

Design and Implementation of 64-Bit Vedic Multiplier and Square Architectures

CH.Roja¹, K. Chinna Akkaiah²

¹Student at the Department of Electronic and Communication Engineering, QIS College of Engineering and Technology, Ongole

²Assistant Professor in the Department of Electronic and Communication Engineering, QIS College of Engineering and Technology, Ongole.

challa.roja1993@gmail.com

Abstract

This paper presents the design of high speed multiplier and squaring architectures based upon ancient Indian Vedic mathematics sutras. In existing Vedic multiplier architectures, the partial product terms are computed in parallel and then added at the end to get the final result. In this work, all the partial products are adjusted using concatenation operation and are added using single carry save adder instead of two adders at different stages. The high speed Vedic multiplier architecture is then used in the squaring modules. The reduced number of computations in multiplication due to adjusting using concatenation operation and one carry save adder only, the designed multiplier offers significant improvement in speed. The designed architectures are realized using Xilinx Spartan-3E FPGA. The comparison shows the 28.72% and 38.59% reduction in propagation delay for the designed 32-bit multiplier as compared to the existing multiplier designs.

Keywords: VLSI, Vedic, Urdhava Tiryakbhyam, Dwandwa-yoga, Multiplier, Square, IXI.

INTRODUCTION

The word 'Vedic' was driven from the word 'Veda' which is ancient store-house of all knowledge. Vedic mathematics provides the solution to the problem of long computation time by reducing the time delay needed for the operations to be performed. It has originated from "Atharva Vedas" the fourth Veda. Atharva Veda mainly deals with the branches like engineering, mathematics, sculpture, medicines and all other sciences. Vedic mathematics deals with all areas of mathematics either it is pure or applied. It was rediscovered from the Vedas between 1911 and 1918 by Sri Bharati Krishna Tirtha Ji. Vedic mathematics has been formulated on sixteen sutras and thirteen sub-sutras. These sutras offer magical short cut methods to all basic mathematical operations. All the advantages drives from the fact that Vedic mathematics approach is totally different & considered very close to the way a human mind works. Vedic mathematics can be applied to every branch of mathematics including arithmetic, algebra and geometry. The powerful applications of Vedic mathematics are in fields of Digital Signal Processing (DSP), Chip Designing, Discrete Fourier

Transform (DFT), High Speed Low Power VLSI Arithmetic and Algorithms and encryption systems [1]. Arithmetic Logic unit (ALU) is the important unit in processors that performs basic arithmetic operations like addition, subtraction, multiplication and logical operations. Today's processors operate at very high clock speeds. Hence it is imperative to have faster additions, multiplications, Squaring, and cubing etc. operations. Multiplication is the most basic & frequently used operation in ALUs. It is an operation of scaling one number by another. Multiplication operations also form the basis of other complex operations such as squaring, cubing, convolution, fast-Fourier transforms etc. Multiplication is the most time consuming amongst the basic mathematical calculations performed by arithmetic units. With these considerations, it is always important to have fast and efficient mechanism to implement multiplication function. Therefore, circuit designers are looking forward for new algorithms & methods to speed up all the arithmetic operations. Design and implementation of high speed multiplier and square modules. These Vedic mathematics based modules along with other modules can be integrated in an ALU. This Vedic

ALU will be far more efficient than the conventional one.

VEDIC MULTIPLIER ARCHITECTURE

This section introduces multiplication operation using *Vedic IXI Methodology* and then illustrates architecture of 2x2 multiplier modules and finally architecture of n-bit multiplier. *Urdhva-Tiryakbhayam* sutra has been used for multiplication purpose. The fig. 1 explains multiplication of two decimal numbers using IXI technique:

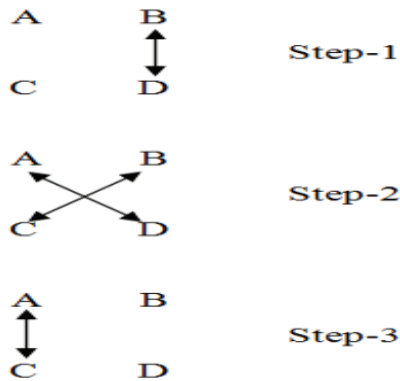


Figure 1: IXI Methodology for Multiplication

In this method initially multiplication of the rightmost digit of multiplier is performed with the rightmost digit of the multiplicand giving the LSB of the product term as shown in step-1 of fig. 1. Then multiplication of the leftmost digit of the multiplicand with the rightmost digit of the multiplier and the rightmost digit of the multiplicand with the leftmost digit of multiplier is performed and then added. Thus forming the middle part of the product term as in step-2 of fig.1. At the last step, in step-3 the leftmost part of the multiplicand is multiplied with the leftmost part of the multiplier forming the leftmost part of the product term. In this way the multiplication process is carried out. Similar logic of cross multiplication and addition can be extended to implement any number of bits. Each iteration gives the coefficient of the final product.

2x2 Multiplier Module

The 2x2 multiplier module can be implemented using four AND Gates and two Half-adder modules and is shown in fig.2. The total delay for 2x2 multiplier is only two half-adder delay.

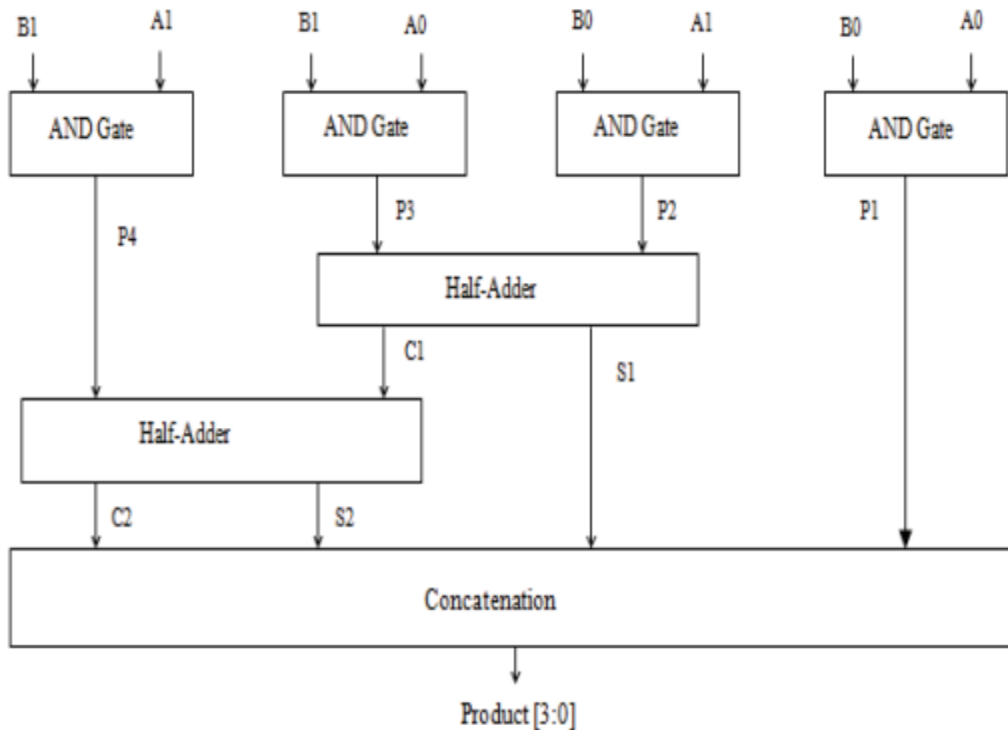


Figure 2: Architecture of 2x2 Multiplier Module

It consists of four AND gates used for AND operation of single bit numbers, two half adders for the addition of numbers obtained in the previous section to provide the sum and carry. At the end concatenation of all the final results is done to obtain the final product of 4-bits. This architecture is basically implementation of IX1 technique. Firstly, right most digits are multiplied to give the LSB of the final product. Then LSB of first digit is multiplied with the MSB of second digit and MSB of first digit with the LSB of the second digit and are added together using the half adder block. The sum obtained from this adder block is the second digit of the product. At the next step MSB's of both the numbers are ANDed together and the result is added with the previous carry through half adder. So, the obtained sum makes the third digit of the final product and the carry will be the MSB of the product.

N-bit Vedic Multiplier Module

The architectural concept of n-bit Vedic Multiplier is shown in fig. 3. This architecture consists of four $n/2$ bit multipliers used for calculating the partial products. Next, the results of these $n/2$ bit multipliers are adjusted using concatenation operation to have all the partial product terms of equal bit-length. The partial product of right most multiplier is concatenated with the partial product of leftmost multiplier and the partial products of middle two multipliers are concatenated with $(n/2)$ zeroes each. All the numbers obtained after concatenations are added together using the single carry save adder. At the end, the sum obtained from the carry save adder is concatenated with the LSB partial product of right most multiplier to get the required final product

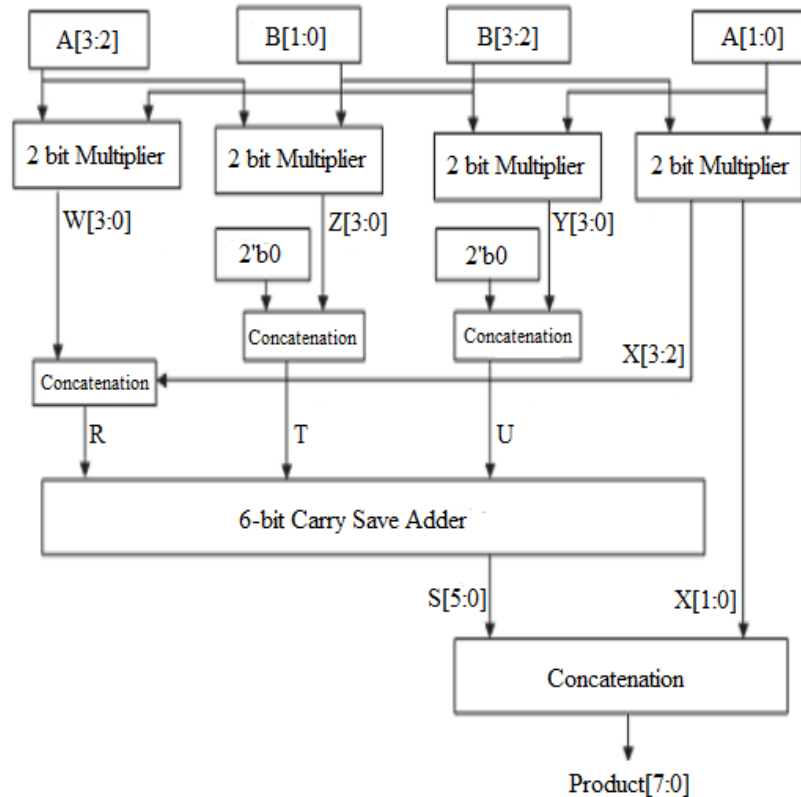


Figure 3(a): Architecture of 4-bit Multiplier

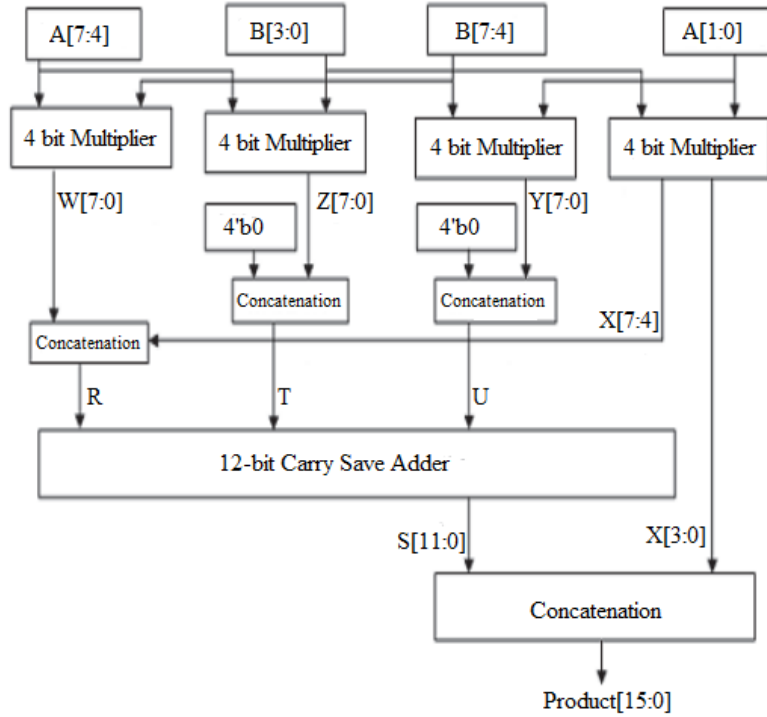


Figure 3(b): Architecture of 8-bit Multiplier

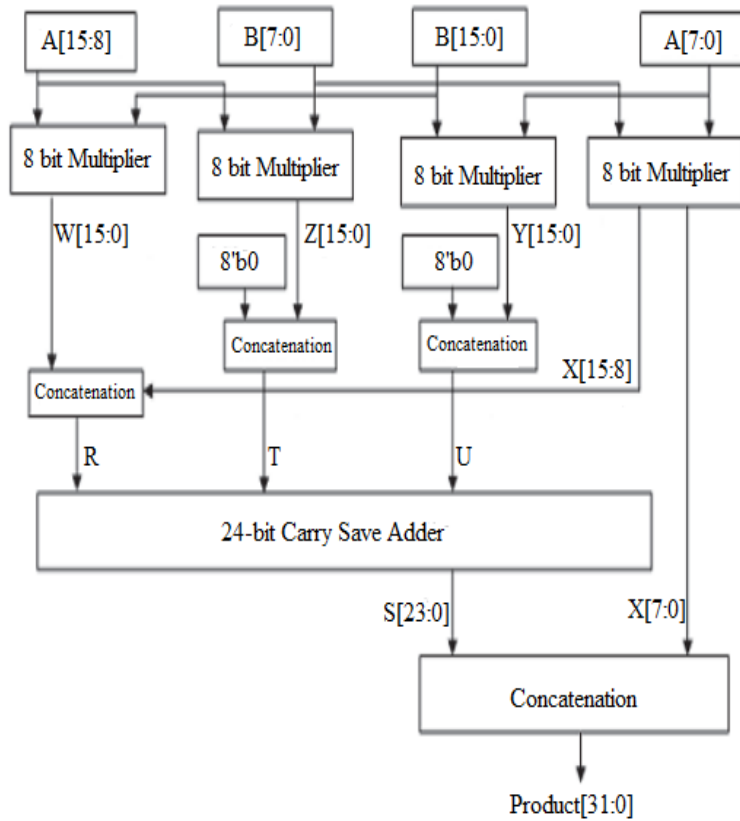


Figure 3(c): Architecture of 16-bit Multiplier

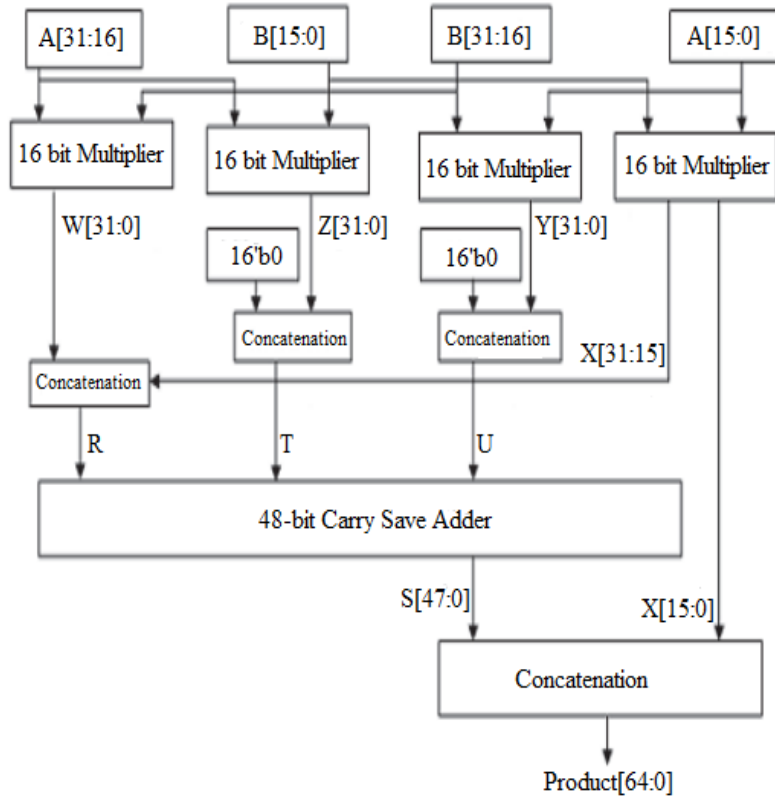


Figure 3(d): Architecture of 32-bit Multiplier

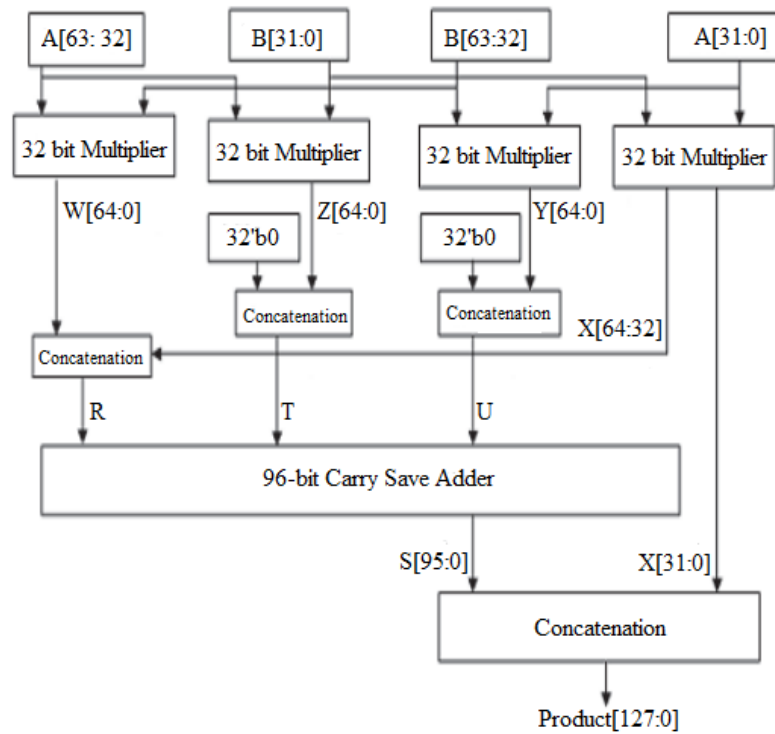


Figure 3(e): Architecture of 64-bit Multiplier

SQUARING ARCHITECTURE

Squaring is also the fundamental arithmetic operation to be performed. For squaring of a number *Urdhava-Tiryakbhyam sutra* along with *Dwandwa-yoga* is used. In this 'Duplex' technique is used. In the Duplex (D), twice the product of the outermost pair is calculated, and then added with twice the product of the next outermost pair, and so on till no pairs are left. When there is odd number of bits in the original sequence, there is one bit left in the middle, and this is entered as its square.

For 1-bit number (X0),

$$D = X0 * X0.$$

For 2-bit number (X0X1),

$$D = 2 * X1 * X0.$$

For 3-bit number (X2X1X0),

$$D = 2 * X2 * X0 + X1 * X1.$$

For 4-bit number (X3X2X1X0),

$$D = 2 * X3 * X0 + 2 * X2 * X1.$$

For finding the square the duplexes are calculated starting from the either side may be left or right and is proceeded to the other direction and vice-versa. At the end duplexes are added to get the final result

2-bit Squaring Circuit

N-bit Squaring Circuit

For calculating the square of 2-bit number one AND gate and one Half-adder are used as shown in fig. 4.

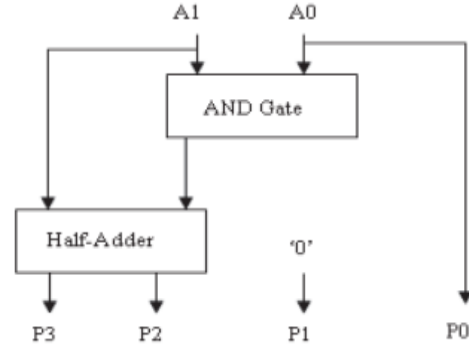
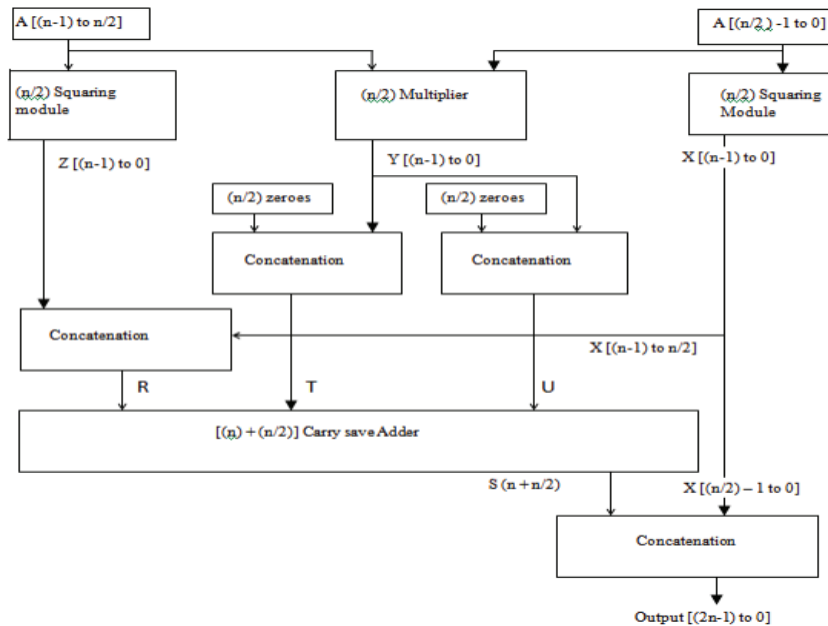


Fig. 4 Architecture of 2-bit Squaring Circuit

The LSB of the input is directly taken as the LSB of the final output. The next bit of the output will always be '0'. So, it is by default set to zero value always. Then both the inputs are ANDed together. The result is added to the MSB of the input numbers. The sum of the half-adder is taken as the third bit of the final result and the carry bit of the half-adder is the MSB of the final output. Thus, we get the final output of 4-bit for squaring module of two-bit number. This two-bit squaring circuit is the base circuit for developing the n-bit squaring module.

The architecture for n-bit squaring module is shown in fig. 5.



In this architecture, one multiplier of $n/2$ bits along with two squaring circuits of $n/2$ bits is used for the calculation of the partial products. At the last step all the calculated partial products are adjusted accordingly and are added together using the single carry save adder to get the final output.

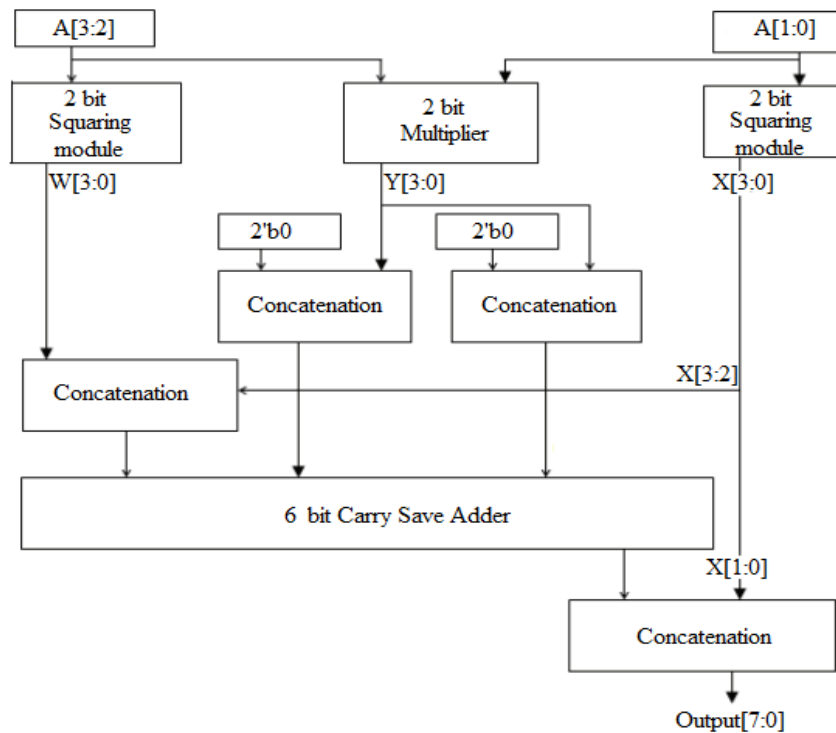


Figure 5(a): 4-bit Squaring Architecture

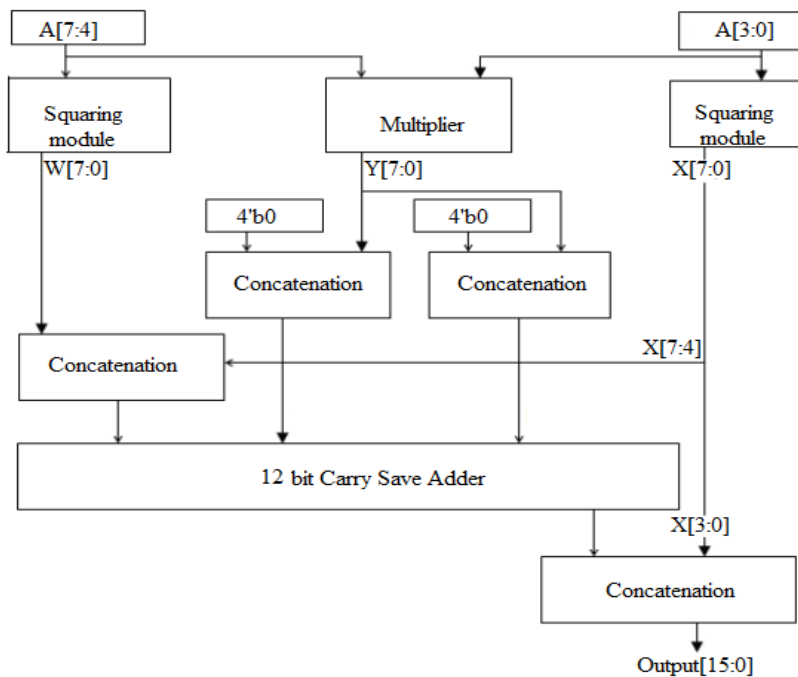


Figure 5(b): 8-bit Squaring Architecture

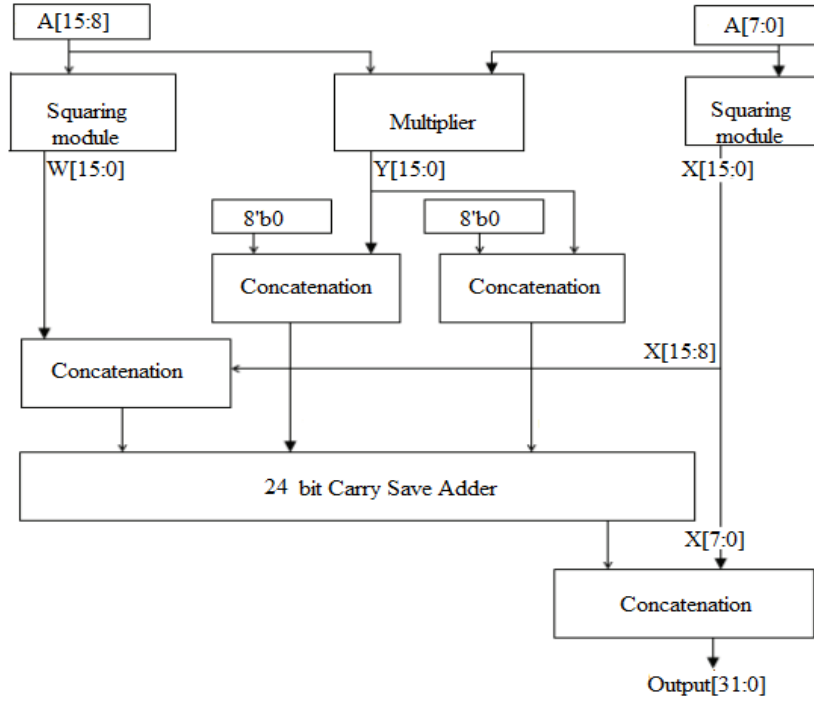


Figure 5(c): 16-bit Squaring Architecture

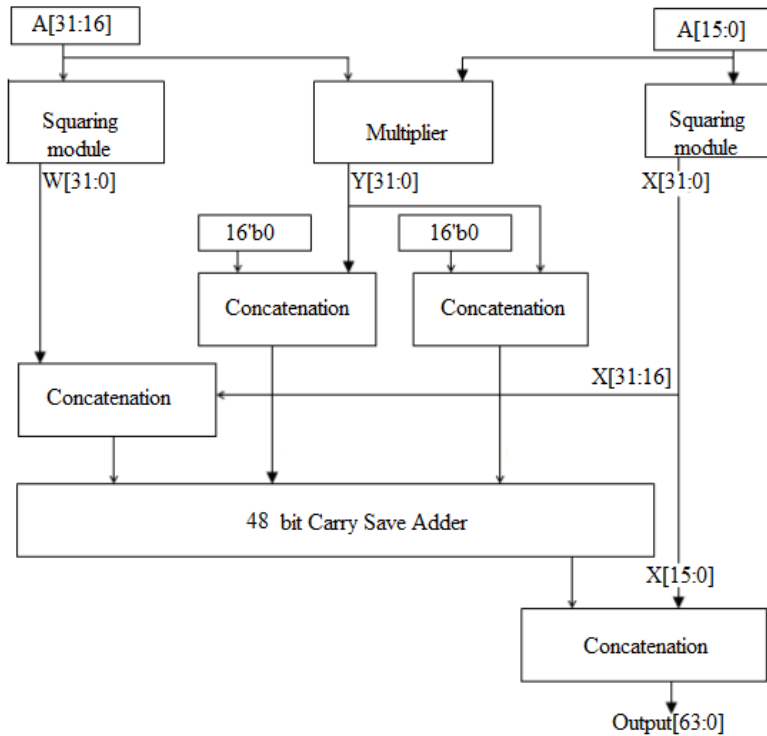


Figure 5(d): 32-bit Squaring Architecture

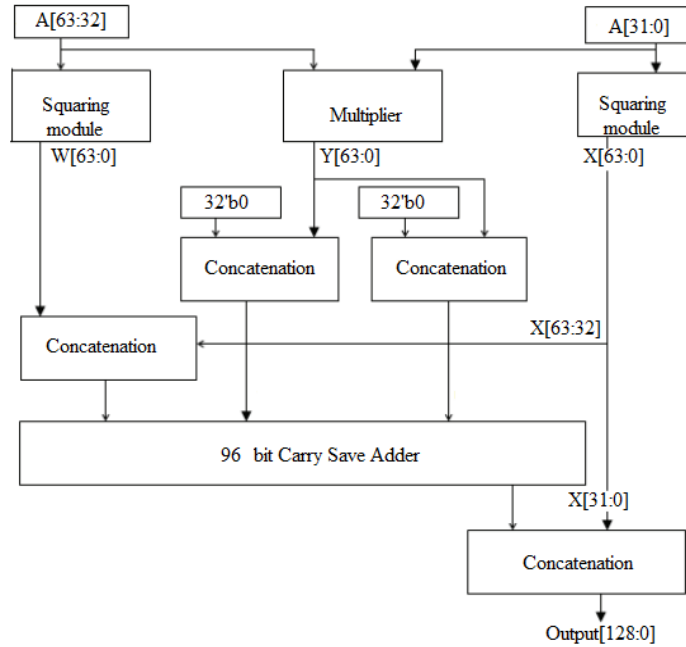


Figure 5(e): 64-bit Squaring Architecture

RESULTS

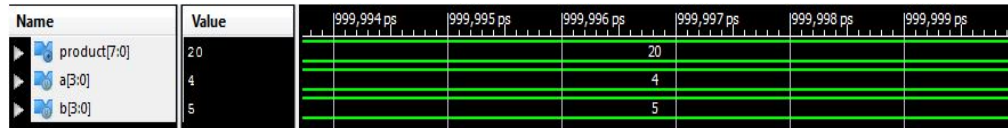


Fig 6.1 4-bit multiplier

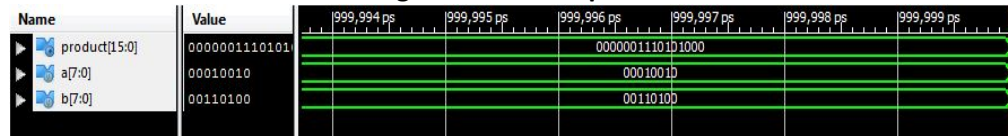


Fig 6.2 8-bit multiplier

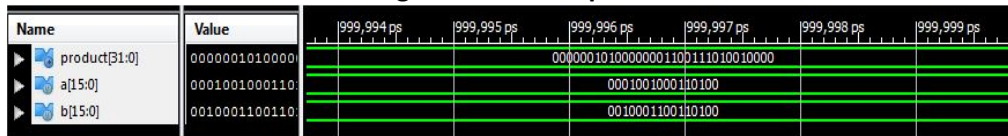


Fig 6.3 16-bit multiplier

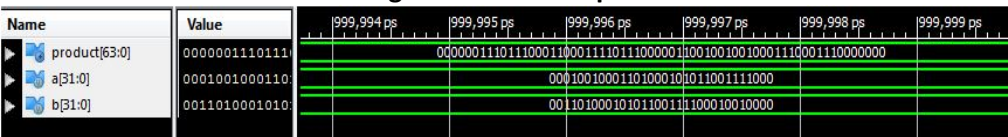


Fig 6.4 32-bit multiplier

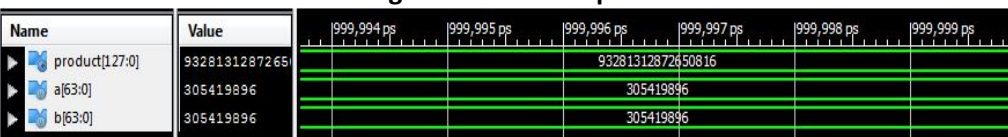




Fig 6.5 64-bit multiplier

- Cubing Circuit using Vedic Mathematics for Finite Field Arithmetic”, International Conference on Advances in Recent Technologies in Communication and Computing, pp.873-875, 2009.
10. Himanshu Thapliyal, Saurabh Kotiyal, M.B Srinivas, “Design and analysis of a novel parallel square and cube architecture based on ancient Indian Vedic Mathematics”, Circuits and Systems, 48th Midwest Symposium, pp. 1462, Vol. 2, 7-10 Aug. 2005.
 11. H.Thapliyal, M.B. Srinivas, “High Speed Efficient N X N Bit Parallel Hierarchical Overlay Multiplier Architecture Based on Ancient Indian Vedic Mathematics”, Transactions on Engineering, Computing And Technology, Vol. 2, December 2004.
 12. Syed Shahzad Hussain Shah, Muhammad Naseem Majoka, Gulistan Raja, “Design and Implementation of 32-bit Vedic Multiplier on FPGA”, First International Conference on Modern Communication & Computing Technologies, 26-28 February, 2014.
 13. Ramalatha M, Thanushkodi k, Deena Dayalan K, Dharani P, “ A Novel Time and Energy Efficient Cubing Circuit using Vedic Mathematics for Finite Field Arithmetic”, IEEE International Conference on Advances in Recent Technologies in Communication and Computing, pp. 873- 875, 2009.
 14. L.Sriraman, K. Saravana Kumar, T.N. Prabakar, “Design and FPGA Implementation of Binary Squarer Using Vedic Mathematics”, IEEE ICCNT, July 4-6, 2013.

Author Details

	<p>K.CHINA AKKIAH B.TECH at Prakasam Engineering College,kandukuru in 2010 M.TECH at NIT Calicut,kerala in 2013 present working as Assistant Professor in the department of Electronic and Communication Engineering, QIS Collage of Engineering and Technology,Ongole</p>
	<p>CH.ROJA B.TECH at ST.ANNS ENGINEERING COLLEGE,CHIRALA in the department of electronics and communication engineering in 2014 pursuing M.TECH in VLSI&EMBEDDED SYSTEMS from QIS COLLEGE OF ENGINEERING AND TECHNOLOGY,ONGOLE.</p>