ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Low Power, Time Efficient & Less Complexity Vedic Square & Cube Architectures for 16 Bit Implementation

Kallakola Mounika
M.Tech, VLSI,
Dept of ECE,
Kshatriya College of Engineering.

Kanoor Devendher
M.Tech, VLSI,
Dept of ECE,
Kshatriya College of Engineering.

ABSTRACT:

The main objective of this project is to design a Low power, Time efficient and less area square and cube architectures using Vedic sutra. Square and cube architectures can be used as element in such applications like DSP, DIP, and communications etc. These are very efficient because the operation not as with the normal, and also the have the pipelined execution feature. Often times square and cube are the most time-consuming operations in many of digital signal processing applications and computation can be reduced using modified booth(Radix8) and the overall processor performance can be improved for many applications. We proposed a novel approach of designing the square and cube architectures by using the one of the fast and less power method called Anurupyena sutra of Vedic mathematics.

In proposed design Vedic multiplication method Urdhva Tiryakbhyam Sutra is chosen for the sub module design. There by decreases the critical path delay compare to that of any other alternative method. Squaring utilized Duplex property properties of Urdhva Tiryakbhyam and cubing used Anurupyena sutra. The proposed system aimed at performing both these square and cube architectures using a single algorithm which is time efficient and has simple architecture based on Vedic mathematics. The modified system enabled performing 8-bit multiplication and addition besides 16-bit squaring and cubing. Thus the thesis comprises of a single architecture that can perform addition, multiplication and also special case of multiplication that is square and cube of 16-bit numbers.

1. INTRODUCTION:

Vedic Mathematics hails from the ancient Indian scriptures called “Vedas” or the source of knowledge. The “Vedic Mathematics” is called so because of its origin from Vedas. To be more specific, it has originated from “Atharva Vedas” the fourth Veda. This system of computation covers all forms of mathematics, be it geometry, trigonometry or algebra. The striking feature of Vedic Mathematics is the coherence in its algorithms which are designed the way our mind naturally works. This makes it the easiest and fastest way to perform any mathematical calculation mentally. Vedic Mathematics is believed to be created around 1500 BC and was rediscovered between 1911to 1918 by Sri Bharti Krishna Tirthaji (1884-1960) who was a Sanskrit scholar, mathematician and a philosopher. He organized and classified the whole of Vedic Mathematics into16 formulae or also called as sutras. These formulae form the backbone of Vedic mathematics which are listed as

1. Ekadhikena Purvena
2. Nikhilamnavatascaramam Dasatah
3. Urdhva - tiryagbhyam
4. ParavartyaYojayet
5. SunyamSamyaSamuccaye
6. Anurupye - Sunyamanyat
7. Sankalana - Vyavakalanabhyam
8. Puranapuranabhyam
9. Calana - Kalanabhyam
10. EkanyunenaPurvena
11. Anurupyena
12. Adyamadyenantya - mantyena
13. YavadunamTavadunikrtyaVargancaYojayet
14. AntyayorDasakepi
15. Antyayoreva
16. GunitaSamuccayah.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Great amount of researchhas been done all these years to implement algorithms ofVedic mathematics on digital processors. It has beenobserved that due to coherence and symmetry in thesealgorithms it can have a regular silicon layout and consume less area along with lower power consumption.

II. RELATED WORKS

Up till now the square and cube systems are designed using normal methods composes of multiplier. Normally the multiplier is more power consumed digital design as this internally consists the partial product generation and multi operand addition and final addition that causes the increases the area and power in advancement of present VLSI features we cannot accept this type of designing. Even in for the design of performance increased multiplier the present researches are moving to Vedic mathematic approaches. Vedic mathematic is one of the promising alternatives for the present ALU requirements. The idea for designing the multiplier and adder unit was adopted from ancient Indian mathematics "Vedas".

Based on those formulae, the partial products and sums are generated in single step which reduces the carry propagation from LSB to MSB. The implementation of the Vedic mathematics and their application to the complex multiplier ensured substantial reduction of propagation delay in comparison with Distributed Array (DA) based architecture and parallel adder based implementation which are most commonly used architectures. It was shown that there is an appreciable saving in the processing time of the Vedic multiplier as when compared to that of a conventional multiplier.

III. VEDIC MATHEMATICS SQUARE ARCHITECTURE:

The Dwandwa Yoga or 'duplex combination' can be used for general purpose squaring. The square of a number can be calculated by using the _duplex,, D property of dwandwa yoga. According to duplex property, for an even number of elements the result is taken as twice the product of the outermost pair and then twice the product of the next outermost pair and so on till no pairs are left. For an odd number of elements, there is one bit left itself in the middle and this enters as its square along with the product elements. It can be explained in the following example.

D(a) = a2
D(ab) = 2ab
D(abc) = 2ac + b2
D(abcd) = 2ad + 2bc
D(abcde) = 2ae + 2bd + c2
D(abcdef) = 2af + 2be + 2cd and so on....

As we can see above, D of any number is the sum of square of the middle number and two times the product of the other pairs. Square of a number is given by

(ab)2= D(a) | D(ab) | D(b)
(abc)2= D(a) | D(ab) | D(abc)|D(bc) | D(c)
(abcd)2= D(a)|D(ab) |D(abc)|D(abcd) | D(bcd) | D(bc) | D (c)

Example:

(45)^2=
D(5) = 5^2 = 25 = 5 = A
D(45) = 2*4*5 = 40 = 40+2 = 42 = 2= B
D(4) = 4^2= 16 = 16+4 = 20 = C

Now the required result after squaring 45 is CBA = 2025. Thus for a single bit number, the D is square of the number itself. For a 2 bit number, it is twice their product. For a 3 bit number, it is the sum of twice the product of the outermost pair and square of the middle number. For a 4 bit number, it is the sum of twice the product of the outermost pair and twice the product of the innermost pair. In our proposed architecture we use the advance multioper and addition method along with parallel processing of the internal method. The Vedic square has all the advantages of the Vedicmultiplier.lot more it is degree faster and smaller footprint compared to that of the array, Booth. Rather than adding the multiplier result twice and then adding to the Concatenate MSP square lower half and LSP square upper half, We add them together thatconsumes the

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

less area and increases speed of operation.In the proposed algorithm the square of a binary number can be calculated based on the duplex property of dwandwa yoga logic as shown in Fig.1

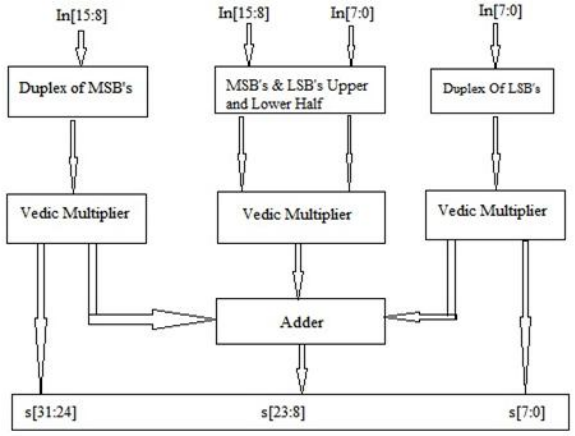

Fig1: Square Architecture

Cube Architecture:

The cube architecture can be designed by using the Vedic sutra called Anurupyena Sutra even though there are different sutras for the cube calculations but this Anurupyena Sutra is ideal choice. In this method we use the hierarchy design for the sub modules design. For multiplication we use the Urdhva tiryakbyham sutra. The steps that are involved in Anurupyena Sutra are described below.

Step 1: calculate the cubes of LSP and MSP
Step 2: calculate the squares of LSP and MSP
Step 3: multiply the squared LSP result with MSP and squared MSP result with LSP (Intermediate results)
Step 4: multiply the intermediate results with '3'(0011)
Step 5: concatenate the all results

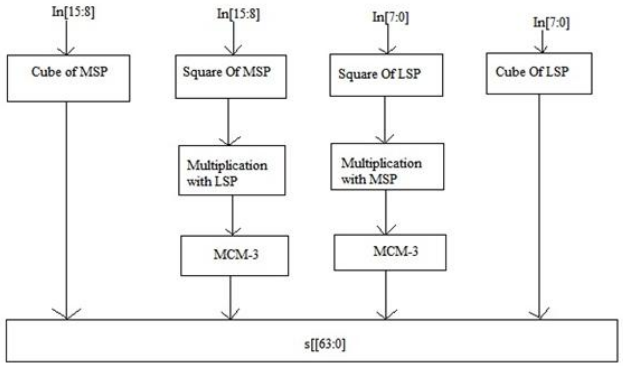

Fig 2: Cube Architecture

For multiple constant multiplication we add the input (X) and left shifted input (2X) to produce the 3Xoutput.There by unnecessary multiplier can be avoided. Proposed MCM-3 is shown in fig.2

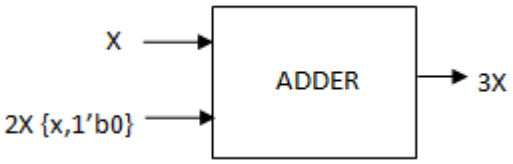

Fig 3:MCM-3 Module

Example:

(45)^3=
MSP=4,LSP=5
Step 1: Cube Of MSP & LSP
(4)^3=64
(5)^3=125
Step 2: Squares Of MSP & LSP
(4)^2=16
(5)^2=25
Step 3: (MSP^2 * LSP),(LSP^2*MSP)
(4^2*5)=80
(5^2*4)=100
Step 4: Multiplication Of Intermediate Results with 3
3*80=240
3*100=300
Step 5: Concatenation the results
64|240|300|125=91125

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Note the carryovers carefully in the example above. From the right most columns, 12 are carried over to the left. This gets added to 300, giving 312, of which 2 remains and 31 is carried over further to the left. This gets added to 240, giving 271. 1 remains and 27 are carried over to the left-most column, further adding with 64 and giving the final sum of 91. Each column should consist of one digit, with all the excess digits carried over to the left until the left-most column, which obviously will not have any carryover out of it.

IV. 4 BIT VEDIC MULTIPLIER

Multiplication methods are extensively discussed in Vedic mathematics. Various tricks and short cuts are suggested by VM to optimize the process. Following Fig.2 shows the architecture of 2*2 Vedic multiplier.

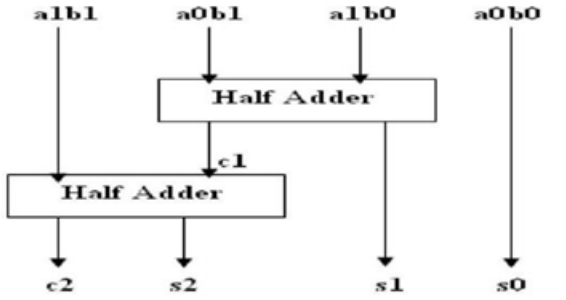

Fig 4. The architecture of 2*2 vedic multiplier.

The 4x4 bit Vedic multiplier module is implemented using four 2x2 bit Vedic multiplier modules as discussed. Let"sanalyze 4x4 multiplications, say A= A3 A2 A1 A0 and B= B3 B2 B1 B0. The output line for the multiplication result is – S7 S6 S5 S4 S3 S2 S1 S0 .Let"s divide A and B into two parts, say A3A2 & A1 A0 for A and B3 B2 & B1B0 for B. Using the fundamental of Vedic multiplication, taking two bit at a time and using 2 bit multiplier block, we can have the following structure for multiplication as shown in below fig.3.

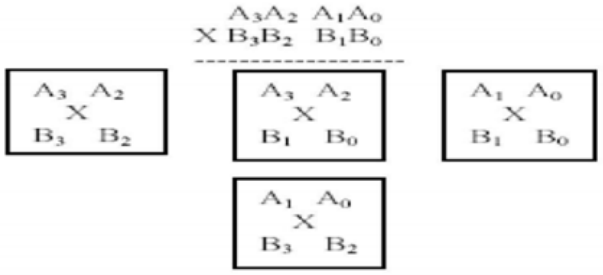

Fig 4. Structure for multiplication.

SIMULATION RESULTS :

In our work, 16-bit squaring and cube architectures aredesigned in Verilog HDL. Simulation and Synthesis are done using Xilinx - Project Navigatorand XilinxISE simulator.

The power consumption results of the proposed system are 0.203W for cube architecture and 0.065W for square architecture. Also the time required for cube and square architectures 35ns and 3.2ns.Also the hardware required for the proposed system used up to 25-40%.

V.CONCLUSION:

Digital signal processing systems are the systems mainly utilizing square and cube operations. It is the bottleneck of many of the DSP operations, hence by increasing the speed of square and cube operations we can improve the overall performance. Vedic square and cube multiplier can be designed with the advanced design methods that makes the efficient design and reduces the operation time. The extra advanced logic MCM based design can be implemented. Rather than using the normal multiplier MCMuses very less area.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The 8-bit, 16-bit architectures are designed andcomparison evaluation is carried out in this paper.

Category	Square (8 Bit)	Cube (8 Bit)	Square (16 Bit)	Cube (16 Bit)	Conventional Methods
Speed	5.7ns	41ns	3.2ns	35ns	50ns
Power	0.8W	1.0 W	0.065W	0.065W	1-1.5W
Complexity	40-50%	40-50%	25-40%	25-40%	75%

VI.REFERENCES:

[1]Vaijyanathkunchigi, Linganagouda Kulkarni and Subhash Kulkarni "Low Power Square and Cube Architectures Using Vedic Sutras" 2014 Fifth International Conference on Signals and Image Processing.

[2]Swami Bharati Krisna Tirtha, "Vedic Mathematics," Motilal Banarsidass Publishers, Delhi, 1965.

[3]Purushottam D. Chidgupkar and Mangesh T. Karad, ―T Implementation of Vedic Algorithms in Digital Signal Processing‖, Global J. of Engng. Educ., Vol.8, No.2 © 2004 UICEE Published in Australia.

[4]Himanshu Thapliyal and Hamid R. Arabnia, ―A Time-Area- Power Efficient Multiplier and Square Architecture Based On Ancient Indian Vedic Mathematics‖, Department of Computer Science, The University of Georgia, 415 Graduate Studies Research Center Athens, Georgia 30602-7404, U.S.A.

[5]E. Abu-Shama, M. B. Maaz, M. A. Bayoumi, ―A Fast and Low Power Multiplier Architecture‖, The Center for Advanced Computer Studies, The University of South western Louisiana Lafayette, LA 70504.

[6]Harpreet Singh Dhillon and Abhijit Mitra, ―A Reduced- Bit Multiplication Algorithm for Digital Arithmetics‖, International Journal of Computational and Mathematical Sciences 2;2 www.waset.org Spring 2008.