS complexity reduces, and consequently this increases the speed of computation and minimizes the area of design.

REFERENCES

- [1] A. Deepa, C. N. Marimuthu, and C. Murugesan, "An efficient high speed squaring and multiplier architecture using Yavadunam Sutra and bit reduction technique," *Journal of Physics: Conference Series*, vol. 1432, no. 1, 2020.

- [2] 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, 2019.
- [3] R. Sharma, M. Kaur, and G. Singh, "Design and FPGA implementation of optimized 32-bit Vedic multiplier and square architectures," *2015 International Conference on Industrial Instrumentation and Control, ICIC*, pp. 960–964, March 2015.
- [4] M. Munawar, T. Khan, and M. Rehman, "Low power and high speed Dadda multiplier using carry select adder with binary to excess-1 converter," *2020 International Conference on Emerging Trends in Smart Technologies, ICETST*, Pakistan, April 2020.
- [5] A. Jain, S. Bansal, S. Khan, S. Akhter, and S. Chaturvedi, "Implementation of an efficient $N \times N$ multiplier based on Vedic mathematics and Booth-Wallace Tree multiplier," *19th International Conference on Power Electronics, Control and Automation, ICPECA- Proceedings*, pp. 23–27, 2019.
- [6] A. Deshpande and J. Draper, "Squaring units and a comparison with multipliers," *Midwest Symposium on Circuits and Systems*, pp. 1266–1269, 2010.
- [7] S. Akhter, S. Chaturvedi, and S. Khan, "A distinctive approach for vedic-based squaring circuit," *7th International Conference on Signal Process. Integrated Networks*, Spain, vol. 2, pp. 27–30, 2020.
- [8] S. H. G. O. P. Asha and A. Y. T. Arranum, "Design and implementation of Vedic Sutras for square and cube architecture on FPGA," *International Journal of VLSI System Design and Communication Systems*, vol. 04, no. 11, pp. 1222–1225, 2016.
- [9] D. Yaswanth, S. Nagaraj, and R. V. Vijeth, "Design and analysis of high speed and low area vedic multiplier using carry select adder," *International Conference on Emerging Trends in Information Technology and Engineering, ic-ETITE*, pp. 8–12, 2020.
- [10] A. Deepa and C. N. Marimuthu, "Squaring using Vedic mathematics and its architectures : a survey," *International Journal of Intellectual Advancements and Research in Engineering Computations*, vol. 6 no. 1, 2018.
- [11] P. S. Kasliwal, B. P. Patil, and D. K. Gautam, "Performance evaluation of squaring operation by vedic mathematics," *IETE Journal of Research*, vol. 57, no. 1, pp. 39–41, 2011.
- [12] K. Sethi and R. Panda, "An improved squaring circuit for binary numbers," *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 2, pp. 111–116, 2012.
- [13] A. Kumar and D. Kumar, "Hardware implementation of $16 * 16$ bit multiplier and square using Vedic mathematics," *International Conference on Signal, Image and Video Processing (ICSIVP)*, pp.309- 314, Jan 2012.
- [14] G. G. Kumar, C. V. Sudhakar, and M. N. Babu, "Design of high speed Vedic square by using Vedic multiplication techniques," *International Journal of Scientific & Engineering Research*, vol. 4, no. 1, pp. 1–4, 2013.
- [15] P. Ramanamma, "Low power square and cube architectures using Vedic Sutras," *International Journal of Engineering Research and General Science*, vol. 5, no. 3, pp. 241–248, 2017.
- [16] A. Deepa and C. N. Marimuthu, "High speed VLSI architecture for squaring binary numbers using Yavadunam Sutra and bit reduction technique," *International Journal of Applied Engineering Research*, vol. 13, no. 6, pp. 4471–4474, 2018.
- [17] S. Nithyashree and Y. Chandu, "Design of an efficient vedic binary squaring circuit," *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT 2018 - Proceedings*, pp. 362–366, 2018.
- [18] S. Barve, S. Raveendran, C. Korde, T. Panigrahi, Y. B. Nithin Kumar, and M. H. Vasantha, "FPGA implementation of square and cube architecture using vedic mathematics," *Proceedings - 2018 IEEE 4th International Symposium on Smart Electronic Systems, iSES*, pp. 6–10, 2018.
- [19] A. Deepa and C.N. Marimuthu, "VLSI Design of a Squaring Architecture Based on Yavadunam Sutra of Vedic Mathematics," *Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 1162–1167, Aug 2020.
- [20] B. Koyada, N. Meghana, M. O. Jaleel, and P. R. Jeripotula, "A comparative study on adders," *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET*, vol. 2018-January, no. 0, pp. 2206–2230, 2018.
- [21] B. S. Kandula, P. V. Kalluru, and S. P. Inty, "Design of area efficient VLSI architecture for carry select adder using logic optimization technique," *Computational Intelligence*, no. May, pp. 1–11, 2020.
- [22] N. U. Kumar, K. B. Sindhuri, K. D. Teja, and D. S. Satish, "Implementation and comparison of VLSI architectures of 16- bit carry select adder using Brent Kung adder," *2017 Innovations in Power and Advanced Computing Technologies, i-PACT*, vol., pp. 1–7, Jan 2017.
- [23] D. Yaswanth, S. Nagaraj, and R. V. Vijeth, "Design and analysis of high speed and low area vedic multiplier using carry select adder," *International Conference on Emerging Trends in Information Technology and Engineering, ic-ETITE*, pp. 8–12, 2020.
- [24] A. Eshack and S. Krishnakumar, "Pipelined Vedic multiplier with manifold adder complexity levels," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 3, pp. 2951–2958, 2020.
- [25] C. R. S. Hanuman and J. Kamala, "Hardware Implementation of 24-bit Vedic Multiplier in 32-bit Floating-Point Divider," *2018 4th International Conference on Electrical, Electronics and System Engineering (ICEESE)*, pp. 60–64, 2018.
- [26] K. Golda Hepzibha and C.P. Subha, "A novel implementation of high speed modified Brent Kung carry select adder," *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, Coimbatore, India, Jan 2016.
- [27] S. Akhter and S. Chaturvedi, "Modified binary multiplier circuit based on Vedic mathematics," *2019 6th International Conference on Signal Processing and Integrated Networks, SPIN*, pp. 234–237, 2019.

BIOGRAPHIES OF AUTHORS

Fatemah K. Al-Assfor was born in Iraq. She received the B.S. and MSc. Degrees in Electrical Engineering from University of Basrah at Basrah, Iraq in 1990 and 1995, respectively. She worked as an assistant lecturer at the Department of Electrical Engineering in the same University in the period from 1995 to 1998. Then, worked at the Computer Engineering Department in University of Basrah, Iraq. In 2006 she received the Ph.D. degree in Electrical Engineering in the same University and still working in University of Basrah, Iraq up to now. Her fields of interest include Computer Arithmetic, VLSI Design and DSP applications.



Israa S. Al-Furati was born in Iraq in 1982. She received the B.S. and MSc degrees from Department of Computer Engineering, University of Basrah, Iraq at 2003 and 2008, respectively. She worked as an assistant lecturer at the same Department in 2009 up to now. Her fields of interest are Robotics & Industrial Control and VLSI design. Currently, she is pursuing Ph.D. degree in Electrical Engineering in University of Basrah, since 2016. Her research and thesis center on “Design and Implementation of Multi- robot Formations using the Path following Control”.



Abdulmuttalib T. Rashid was born in Iraq. He received the B.S. degree in Electrical Engineering from Basrah University at Basrah, Iraq in 1986. He received the MSc Degree from the same University at 1992. Worked as Assistant Lecturer, at the Department of Electrical Engineering, University of Omer Al Mukhtar, Libya at 1997 to 2007. Then, at the Department of Electrical Engineering, University of Basrah, Iraq at 2007 up to now. His field of interest is Robotics and Industrial control.