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

An FPGA Implementation of Low Power Square and Cube Architectures using Nikhilam Sutra

Medimi Rani
MTech, VLSI and Embedded Systems,
Department of ECE,
Vignan's Institute of Engineering for Women
Visakhapatnam, AP.

SD.Nageena Parveen
Assistant Professor,
Department of ECE,
Vignan's Institute of Engineering for Women
Visakhapatnam, AP.

Abstract:

In this paper an FPGA implementation of low power square and cube architectures using Nikhilam sutra have been proposed. Multipliers are extensively used in FIR filters, Microprocessors, DSP and communication applications. For higher order multiplications, a huge number of adders or compressors are to be used to perform the partial product addition. The need of low power and high speed Multiplier is increasing as the need of high speed processors are increasing. The multipliers used in Square and cube architecture have to be more efficient in area and also in power. In this paper a multiplier is implemented based on Nikhilam sutra which gives an efficient results when compared to the existing system. Comparison is made between conventional and Vedic method implementations of square and cube architecture. Implementation results show a significant improvement in terms of area and power. Proposed square and cube architectures can be used for area efficient and low power applications. Synthesis is done on Xilinx FPGA Device using, Xilinx Family: Spartan 3E, Speed Grade: - 4.

Keywords- Vedic mathematics; Nikhilam Sutra; Anurupyena; UrdhvaTiryakbhyam; Square; Cube; low power; high speed.

I. INTRODUCTION

As the scale of consolidation keeps growing, more and more sophisticated signal processing systems are being implemented on a VLSI chip." These signal processing applications not only demand great computation capacity but also eat up considerable amount of energy. "While performance and area remain to be two major design goals, power utilization has become a critical concern in today's system design." The need of low power VLSI systems occurs from two main forces, power consumption has become a critical concern in today's system design. The need of low power VLSI systems arises from two main forces. First, with the steady growth of operating frequency and processing capacity per chip, large current has to be delivered and the heat due to large power consumption must be removed by proper cooling techniques. Second, battery life in portable electronic devices is limited. Low power design directly leads to prolonged operation time in these portable devices. Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore, low power multiplier design has been an important part in low power VLSI system design. There has been extensive work on low power multipliers at technology, physical, circuit and logic levels. These low-level techniques are not unique to multiplier modules and they are generally applicable to other types of modules. The characteristics of arithmetic computation in multipliers are not considered well. Digital multipliers are the core components of all the digital signal processors (DSPs) and the speed of the DSP is largely determined by the speed of its multipliers. Two most common multiplication algorithms followed in the digital hardware are array multiplication algorithm and Booth multiplication algorithm. The computation time taken by the array multiplier is comparatively less because the partial products are calculated independently in parallel. The delay associated with the array multiplier is the time taken by the signals to propagate through the gates that form the multiplication array.

Square and cube are frequently performed functions in most of the DSP systems. Square and cube are special cases of multiplication. Square and cube architectures forms the heart of the different DSP operations like Image Compression, Decoding, Demodulation,

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

Adaptive Filtering, Least Mean Squaring etc., and also have numerous applications as mentioned in [1] such as cryptography, computation of Euclidean distance among pixels for a graphics processor or in rectangular to polar conversions in several signal processing circuits where full precision results are not required. Traditionally, square and cube were performed using multiplier itself.

In this paper algorithms and architectures used to design square and cube of a binary number is explored and to create a circuit using the Vedic Sutras. Often times, square and cube are the most time-consuming operations in many of digital signal processing applications and computation can be reduced using the Vedic sutras and the overall processor performance can be improved for many applications [8]. Therefore, the goal is to create a square and cube architectures that is comparable in speed, power and area than a design using an standard multiplier. The motivation behind this work is to explore the design and implementation of Square and Cube architectures for low power.

This paper is organized as follows. Section II gives the Description and analysis of existing architecture. Section III briefs about proposed architecture section IV details about sub modules in proposed architecture section V applications section VI Results and VII about conclusion.

II. DESCRIPTION AND ANALYSIS

VEDIC mathematics is the ancient Indian system of mathematics which mainly deals with Vedic mathematical formulae and their application to various branches of mathematics. The word „Vedic" is derived from the word „Veda" which means the store-house of all knowledge. Vedic mathematics was reconstructed from the ancient Indian scriptures (Vedas) by Sri Bharati Krishna Tirtha (1884-1960) after his eight years of research on Vedas. According to his research, Vedic mathematics is mainly based on sixteen principles or word-formulae which are termed as Sutras. The beauty of Vedic mathematics lies in the fact that it reduces otherwise cumbersome looking calculations in conventional mathematics to a 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. This is a very interesting field and presents some effective algorithms which can be applied to various branches of engineering such as computing and digital signal processing. This paper discusses a possible application of Vedic mathematics to digital signal processing in the light of application of Vedic multiplication algorithm to digital multipliers. Digital multipliers are indispensable in the hardware implementation of many important functions such as fast Fourier transforms (FFTs) and multiply accumulate (MAC). This has made them the core components of all the digital signal processors (DSPs). Two most common multiplication algorithms followed in the math coprocessor are array multiplication algorithm and booth multiplication algorithm. The array multipliers take less computation time because the partial products are calculated independently in parallel. The delay associated with the array multiplier is the time taken by the signals to propagate through the gates that form the multiplication array. This paper presents a simple digital multiplier architecture based on the ancient Vedic mathematics Sutra (formula) called Urdhva Tiryakbhyam (Vertically and Cross wise) Sutra which was traditionally used for decimal system in ancient India, this Sutra is shown to be a much more efficient multiplication algorithm as compared to the conventional counterparts.

Another paper has also shown the effectiveness of this sutra to reduce the $N \times N$ multiplier structure into efficient 4×4 multiplier structures. However, they have mentioned that this 4×4 multiplier section can be implemented using any efficient multiplication algorithm. We apply this Sutra to binary systems to make it useful in such cases. In particular, we develop an efficient 4×4 digital multiplier that calculates the partial products in parallel and hence the computation time involved is less."Urdhva Tiryakbhyam" is a Sanskrit word means vertically and cross wise formula is used for smaller number multiplication. "Nikhilam Navatascaramam Dashatah" also a Sanskrit term indicating "all from 9 and last from 10", formula is used for large number multiplication. The architecture of the designed Vedic multiplier comes out to be very similar to that of the popular array multiplier and hence it should be noted that Vedic mathematics provides much simpler derivation of array multiplier than the conventional mathematics.

Urdhva Tiryakbhyam Sutra:

This is the general formula which is applicable to all cases of multiplication. Urdhva Triyagbhyam means

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

"Vertically and Crosswise", which is the method of multiplication followed.
In this paper, we studied comparative of different multipliers is done for low power requirement and high speed. The paper gives information of "UrdhvaTiryakbhyam" algorithm of Ancient Indian Vedic Mathematics which is utilized for multiplication of two or more operands for

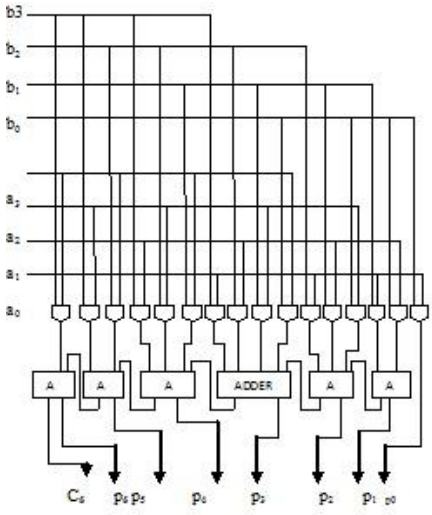

Figure 1: Hardware architecture of the Urdhva tiryakbhyam multiplier improving the speed, area parameters of multipliers. Vedic Mathematics has been suggests one more formula for multiplication of large number i.e. "Nikhilam Sutra" which can also increase the speed of multiplier by reducing the number of iterations

Square algorithm

In order to calculate the square of a number, we have utilized "Duplex" D property of Urdhva Triyakbhyam. In the Duplex, we take twice the product of the outermost pair and then add twice the product of the next outermost pair and so on till no pairs are left. When there are odd numbers of bits in the original sequence, there is one bit left by itself in the middle and this enters as its square.

Thus for 987654321[3],

D = 2 × (9 × 1) + 2 × (8 × 2) + 2 × (7 × 3) + 2 × (6 × 4) + 5 × 5 = 165.

Further, the Duplex can be explained as follows:
For a 1 bit number D is its square.
For a 2 bit number D is twice their product
For a 3 bit number D is twice the product of the outer pair + square of the middle bit.
For a 4 bit number D is twice the product of the outer pair + twice the product of the inner pair.

The algorithm is explained for 4 x 4 bit number. The Vedic square has all the advantages as it is quite faster and smaller than the array, Booth and Vedic multiplier.

CP = Cross Product (Vertically and Crosswise)

$X_3\ X_2\ X_1\ X_0$	Multiplicand
$X_3\ X_2\ X_1\ X_0$	Multiplier
H G F E D C B A	
$P_7\ P_6\ P_5\ P_4\ P_3\ P_2\ P_1\ P_0$	Product

Algorithm for 4 × 4 bit Square Using Urdhva Tiryakbhyam D - Duplex

PARALLEL COMPUTATION :

The parallel computation of bits in urdhva triya kbhyam sutra has shown below

1. D = X 0 × X0 = A
2. D = 2 × X1 × X0 = B
3. D = 2 × X2 × X0 + X1 × X1 = C
4. D = 2 × X3 × X0 + 2 × X2 × X1 = D
5. D = 2 × X3 × X1 + X 2 × X2 = E
6. D = 2 × X3 × X2 = F
7. D = X3 × X3 = G

The diagrammatical representation of square architecture using urdhva tiryakbhyam sutra is shown below

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

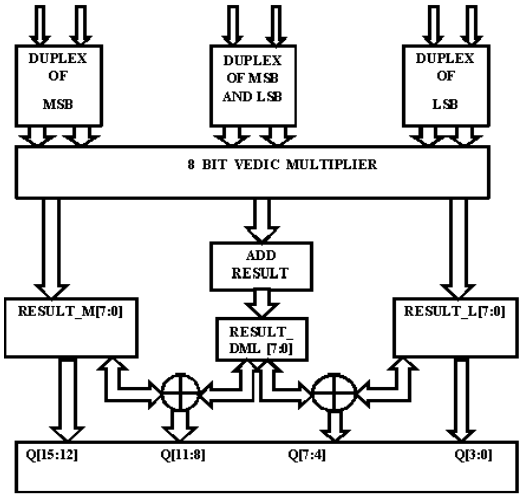

Figure 5: Existing Square architecture

This is the square architecture based upon the dwadwa yoga and the duplex properties of urdhva tiryakbhyam sutra in this we are using the 8 bit Vedic multiplier of urdhva tiryakbhyam sutra the duplex property is nothing but taking twice the input value if we have given 8 bit input than it will consider [3:0] as LSB and [4:7] as MSB the duplex of this MSB and LSB bits are considered. first the duplex of LSB is taken and then duplex of MSB and LSB is taken and finally duplex of MSB has been taken in this square architecture and these are given as input to the multiplier the multiplied bit of MSB and LSB has been added twice .now the 0:3 of multiplied bit is considered directly to result and 4:7 is added to multiplied bit of MSB and LSB 0:3 his process continuous till the final set of MSB bit

Cube algorithm

The cube of the given input value is based on the Anurupye Sutra of Vedic Mathematics anurupye sutra literally means If one is in ratio, the other is zero which states If you start with the cube of the first digit and take the next three numbers (in the top row) in a Geometrical Proportion (in the ratio of the original digits themselves) then you will find that the 4th figure (on the right end) is just the cube of the second digit[3].

If a and b are two digits then according to Anurupye Sutra, This sutra has been utilized in this work to find the cube of a number. The number given which is given as input to find the cube of it consisting of

$$\begin{array}{r} a^3 \quad a^2b \quad ab^2 \quad b^3 \\ 2a^2b \quad 2ab^2 \\ \hline a^3 + 3a^2b + 3ab^2 + b^3 = (a+b)^3 \end{array}$$

Multiplication in cube algorithm

N bits is divided in two partitions of N/2 bits, say a and b, and then the Anurupye Sutra is applied to find the cube of the given input number. In the above algebraic explanation of the Anurupye Sutra, we have seen that a3 and b3 are to be calculated in the final computation of (a+b)3.

The intermediate a3 and b3 can be calculated by recursively applying Anurupye sutra.

$$\begin{array}{r} (15)^3 = 1 \quad 5 \quad 25 \quad 125 \\ 10 \quad 50 \\ \hline 3375 \end{array}$$

Example for multiplication in cube algorithm

The diagrammatical representation of cube architecture using urdhva tiryakbhyam sutra is shown below

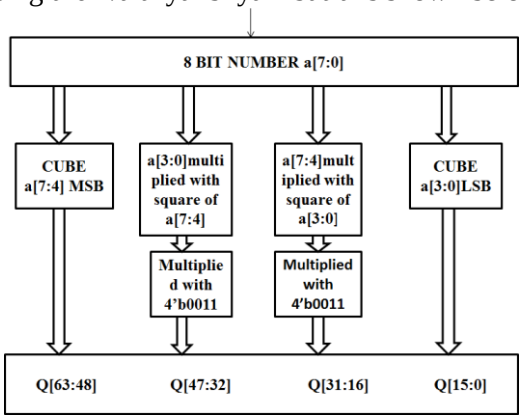

Figure 6: Existing Cube architecture

This is the cube algorithm for urdhva tiryakbhyam sutra we can say that this is the simple diagrametical

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

representation of algebraic expression [a+b] 3 here we are taking the MSB bit and LSB bit of input MSB is considered as (a) and LSB is considered as (b) in this way we are finding out the cube of given input value.

III NIKHILAM SUTRA MULTIPLIER ARCHIT ECTURE DESIGN

Assume that the multiplier is X and multiplicand is Y. Though the designation of the numbers is different but the architecture implemented is same to some extent for evaluating both the numbers.

The mathematical expression of modified nikhilam sutra is given below.

P=X*Y=(2^k2)*(X+Z2*2^(k1-2))+Z1*Z2. (1)

Where k1, k2 = the maximum power index of input numbers X and Y respectively.

Z1 and Z2 = the residues in the numbers X and Y respectively. The hardware construction of the above expression is partitioned into three blocks.

i. Base Selection Module

ii. Power index Determinant Module

iii. Multiplier.

The base selection module (BSM) is used to select the maximum base with respect to the input numbers. The second sub-module power index determinant (PID) is used to extract the power index of k1 and k2. The multiplier comprises of base selection module (BSM), power index determinant (PID), subtractor, barrel shifter, adder/subtractor as sub-modules in the architecture.

A. Base selection module.

The base selection module of this multiplier has power index determinant (PID) as the sub-module along with barrel shifter, adder, average determinant, comparator and multiplexer.

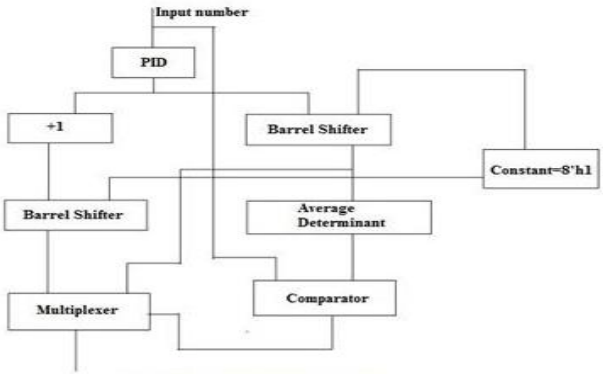

Figure 2: Base Selection Module, BSM

Operation:

An input 8-bit number is fed to power index determinant (PID) to interpret maximum power of number which is fed to barrel shifter and adder. The output of the barrel shifter is „n" number of shifts with respect to the adder output and the input based to the shifter. Now, the outputs of the barrel shifter are given to the multiplexer with comparator input as a selection line. The outputs of the average determinant and the barrel shifter are fed to the comparator. The required base is obtained in accordance with the multiplexer inputs and its corresponding selection line.

B. Power index determinant.

The input number is fed to the shifter which will shift the input bits by one clock cycle. The shifter pin is assigned to shifter to check whether the number is to be shifted or not. In this power index determinant (PID) the sequential searching has been employed to search for first „1" in the input number starting from MSB. If the search bit is 0 then the counter value will decrement up to the detection of input search bit is 1. Now the output of the decrementer is the required power index of the input number.

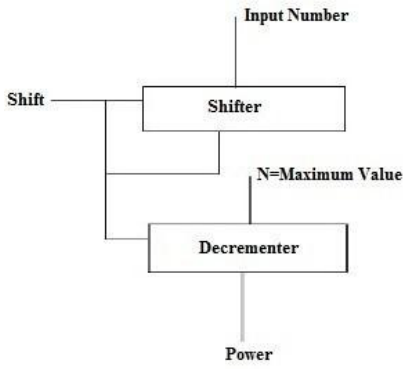

Figure 3: Power index determinant

IV SQUARE AND CUBE ARCHITECTURE

The base selection module and the power index determinant form integral part of multiplier architecture. The architecture computes the mathematical expression in equation1.Barrel shifter used in this architecture.

The two input numbers are fed to the base selection module from which the base is obtained. The outputs of base selection module (BSM) and the input numbers X and Y are fed to the subtractors. The subtractor blocks are required to extract the residual parts z1 and

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

z2. The inputs to the power index determinant are from base selection module of respective input numbers.

The sub-section of power index determinant (PID) is used to extract the power of the base and followed by subtractor to calculate the value. The outputs of subtractor are fed to the multiplier that feeds the input to the second adder or subtractor. Likewise the outputs of power index determinant are fed to the third subtractor that feeds the input to the barrel shifter. The input number X and the output of barrel shifter are rendered to first adder/subtractor and the output of it is applied to the second barrel shifter which will provide the intermediate value. The last sub-section used in this multiplier architecture is the second adder/subtractor which will provide the required result.

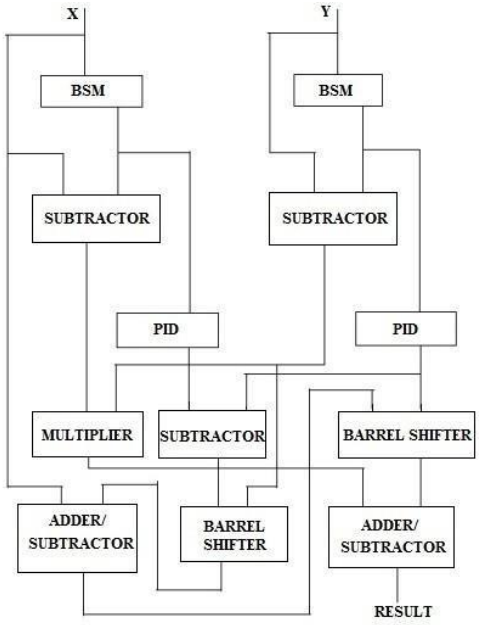

Figure 4: Nikhilam sutra Multiplier architecture

V. APPLICATIONS

1. The speed of multiplication operation is of great importance in DSP. Digital Signal processing is a technology that is present in almost every engineering discipline. It is also the fastest growing technology of the century and hence it posses tremendous challenges to the engineering community. Faster addition and multiplication are of extreme importance in DSP for Convolution, DFT and Digital filters .The core computing process is always a multiplication routine and hence DSP engineers are constantly looking for new algorithms and hardware to implement them. The methods in Vedic Multipliers are complementary directly and easy.
2. Low power VLSI system design.
3. Frequency domain filtering (FIR and IIR), frequency-time transformations (FFT), Correlation, Digital Image processing.
4. Because of high speed of Vedic multiplication ALU utilizes this algorithm to give reliable output.

VI. RESULTS

The following figures show the simulation and also synthesis results of proposed Square and Cube Architecture.

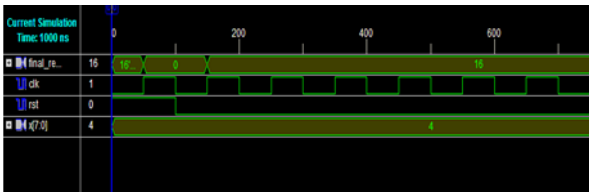

Figure 7: Simulation results of proposed Square Architecture.

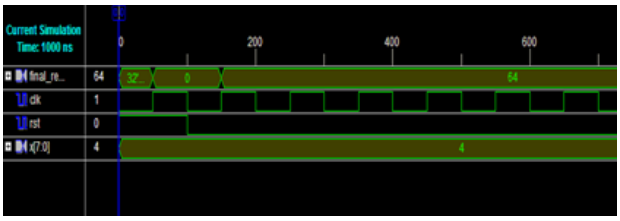

Figure 8: Simulation Results of proposed Cube Architecture.

SQUARE_URDHVATIRYAKBHYAM Partition Summary			
No partition information was found.			

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of 4 input LUTs	188	4,896	3%
Logic Distribution			
Number of occupied Slices	99	2,448	4%
Number of Slices containing only related logic	99	99	100%
Number of Slices containing unrelated logic	0	99	0%
Total Number of 4 input LUTs	188	4,896	3%
Number of bonded IOBs	26	108	24%
IOB Flip Flops	15		
Number of GCLKs	1	24	4%
Total equivalent gate count for design	1,266		
Additional JTAG gate count for IOBs	1,248		

Figure 9: Area of Square UrdhvaTiryakbhyam obtained from synthesis results

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

SQUARE_NIKHILAMSUTRA Partition Summary

No partition information was found.

Device Utilization Summary

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	115	4,896	2%
Logic Distribution			
Number of occupied Slices	60	2,448	2%
Number of Slices containing only related logic	60	60	100%
Number of Slices containing unrelated logic	0	60	0%
Total Number of 4 input LUTs	115	4,896	2%
Number used as logic	115		
Number used as a route-thru	0		
Number of bonded IOBs	26	108	24%
IOB Flip Flops	16		
Number of BUFGs	1	24	4%
Number of MULT18X18SIOs	1	12	8%
Total equivalent gate count for design	1,103		
Additional JTAG gate count for IOBs	1,248		

Figure 10: Area of Square Nikhilam Sutra obtained from Synthesis Results

CUBE_URDHVATIRYAKBHYAM Partition Summary

No partition information was found.

Device Utilization Summary

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	188	4,896	3%
Logic Distribution			
Number of occupied Slices	99	2,448	4%
Number of Slices containing only related logic	99	99	100%
Number of Slices containing unrelated logic	0	99	0%
Total Number of 4 input LUTs	188	4,896	3%
Number of bonded IOBs	42	108	38%
IOB Flip Flops	32		
Number of GCLKs	1	24	4%
Number of MULT18X18SIOs	1	12	8%
Total equivalent gate count for design	1,402		
Additional JTAG gate count for IOBs	2,016		

Figure 11: Area of Cube UrdhvaTiryakbhyam obtained from synthesis results

CUBE_NIKHILAMSUTRA Partition Summary

No partition information was found.

Device Utilization Summary

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	118	4,896	2%
Logic Distribution			
Number of occupied Slices	60	2,448	2%
Number of Slices containing only related logic	60	60	100%
Number of Slices containing unrelated logic	0	60	0%
Total Number of 4 input LUTs	118	4,896	2%
Number used as logic	115		
Number used as a route-thru	3		
Number of bonded IOBs	42	108	38%
Number of GCLKs	1	24	4%
Number of MULT18X18SIOs	1	12	8%
Total equivalent gate count for design	984		
Additional JTAG gate count for IOBs	2,016		

Figure 12: Area of Cube Nikhilam Sutra obtained from Synthesis Results

Table 1: Comparison of parameters for square architecture

Sl no.	Parameter	Existing Design	Proposed Design
1.	Number of LUT	188	118
2.	Number of slices	99	60
3.	IOB flip flops	32	16
4.	Number of bonded IOBs	42	26
5.	Number of GCLKs	1	1
6.	Power	76mW	54mW

Table 2: Comparison of parameters for cube architecture

Sl no	Parameter	Existing Design	Proposed Design
1.	Number of LUT	188	118
2.	Number of slices	99	60
3.	IOB flip flops	32	16
4.	Number of bonded IOBs	42	26
5.	Number of GCLKs	1	1
6.	Power	90mW	54mW

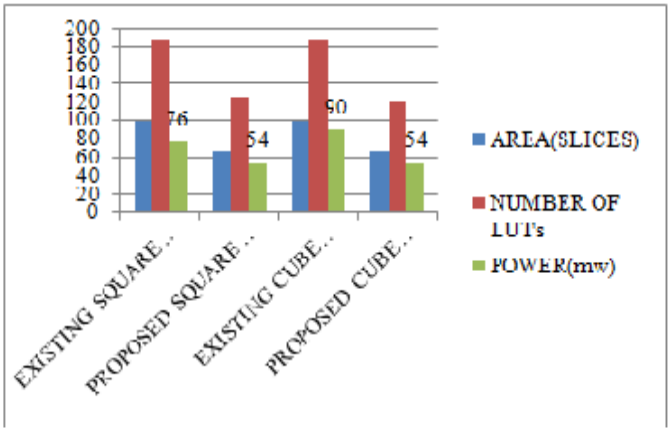

Figure 13: Comparison between conventional and proposed

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

BASYS SPARTAN 3E FPGA results of Square architecture:

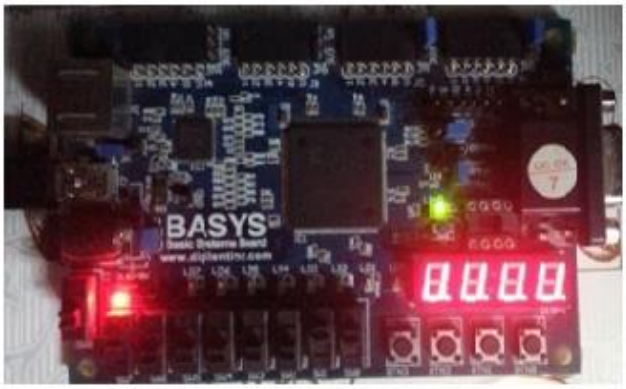

Figure 14: Input given to the board is 3 for a square architecture

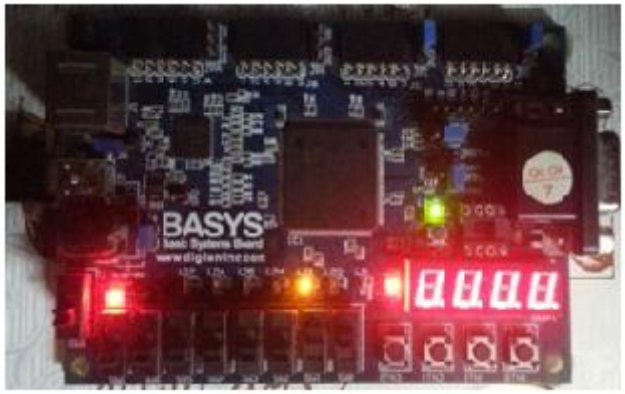

Figure 15: Output obtained on board is 9 for a Square architecture

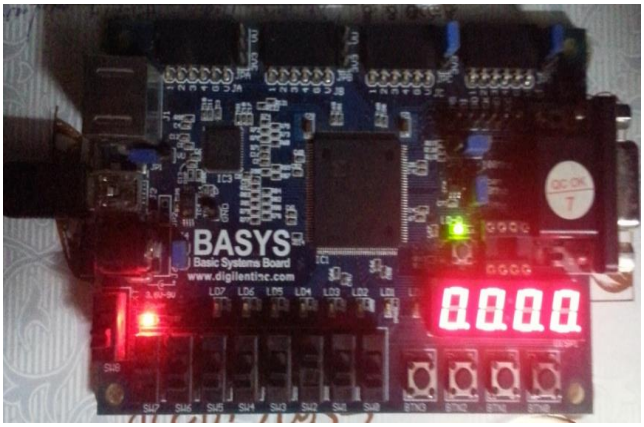

Figure 16: Input given to the board is 4 for cube architecture

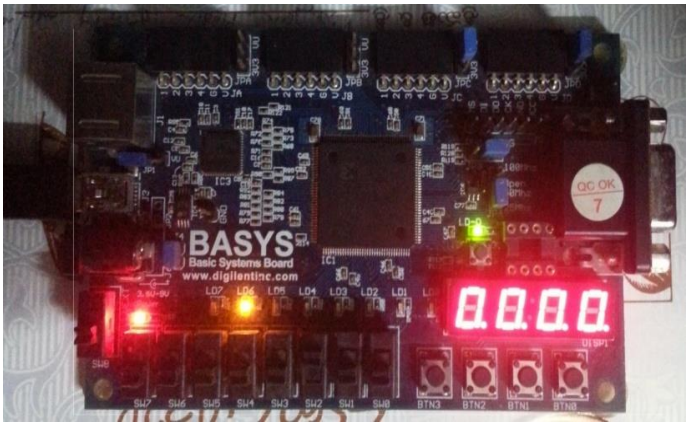

Figure 17: Output obtained from the board is 64 for cube architecture

VII CONCLUSION

The designs and implementation of square and cube architecture using nikhilam sutra have been implemented on Spartan device. And it is observed that the proposed architectures have performed better than the existing architecture and the comparison has been done on parameters like power and area .The power consumption has been reduced when compare to the existing architecture and area consumed is also reduced It is therefore seen that the proposed architecture is much more efficient than the conventional architecture. Square and cube based on Nikhilam sutra are such algorithms which can reduce the power and hardware requirements for multiplication of numbers.

REFERENCES

[1]Jagadguru Swami Sri Bharati Krishna Tirthji Maharaja,"Vedic Mathematics", Motilal Banarsidas, Varanasi, India, 1986.

[2]Vaijyanath Kunchigi, Linganagouda Kulkarni and Subhash Kulkarni 32-BIT MAC UNIT DESIGN USING VEDIC MULTIPLIER – published at: "International Journal of Scientific and Research Publications (IJSRP), Volume 3, Issue2, Feburary 2013 Edition".

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

[3]Krishnaveni D and Umarani T.G. "Vlsi implementation of Vedic multiplier with reduced delay" International Journal of Advanced Technology & Engineering Research (IJATER), Volume 2, Issue 4, July 2012

[4]Bhavin D maru and Altaf Darvadia "VHDL implematation of 8-bit vedic multiplier using barrel shifter" International Journal for scientific research and development(IJSRD), volume 2, Issue1 2014

[5]Gundlaalle Nanda Kishore and k.v. Rajendra Pras ad "Fpga implementation of 8- bit vedic multiplier by using complex numbers" International Journal of engi neering research and application (IJERA), volume 4, Issue6 ,June 2014

[6]Ramachandran.s and Kirti.s.pande "design, implem entation and performance analysis of an intrg rated vedic multiplier architecture "International Journal of computational engineering research (IJC ER), volume 2, Issue3,June 2012

[7] M. Ramalatha, K. D. Dayalan , P. Dharani, and S. D. Priya , "High Speed Energy Efficient ALU Design using Vedic Multiplication Technique ," Lebanon , pp. 600-603, July 2009.

[8] Amandeep Singh "Design and Hardware reali zation of a 16 Bit Vedic Arithmetic Unit" June 2010

[9] M.E.Paramasivam, Dr.R.S.Sabeenian, „„An Effic ient Bit Reduction Binary Multiplication Algorithm using Vedic Methods", IEEE pp 25-28, 2010.