ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

VEDIKALACH IMPLEMENTATION IN VLSI

Prof. R.U. Shekokar, Abhijeet Honwalkar, Akshay Patil, Swapnil Khalse

Department of E&TC, RMD Sinhgad School of Engineering, Warje Pune, India

ABSTRACT: A typical The need of high speed multiplier is increasing as the need of increasing high speed processors. A multiplier is one of the hardware blocks in most fast processing system. This is not only a high delay block but also a major source of power dissipation. A conventional processor requires substantially more hardware resources and more processing time in the multiplication operation, rather than addition and subtraction. In this project, we implement a high speed Multiplier using algorithm mentioned in Indian ancient Vedic mathematics which is utilized for all case of multiplication and to improve the performance of multiplier. In this method, as the number of multiplier bits increases it requires less number of calculations compare to other existing multiplication Algorithm. In this paper the Vedic multiplier is compared with the existing method such as booth and array multiplier. The Vedic multiplier is designed using Urdhva and Nikhilam sutras. This algorithm reduces the overall delay of the multiplier unit.

KEYWORDS: Vedic Multiplier, Image Processing, Digital Signal processing, Urdhava Tiryakbhyam sutra.

I. INTRODUCTION

With the up-to-the-minute encroachment of VLSI technology the insist for portable and embedded digital signal processing (DSP) systems has increased efficiently multipliers are key components of many high performance systems such are FIR filters, Microprocessors, Digital Signal Processors etc .In order to perform multiplications, a large number of adders or components are used. The ancient system of mathematics named as Vedic mathematics was rediscovered from the Vedas. In contrast to conventional mathematics, Vedic mathematics is simpler and easy to understand. In 1884. Swami Bharati Krishna Tirthaji Maharaj re-introduced the concept of ancient system of Vedic mathematics.The utterance 'Vedic' is consequential from the word ‘Veda’ which earnings the store-house of all knowledge. The Vedic mathematics includes sixteen-sutras or formulae and thirteen sub-sutras. The variety of applications of Vedic mathematics includes theory of numbers, compound multiplication, algebraic operation, calculus, squaring, cubing, cube root, simple quadratic, coordinate geometry and wonderful Vedic Numeric Code. So we can interpret as Vedic mathematics is a sphere of influence which presents various effective algorithms that can be applied in different twigs of engineering such as digital signal processing computing and especially VLSI Signal Processing applications. In the present world all the signal and data processing operations involves multiplication. Though speed is an important factor in the 3-dimensional VLSI problem and also a constraint in the multiplication operation, So increase in speed can be achieved by sinking the number of steps in the computation process. Hence the efficiency of the system can be evaluated by the help of Speed and area consumed by the components. of multiplier determines the efficiency of a system.

II. LITERATURE REVIEW

A. **Array Multiplier**: The delay associated with the array multiplier is the time taken by the signals to propagate through the gates that form the multiplication array. [3][4]

B. Booth Multiplier: Large booth arrays are required for high speed multiplication and exponential operations which in turn require large partial sum and partial carry registers.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

Multiplication of two n-bit operands using a radix-4 booth recording multiplier requires approximately n / (2m) clock cycles to generate the least significant half of the final product, where m is the number of Booth recorder adder stages. Thus, a large propagation delay is associated with this case.[4]

C. Vedic method: There are number of techniques that to perform binary multiplication. In general, the choice is based upon factors such as latency, throughput, area, and design complexity. More efficient parallel approach uses some sort of array or tree of full adders to sum partial products. Array multiplier, Booth Multiplier and Wallace Tree multipliers are some of the standard approaches to have hardware implementation of binary multiplier which are suitable for VLSI implementation at CMOS level.[4]

An Efficient multiplier should have following characteristics:-

Accuracy:-A good multiplier should give correct result.

Speed:-Multiplier should perform operation at high speed.

Area:-A multiplier should occupy less number of slices and LUTs.

Power:-Multiplier should consume less power.[3]

III. PROPOSED METHOD

A. Vedic Multiplier

The use of Vedic mathematics lies in the fact that it reduces the typical calculations in conventional mathematics to very simple one. This is so because the Vedic formulae are claimed to be based on the natural principles on which the human mind works.

Vedic Mathematics is a methodology of arithmetic rules that allow more efficient speed implementation. It also provides some effective algorithms which can be applied to various branches ofengineering such as computing.[2]

B. Urdhva Tiryakbhyam Sutra

Urdhva Tiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It means "Vertically and Crosswise" [15-16]. The digits on the two ends of the line are multiplied and the result is added with the previous carry. When there are more lines in one step, all the results are added to the previous carry. The least significant digit of the number thus obtained acts as one of the result digits and the rest act as the carry for the next step. Initially the carry is taken to be as zero. The line diagram for multiplication of two 4-bit numbers show in fig a.[1]

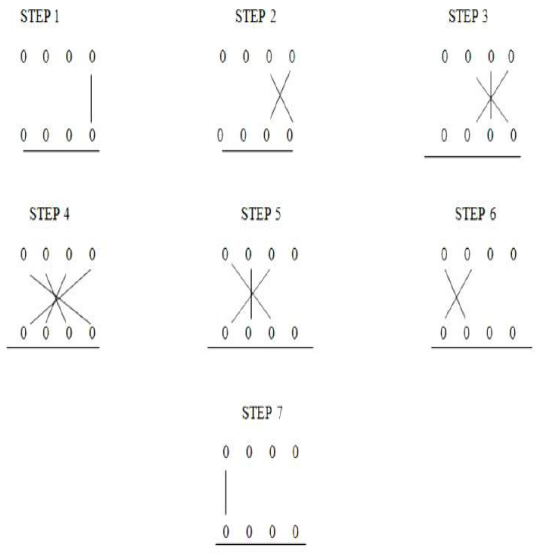

Fig 1: Urdhva Tiryakbhyam Sutra

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

C. 2X2 Architecture Vedic multiplier

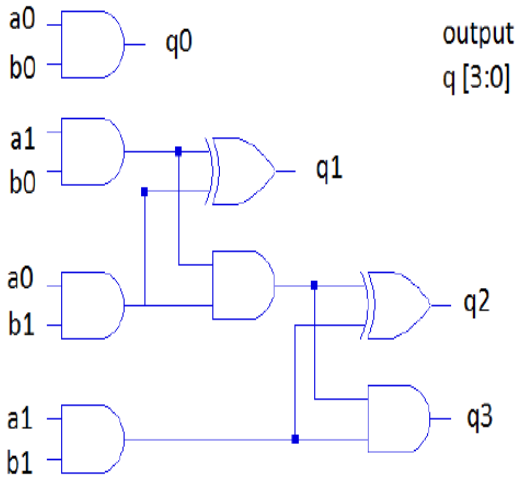

Fig 2: 2X2 Architecture Vedic multiplier

Converting the above fig. b to a hardware equivalent we have 3 and gate which will ac as 2 bit multiplier and two half adders to add the product to get the final product.here is the hardware details of the multiplier. Where "a" and "b" are two numbers to be multiplied and "q" is the product. With this design we are now ready to code this in verilog easily using and gates and HA (half adders). To make the design more modular we try to write code for HA first and then instantiate it to have the final product. This 2X2 Vedic multiplier module can be implemented using four input AND gates & two half-adders as specified in the block diagram shown in Fig. b Very precisely, the total delay is only 2-half adder delays after the generation of final bit products, which is very similar to Array multiplier.[2]

D. 4X4 Architecture Vedic multiplier

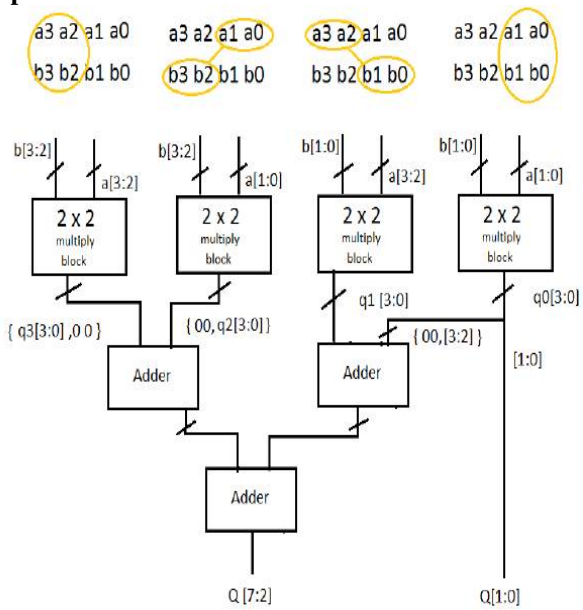

Fig 3: 4X4 Architecture Vedic multiplier

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

The word "Urdhva-Tiryakbhyam" resources vertical and crosswise multiplication.This multiplication formula is pertinent to all cases of algorithm for N bit numbers .Conventionally this sutra is used for the multiplication of two numbers in decimal number system. The architecture of 4x4 vedic multiplier show in fig. c The same concept can be applicable to binary number system which is being discussed in this paper. Advantage of using this type of multiplier is that as the number of bits increases, delay and area increases very slowly as compared to other multipliers.[2]

E. Simulation Results in DSCH3:

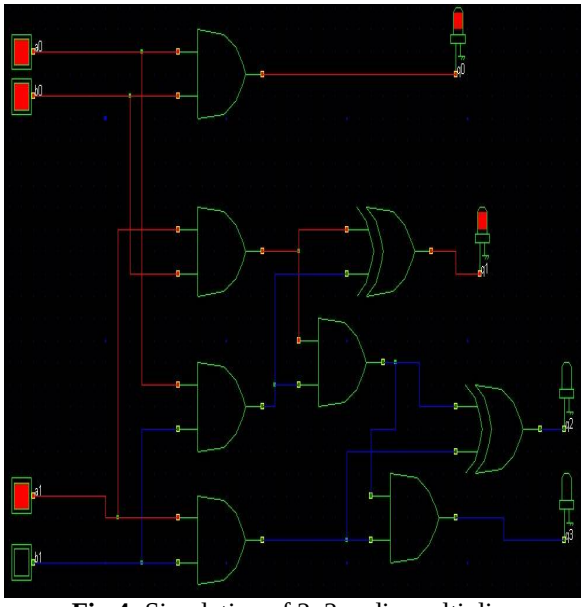

Fig 4: Simulation of 2x2 vedic multiplier

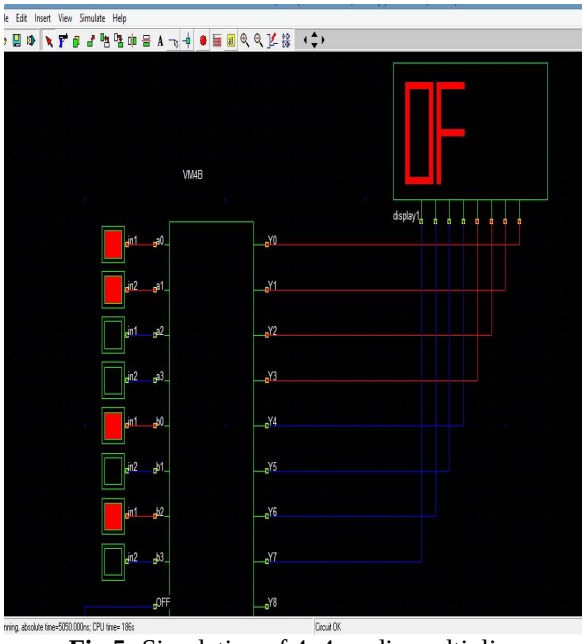

Fig 5: Simulation of 4x4 vedic multiplier

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

F. Design of 4x4 Vedic Multiplier In Xilinx ISE 14.4

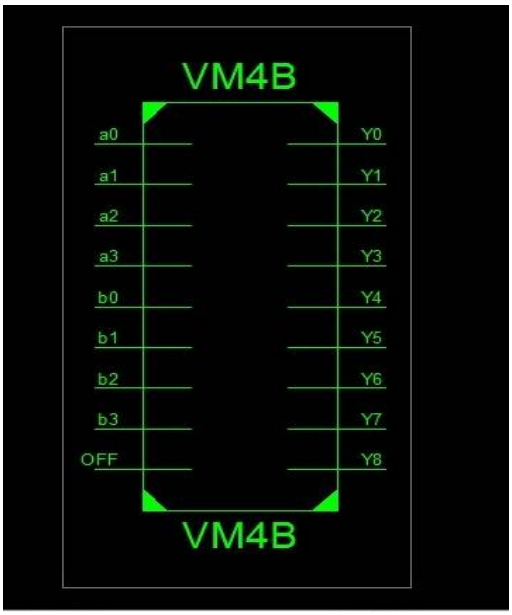

Fig 6: Rtl Schematic

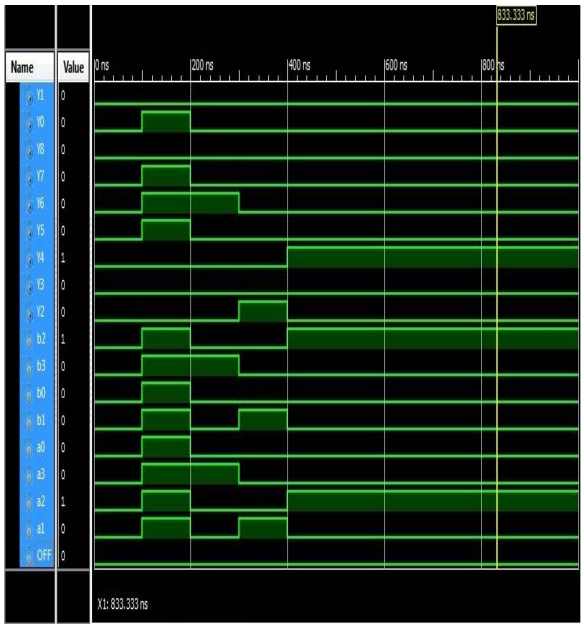

Fig 7: Behavioral Simulation

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

III. RESULTS & CONCLUSION

Viewing below a Synthesis Report. The synthesis report allows us to view the results of the net list-generation synthesis process. In this report, summary of 4x4 vedic multiplier synthesis options, and a summary and analysis of the net list generation are shown .default path analysis and timing constraints are shown in in this synthesis report.

```
Timing constraint: Default path analysis
  Total number of paths / destination ports: 387 / 9
-------------------------------------------------------------------------------
Delay:               15.131ns (Levels of Logic = 10)
  Source:            b0 (PAD)
  Destination:       Y7 (PAD)

  Data Path: b0 to Y7
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             6   1.218   0.844  b0_IBUF (b0_IBUF)
     LUT2:I0->O            2   0.704   0.451  w261 (w26)
     LUT4:I3->O            2   0.704   0.451  w761 (w76)
     LUT4:I3->O            2   0.704   0.451  w721 (w72)
     LUT4:I3->O            2   0.704   0.622  Mxor_w6_Result1 (w6)
     LUT3:I0->O            2   0.704   0.622  w571 (w57)
     LUT3:I0->O            2   0.704   0.526  w441 (w44)
     LUT3:I1->O            2   0.704   0.622  w471 (w47)
     LUT4:I0->O            1   0.704   0.420  Mxor_Y7_Result1 (Y7_OBUF)
     OBUF:I->O                 3.272          Y7_OBUF (Y7)
    ----------------------------------------
    Total                     15.131ns (10.122ns logic, 5.009ns route)
                                       (66.9% logic, 33.1% route)

=========================================================================


Total REAL time to Xst completion: 6.00 secs
Total CPU time to Xst completion: 6.02 secs
```

Fig 8: Synthesis report

IV. CONCLUSION

Vedic It can be concluded that Booth Multiplier is superior in all respect like speed, delay, area, complexity, power consumption. However Array Multiplier requires more power consumption and gives optimum number of components required, but delay for this multiplier is larger than Wallace Tree Multiplier. Hence for low power requirement and for less delay requirement Booth's multiplier is suggested. Ancient Indian Vedic Mathematics gives efficient algorithms or formulae for multiplication which increase the speed of devices. Urdhva Tiryakbhyam, is general mathematical formula and equally applicable to all cases of multiplication. Also, the architecture based on this sutra is seen to be similar to the popular array multiplier where an array of adders is required to arrive at the final product. Due to its structure, it suffers from a high carry propagation delay in case of multiplication of large number. This problem can solved by Nikhilam Sutra which reduces the multiplication of two large numbers to the multiplication of two small numbers. The power of Vedic Mathematics can be explored to implement high performance multiplier in VLSI applications. Nikhilam Sutra in Vedic Mathematics is less complex than Urdhva Tiryakbhyam which can be tested with its implementation with different logics in VLSI.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

V. FUTURE SCOPE

In this way we can conclude that Vedic multiplier is better than conventional multiplier but after including new encoding algorithm new design is going to be faster. It reduces the complexity associated with Vedic multiplier. Due to reduction in hardware it is going to be less costly than previous structures. Propagation delay in case of multiplier is high but here it is going to be less. We can simply apply this multiplier in various applications such as in implementation of RSA, FFT and in all Digital Signal Processing algorithms. The future work includes is the fixed-width model of this multiplier for low hardware complexity & more reduction in cost and size.

REFERENCES

[1] Swami Bharati Krishna Tirthaji Maharaja, "Vedic Mathematics", Motilal Banarsidas Publishers, 1965.
[2] http://verilog-code.blogspot.in/2014/01/design-and-implementation-of-16-bit.html.
[3] Sumit R. Vaidya Department of Electronic and Telecommunication Engineering OM College of Engineering Wardha, Maharashtra, India. vaidyarsumit@gmail.com
[4] "A Novel Parallel Multiply and Accumulate (V-MAC)Architecture Based On Ancient Indian Vedic Mathematics" Himanshu Thapliyal and Hamid RArbania.
[5] ASIC Implementation of 4 Bit Multipliers" Pravinkumar Parate ,IEEE Computer society. ICETET,2008.25.
[6] Moises E. Robinson and Ear Swartzlander, Jr."A Reduction Scheme to Optimize the Wallace Multiplier" Department of Electrical and Computer Engineering, University of Texas at Austin, USA. Tam Anh Chu, "Booth Multiplier with Low Power High Performance Input Circuitary", US Patent, 6.393.454 B1,May 21, 2002.
[7] Morris Mano, "Computer System Architecture", PP. 346-347, 3rd edition, PHI. 1993.
[8] ASIC Implementation of 4 Bit Multipliers" Pravinkumar Parate ,IEEE Computer society. ICETET,2008.25.