High Speed VLSI Architecture For Squaring Binary Numbers Using Modified Yavadunam Sutra

[1]CH.PRIYANKA, [2]D.VYSHNAVI, [3]J.MEGHANA, [4]C.JAYA KRISHNA, [5]R.MURALI KRISHNA

[1,2,3,4]B.Tech Students, [5]Associate professor, Department of ECE, Swarnandhra College of Engineering and Technology ,Seetharampuram ,Andhra Pradesh, India.

[1]priyankachodisetti999@gmail.com, [2]vyshnavidevata1998@gmail.com,

[3]meghanajayavarapu2528@gmail.com, [4]jayakrishna3377@gmail.com, [5]muralirapaka@gmail.com

Abstract - The boom of high speed recent communication hardly. requires the efficient mathematical operations. The favoured performance outcomes of any architecture are possible only by the effective mathematical operations. Squaring plays an essential role in high speed applications like animation, digital signal processing, and image processing, etc where the speed is a crucial performance characteristic. In the existing model the squaring operation is algorithm based and if input is greater or equal to 2^n-1 it follows mode1 else mode2, which takes more computing time for squaring operation. This project explains about the modified Yavadunam sutra. This method produces the squared output by using partial product equations with less number of gates as a result it takes less computing time and increases the speed of operation.

Keywords- Squaring, Vedic Mathematics, Yavadunam Sutra, Bit reduction.

I. INTRODUCTION

The emerging trend in Science and Technology results in high speed applications. Digitalization has made us to concentrate on the speed of the devices than the other performance outcomes. In most of the high speed applications squaring operation plays an vital role. Among all the arithmetic operations the multiplication and squaring operations plays the most essential role. To achieve high performance several architectures are adopted. The performance analysis of any circuit is estimated by the two main parameters i.e. area and speed. In many applications multipliers are used to find the squares or the orders of power operations. Hence to obtain the square of a binary number multipliers are used. In most high speed applications, various multipliers like Braun, Wallace tree, Dadda-multiplier; Baugh-Wooley methods of 2's complement, Booth multiplier are used. In the midst of all Booth's algorithm and Recursive decomposition are mostly used. Rather than the conventional multipliers, the Vedic multipliers are also emerging into the trends. The Vedic multipliers based on Vedic sutras, most likely Urdhava Tiryagbhyam and Nikhilam Sutras are commonly in practice. On comparing Vedic multipliers with the conventional Vedic multipliers are fast and require less area. Yavadunam Tavadunikrtya Vargancha Yojayet, an Upa sutra of Yavadunam sutra is a typical sutra for squaring. Vedic Mathematics is one of the old Mathematic method given to us by our ancient saints. This was revived by Swami Bharati Krishna Tirthaji Maharaj. The entire computations were done with the help of anyone of the 16 Sutras or 13 Upa sutras, the word formulae which assist to solve the whole variety of the mathematical problems.

Any sutra or Upa sutra can be put together to solve any hassle and it is not essential for the student to stick to any specific formulation. Moreover they can formulate distinct strategies. As a result Vedic maths is speedy, fascinatory, contemporary, which develops logical thinking and interpretation talents and rational. In Vedic Maths the multiplication operation can be turned around to accomplish one-line division of decimal numbers. Similarly the squaring method can be turned into a one line square root of decimal number .

II. RELATED WORK

To obtain the square of the binary numbers using multipliers like Braun Array, Baugh Wooley methods of 2's complement, Booth multiplier, Wallace trees, Dadda, etc are not efficient in their performance. But at present they ensure to be obsolete due to the raising trends in high speed communication and many techniques are promising for further more faster operations. In Urdhava sutra has been used to perform squaring. Here the delay has been reduced as the computation of partial products and addition of the partial products are performed simultaneously. The area has been increased as 4 Vedic multipliers are used to square a n bit binary number. so proposed a high speed binary squaring architecture using

 DOI : 10.18231/2454-9150.2019.0113

Vertical and crosswise method. The delay has been reduced but the implementation used two squaring circuits thereby resulting in increase in device utilization. The realisation states that it occupies more area. In a 4 bit binary squaring using Dwandayoga sutra is designed. It is quite faster and simpler to compute when compared to the traditional techniques. It has been proved only for 4 bits and as the number of bits increases it becomes complicated. Until now there is no standard architecture available for the squaring sutra Yavadunam. It is difficult to design a particular architecture as the sources of input will be of various ranges. To overcome this problem the same view is extended for the binary numbers. In the previous papers a unique hardware has been designed for a squaring sutra, Yavadunam, the same has been modified by applying bit reduction technique to reduce the delay.

III. DESIGN OF YAVADUNAM SUTRA

Yavadunam Sutra is one of the oldest squaring sutra of Vedic mathematics .It is a accurate method to calculate square of numbers with the specific condition that the number to which is to be squared should be close to the power of 10. The general idea of this sutra is to set up the square of the deficiency. This method to amend the existing squaring architecture in order to achieve the progress in area, speed and power.

To accomplish this weight reduction is made by getting rid of the Most Significant Bit. Then bit reduction Yavadunam Sutra is implemented to obtain the square of the given number. By proposing this modification the number of bits to the squaring architecture is reduced and there the number of components used can be reduced thereby reducing the power and delay. Let us assume squaring of 4 bits binary number A3A2A1A0. Now weight reduction is made and the MSB is removed. Now the input to the architecture becomes A2A1A0, which is a 3bit binary number. Now the limit for the 3 bit number is 8. As the input is 4 bits the output will be 8 bits. If the number to be squared is greater than 2^N-1 and the output will be 6 bits. If the number to be squared is less than 2^N-1 then Squaring of binary numbers involves two modes. One modes is used when the deficiency is positive i.e. if the given input is greater than 2^N-1, the base value. For example if input is 12 the base value is 8 and the deficiency is +4. And the other mode is used when the deficiency is negative i.e. if the given input is less than 2N-1, the base value. For example if number is 4, base value is 8 and the deficiency is 4.

Mode:1 The given number should be greater than 2^N-1 .

Mode:2 The given number should be lesser than 2^N-1 .

Algorithm for Mode: 1

Input: A3A2A1A0

Output: B7…B2B1B0

Step:1) Reduce the weight of the input by removing the MSB, then input becomes A2A1A0 which is the deficiency.

Step:2) Let the deficiency be D2D1D0 = A2A1A0.

Step:3) Square the deficiency value, then the output of squaring=X5…X2X1X0.

Carry=X5X4X3

LHS =X2X1X0 = B2B1B0

Step:4) Add the deficiency and the input.

A3A2A1A0 + D2D1D0 = Y4Y3Y2Y1Y0

Step:5) Add the above output to the carry of the LHS.

Y4Y3Y2Y1Y0 + X5X4X3 = B7B6B5B4 B3 =RHS

Step:6) Concatenating LHS and RHS ,the output is

B7B6B5B4B3B2B1B0 =square of A3A2A1A0.

Algorithm for Mode: 2

Input: A3A2A1A0

Output: B5…B2B1B0

Step:1) Reduce the weight of the input by removing the MSB, then input becomes A2A1A0.

Step:2) Take two's complement of A2A1A0 ,the output is the deficiency. Let the deficiency be

D2D1D0 = A2A1A0.

Step:3) Square the deficiency value, then the output of squaring=X5…X2X1X0.

Carry = X5X4 X3

LHS =X2X1X0 = B2B1B0

Step:4) Now subtract the deficiency from the bit reduced number.

A2A1A0 - D2D1D0 = Y2Y1Y0

Step:5) If the subtractor output is positive then add the above output to the carry X5X4 X3. Y2Y1Y0 + X5X4X3 = B5B4 B3

Step:6) If the above subtractor output is negative then subtract the output Y2Y1Y0 from the carry X5X4X3.

i.e. X5X4X3 - Y2Y1Y0 = B5B4 B3 =RHS

Step:7) Concatenating LHS and RHS, the output is

B5B4B3B2B1B0 =square of A3A2A1A0.

Both the modes are joined together into a single architecture. The number greater than or lesser than 2^N-1 can be identified from the MSB position. The control signal to the multiplexer depends on the MSB of the input i.eAN-1. If the MSB of the number is 0 then the number

 DOI : 10.18231/2454-9150.2019.0113

should be 2's complemented after bit reduction and then squared. If the MSB of the number is 1 then the number should be directly subject to bit reduction and squared. The control signal to select the adder or subtractor depends on the MSB of the input i.e.N-1.When the number is greater than 2^N-1 , AN-1 will be 1 and adder to be selected and the output of the adder should be added with the carry. In other case if the number is less than 2^N-1, AN-1 will be 0 and the subtractor should be selected. Similarly the control signal which select the carry adder or the subtractor depends upon the output Y N-1. If Y N-1 is 0 then carry should be added or the subtractor should be selected.

IV. PROPOSED METHOD

```
                A3   A2   A1   A0
                      *
                A3   A2   A1   A0
-------------------------------------------
                     P3   P2   P1   P0
                P7   P6   P5   P4    X
           P11  P10  P9   P8   X    X
      P15  P14  P13  P12  X    X    X
-------------------------------------------
  Y7  Y6   Y5   Y4   Y3   Y2   Y1   Y0
```

For every 4bit squaring we will get 16 partial products respectively with the help of the partial products we get the output. By default A0 is asigned to Y0,Y1 is assigned as 0 directly because in every 4 bit squaring operation the output of the 2nd bit will be zero, Y2 can be obtained by the ex-or operation of P1 and P5, similarly Y3 by the ex-or operation of P1 and P2, Y4, Y5, Y6, Y7 equations are as shown below.

These equations are applicable for every 4bit binary number

Input=A3A2A1A0 Output:Y7Y6Y5Y4Y3Y2Y1Y0

In order to get the output in the proposed method we have to follow these equations

Y0=A[0]; Y1=1'b0; Y2=P1^P5; Y3=P1^p2;

Y4=(((P10^P6)| P2)^P12); Y5=P9^(A3&(P13^P8));

Y6=((P15^P14)|P13); Y7= P14;

For example let us consider (1100)=12 binary squaring by using the above equations we get as follows:

Y0=A[0]=0;

Y1=1'b0=0;

Y2= P1^P5 =0^0=0;

Y3= P1^p2 =0^0=0;

Y4=(((P10^P6)| P2)^P12)=(((1^0)| 0)^0)=1;

Y5=P9^(A3&(P13^P8))=0^(1&(0^0))=0;

Y6=((P15^P14)|P13)=((1^1)|0)=0;

Y7= P14=1;

Thus we get 10010000=144 which is the square of the given input.

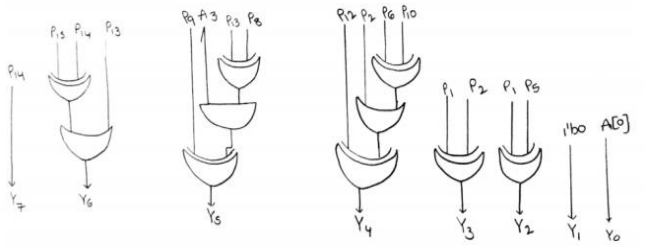

Fig 1:Logic diagram

This method is helpful in reducing the complexity in the previous method where we follow two modes if greater are equal to 8 we follow mode1 else we follow mode 2 but in the proposed there is no such a complexity whatever maybe the 4bit binary number we follow the single process.

V. RESULT

The implementation of the proposed squaring architecture for binary numbers is designed. The simulated output of a 4bit squaring architecture is shown in the figure 2 and figure 3 depicts the RTL schematic of the squaring architecture.

Fig 2:Simulation output

 DOI : 10.18231/2454-9150.2019.0113

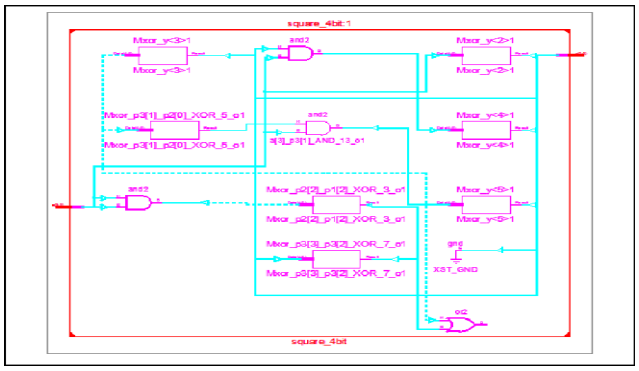

Fig 3: RTL Schematic

Device Utilization	Existing Architecture	Proposed Architecture
Product version	ISE 14.5	ISE 14.5
No. of slice LUTs	15032	15032
No. of fully used LUT-FF pairs	6	6
No. of bonded IOBs	226	226
Delay report	17.2ns	5.5ns
Power report	59.02mw	28.64mw

Table 1:Comparision table

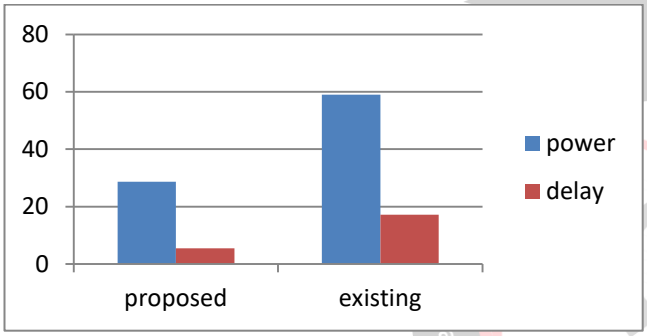

Fig 4: Graph representation

The project has been implemented on xilinix chip version ISE 14.5 using HDL language. We have observed that as compared to the previous squaring architecture synthesis the proposed architecture synthesis is having high performance in means of power consumption is decreased and delay is decreased.

VI. CONCLUSION AND FUTURE SCOPE

The new high speed VLSI architecture for squaring binary numbers using yavadunam sutra is better as compared to the existing Yavadunam algorithm. The speed of proposed squaring algorithm is improved, thereby reducing the delay and number of components. On the whole this proposed architecture optimizes the delay and the speed.

In future the new high speed VLSI architecture for squaring binary numbers using yavadunam sutra can be developed for the n bit binary squaring.

REFERENCES

[1].Arushi Somani, Dheeraj Jain, Sanjay Jaiswal, Kumkum Verma and Swati Kasht, “Compare Vedic Multipliers with Conventional Hierarchical array of array multiplier”, International Journal of Computer Technology and Electronics Engineering (IJCTEE) Vol.2, No. 6, (2012),.

[2].Bengali S.,“Vedic Mathematics and Its Application Computer-Arithmetic”,http://repository.lib.ncsu.edu /ir/bitstream/1840.16/7232/1/etd.pdf, (2011).

[3].Deepa A. and Marimuthu C N., ”A High Speed VLSI Architecture of a Pipelined Reed Solomon Encoder for Data Storage in Communication Systems”, Asian Journal of Research in Social Science and Humanities,Vol.7,No.2,pp.228-238,(2017),.

[4].Dilli Kumar B, Bharati M, “A High Speed and Efficient Design for Binary Number Squaring using Dwandwa Yoga”, International Journal of Advanced Research in Computer Engineering & Technology Vol. 1,No.4, (2012).

[5].Ghosh M, “Design and Implementation of Different Multipliers-VHDL”, http://ethesis.nitrkl.ac.in/66/1/ moumita.pdf, (2017).

[6].Jagadguru Swami Sri Bharati Krishna Tirthaji Maharaj,“Vedic Mathematics”, Delhi: Motilal Banarsidas Publishers Pvt.Ltd. , (2009).

[7].Kabiraj Sethi and Rutuparna Panda, “An improved Squaring circuit for binary numbers”, International Journal of Advanced Computer science and Applications, Vol.03, No.02, (2012).

[8].Pabitra Kumar Mohapatra, Siba Kumar Panda, Sambita Dalal, Shibashis Pradhan,”VHDL implementation of a Novel Low power squaring circuit using YTVY algorithm of Vedic Mathematics”, International Journal of Emerging Trends in Science and Technology, Vol.02, No.3, pp.21392146, (2015).

[9].Prabha S.Kasliwal, B.P.Patil and D.K.Gautam,” Performance evaluation of squaring operation by Vedic Mathematics,” IETE Journal of Research, pp.39-41, (2011).

[10].Shubham Mishra, Shailendra Kumar Dhakad,” A High Speed and device efficient FPGA based Squaring circuit”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol.02, No.11, (2013).

[11].Deepa A. and Marimuthu C N.,” VLSI Implementation of a Squaring Architecture Based On Yavadunam - An Algorithm of an Ancient Indian Vedic Mathematics”, paper submitted to a journal and not yet selected.

 DOI : 10.18231/2454-9150.2019.0113