International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
NCICCT-2015 Conference Proceedings

Digital Roots (Dr) Method Verification for Digital Signal Processors Convolution in Vedic Multiplier I/O

C. Karthi
Assistant Professor
Department of Master of Computer Applications
Sri Balaji Chockalingam Engineering College
ACS Nagar, Arni, India

***Abstract*— Very fast multiplication is important computation process in Digital Signal Processing for convolution and fast Fourier transforms etc.The existing Vedic multiplier is based on the Vedic multiplication for fastest implementation using Sutras (formulae). A fastest method for multiplication based on sixteen Vedic sutras, which are actually Sanskrit formulae describing natural ways of solving a whole range of problem with fast and efficient. Vedic multiplier shows a very improved performance over the modified Booth Wallace multiplier. Vedic Multiplier is faster than array multiplier shows a improved performance over the modified Booth Wallace multiplier. Vedic Multiplier is faster than array multiplier for both signed and unsigned numbers. The results show that Vedic multiplier is an extreme fast multiplier and is well ahead of the modified Booth Wallace multiplier. In this paper Digital Roots (DR) method is used to apply the vedic multiplier I/O digital signals for verify the actual received signals in Digital Signal Processors.**

Keywords*— *Digital Signal Processing;Vedic Sutras;Vedic Multiplier;Convolution;Urdhva Tiryakbhyam

I. INTRODUCTION

High speed arithmetic operations are very important in many signal processing applications. Speed of the digital signal processor (DSP) is largely determined by the speed of its multipliers. In fact the multipliers are the most important part of all digital signal processors; they are very important in realizing many important functions such as fast Fourier transforms and convolutions. Since a processor spends considerable amount of time in performing multiplication, an improvement in multiplication speed can greatly improve system performance. Multiplication can be implemented using many algorithms such as array, booth, carry save, and Wallace tree algorithms.

The computational time required by the array - multiplier is less because the partial products are computed - 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. Arrangement of adders is another way of improving multiplication speed. There are two methods for this: Carry save array (CSA) method and Wallace tree method. In the CSA method, bits are processed one by one to supply a carry signal to an adder located at a one bit higher position. The CSA method has got its own limitations since the execution time depends on the number of bits of the multiplier.

In the Wallace tree method, three bit signals are passed to a one bit full adder and the sum is supplied to the next stage full adder of the same bit and the carry output signal is passed to the next stage full adder of same number of bit and the then formed carry is supplied to the next stage of the full adder located at a one bit higher position. In this method, the circuit lay out is not easy.

Booth algorithm reduces the number of partial products. However, large booth arrays are required for high speed multiplication and exponential operations which in turn require large partial sum and partial carry registers. 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 recoded adder stages. Thus, a large propagation delay is associated with this case. The modified booth encoded Wallace tree multiplier uses modified booth algorithm to reduce the partial products and also faster additions are performed using the Wallace tree.

This Paper proposes a verify novel fast multiplier adopting the sutra of ancient Indian Vedic mathematics called Urdhva tiryakbhyam with the help of Digital Root (DR) Method. The design of the multiplier is faster than existing multipliers reported previously. This paper finally verifies and concludes the input and output signal with DR method.

II. DIGITAL SIGNAL PROCESSING (DSP)

Digital signal processing is one of the core technologies, in rapidly growing application areas, such as wireless communications, audio and video processing and industrial control. The number and variety of products that include some form of digital signal processing has grown dramatically over the last few years. Digital signal processors are used for a wide range of applications, from communications and controls to speech and image processing.

III. DIGITALIZATION

Most of the signals directly encountered in science and engineering are continuous: light intensity that changes with distance; voltage that varies over time; a chemical reaction rate that depends on temperature, etc. Analog-to-Digital Conversion (ADC) and Digital-to-Analog Conversion (DAC) are the processes that allow digital computers to interact with these everyday signals. Digital information is different from its continuous counterpart in two important respects: it is sampled, and it is quantized. Both of these restrict how much information a digital signal can contain.

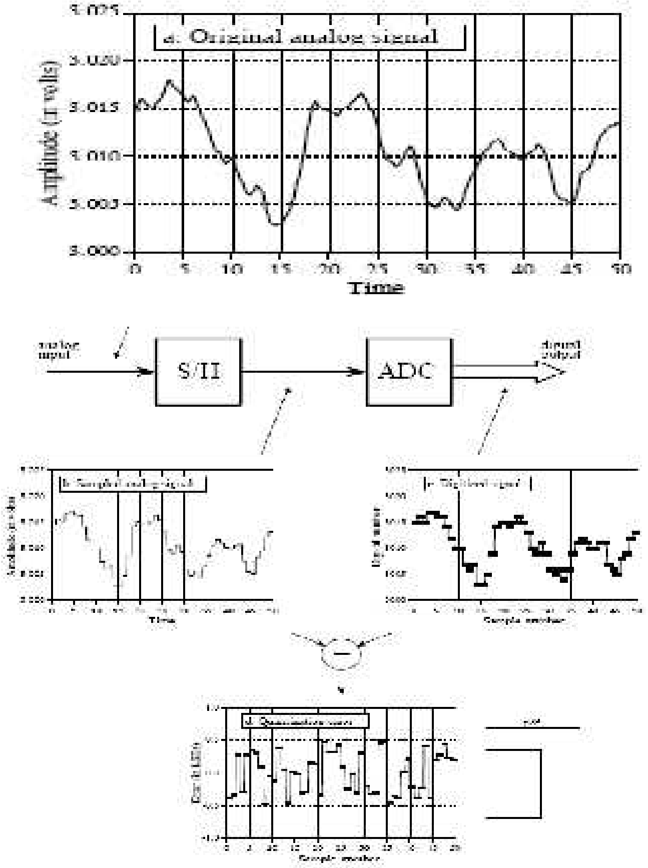


Fig. 1. Waveforms illustrating the digitization process

Fig .1 shown by the difference between (a) and (b), the output of the sample-and hold is allowed to change only at periodic intervals, at which time it is made identical to the instantaneous value of the input signal. Changes in the input signal that occur between these sampling times are completely ignored. That is, sampling converts the independent variable (time in this example) from continuous to discrete.

As shown by the difference between (b) and (c), the ADC produces an integer value between 0 and 4095 for each of the flat regions in (b). This introduces an error, since each plateau can be any voltage between 0 and 4.095 volts. For example, both 2.56000 volts and 2.56001 volts will be converted into digital number 2560. In other words, quantization converts the dependent variable (voltage in this example) from continuous to discrete. Notice that we carefully avoid comparing (a) and (c), as this would lump the sampling and quantization together. It is important that we analyze them separately because they degrade the signal in different ways, as well as being controlled by different parameters in the electronics. There are also cases where one is used without the other. For instance, sampling without quantization is used in switched capacitor filters.

First we will look at the effects of quantization. Any one sample in the digitized signal can have a maximum error of ±½ LSB (Least Significant Bit, jargon for the distance between adjacent quantization levels). Figure (d) shows the quantization error for this particular example, found by subtracting (b) from (c), with the appropriate conversions. In other words, the digital output (c), is equivalent to the continuous input (b), plus a quantization error (d).

An important feature of this analysis is that the quantization error appears very much like random noise. This sets the stage for an important model of quantization error. In most cases, quantization results in nothing more than the addition of a specific amount of random noise to the signal. The additive noise is uniformly distributed between ±½ LSB, has a mean of zero, and a standard deviation of 1/ 12 LSB (-0.29 LSB). For example, passing an analog signal through an 8 bit digitizer adds an rms noise of: 0.29/256 , or about 1/900 of the full scale value. A 12 bit conversion adds a noise of: 0.29 /4096 . 1 /14,000, while a 16 bit conversion adds: 0.29/65536 . 1 /227,000. Since quantization error is a random noise, the number of bits determines the precision of the data. For example, you might make the statement: "We increased the precision of the measurement from 8 to 12 bits."

This model is extremely powerful, because the random noise generated by quantization will simply add to whatever noise is already present in the analog signal. The conversion is broken into two stages to allow the effects of sampling to be separated from the effects of quantization. The first stage is the sample-and-hol (S/H), where the only information retained is the instantaneous value of the signal when the periodic sampling takes place. In the second stage, the ADC converts the voltage to the nearest integer number. This results in each sample in the digitized signal having an error of up to ±½ LSB, as shown in (d). As a result, quantization can usually be modelled as simply adding noise to the signal.

IV.VEDIC MULTIPLICATION ALGORITHMS

A. HISTORY OF VEDIC MATHEMATICS:-

Vedic Mathematics by the late Sankaracarya (BHĀRATĪ KRSNA TĪRTHAJĪ MAHĀRĀJA(1884-1960)) of Goverdhana Pitha is a monumental work. It deals mainly various Vedic mathematical formulae and their applications for carrying out tedious and cumbersome arithmetical operations, and to a very large extent, executing them mentally. In this field of mental arithmetical operations the works of the famous mathematicians Trachtenberg and Lester Meyers (High Speed Math's) are elementary compared to that of Jagadguruji. Some people may find it difficult, at first reading, to understand the arithmetical operations although they have been explained very lucidly by Jagadguruji. It is not because the explanations are lacking in any manner but because the methods are totally unconventional. Some people are so deeply rooted in the conventional methods that they probably, subconsciously reject to see the logic in unconventional methods.

Vedic mathematics is part of four Vedas (books of wisdom). It is part of Sthapatya- Veda (book on civil engineering and architecture), which is an upa-veda (supplement) of Atharva Veda. It covers explanation of several modern mathematical terms including arithmetic, geometry (plane, co-ordinate), trigonometry, quadratic equations, factorization and even calculus.

According to one interpretation, the portions of knowledge encompassing intra-disciplinary science that deals with all branches of mathematics, science, architecture, and engineering, is tied to Atharva Veda. Naturally, the 'mathematical formulae' can be expected to fall under this treatise. Ancient Hindus in the process of constructing sacrificial altars of precise shapes and sizes had to master geometry, arithmetic and algebra that had been encoded into these sutras. S*utra* is the root of the word '*suture'* - the thread that physicians use to tie up a wound.

The word "Vedic" is derived from the word "Veda" which means the store-house of all knowledge. Vedic mathematics is mainly based on 16 Sutras (or aphorisms) dealing with various branches of mathematics like arithmetic, algebra, geometry etc. Although Nikhilam Sutra is applicable to all cases of multiplication, it is more efficient when the numbers involved are large. In addition to this Sutra, Vedic mathematics deals with another multiplication formula, Urdhva tiryakbhyam, which is equally applicable to all cases of multiplication. Attempts have been made in the literature to apply this general multiplication formula to binary arithmetic.

As mentioned earlier, all these Sutras were reconstructed from ancient Vedic texts early in the last century. Many Sub-sutras were also discovered at the same time, which are not discussed here. The beauty of Vedic mathematics lies in the fact that it reduces the 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.

V.URDHVA TIRYAKBHYAM SUTRA

The given Vedic multiplier based on the Vedic multiplication formulae (Sutra). This Sutra has been traditionally used for the multiplication of two numbers. In proposed work, we will apply the same ideas to make the proposed work compatible with the digital hardware.

Urdhva Tiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It means "Vertically and Crosswise". 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 multiplication of two 4 digit numbers is as shown in Fig. 2.

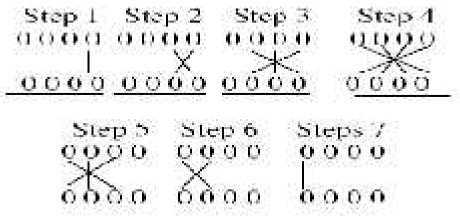


Fig. 2. Line Multiplication for two 4 digit number

For this multiplication scheme, let us consider the multiplication of two decimal numbers (325 × 728). Line diagram for the multiplication is shown in Fig. 3. 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.

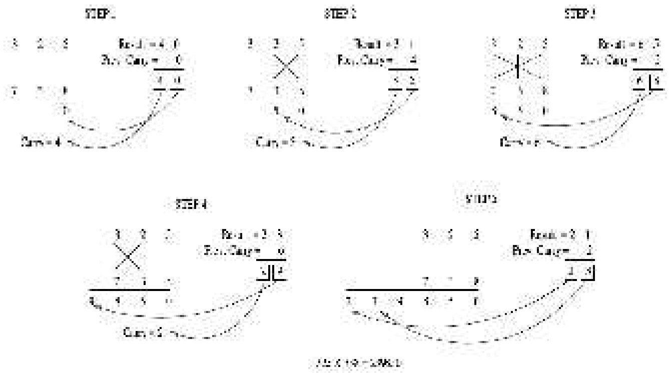


Fig. 3. Multiplication using Urdhva Tiryakbhyam

Now we will extend this Sutra to binary number system. For the multiplication algorithm, let us consider the multiplication of two 8 bit binary numbers A7A6A5A4A3A2A1A0 and B7B6B5B4B3B2B1B0. As the result of this multiplication would be more than 8 bits, we express it as …R7R6R5R4R3R2R1R0. As in the last case, the digits on the both sides of the line are multiplied and added with the carry from the previous step. This generates one of the bits of the result and a carry. This carry is added in the next step and hence the process goes on. If more than one lines are there in one step, all the results are added to the previous carry. In each step, least significant bit acts as the result bit and the other entire bits act as carry. For example, if in some intermediate step, we will get 011, then1 will act as result bit and 01 as the carry. Thus we will get the following expressions:

R0=A0B0
C1R1=A0B1+A1B0
C2R2=C1+A0B2+A2B0+A1B1
C3R3=C2+A3B0+A0B3+A1B2+A2B1
C4R4=C3+A4B0+A0B4+A3B1+A1B3+A2B2
C5R5=C4+A5B0+A0B5+A4B1+A1B4+A3B2+A2B3
C6R6=C5+A6B0+A0B6+A5B1+A1B5+A4B2+A2B4 +A3B3

C7R7=C6+A7B0+A0B7+A6B1+A1B6+A5B2+A2B5 +A4B3 +A3B4
C8R8=C7+A7B1+A1B7+A6B2+A2B6+A5B3+A3B5+A4B4
C9R9=C8+A7B2+A2B7+A6B3+A3B6+A5B4 +A4B5
C10R10=C9+A7B3+A3B7+A6B4+A4B6+A5B5
C11R11=C10+A7B4+A4B7+A6B5+A5B6
C12R12=C11+A7B5+A5B7+A6B6
C13R13=C12+A7B6+A6B7
C14R14=C13+A7B7

C14R14R13R12R11R10R9R8R7R6R5R4R3R2R1R0 being the final product. Hence this is the general mathematical formula applicable to all cases of multiplication. All the partial products are calculated in parallel and the delay associated is mainly the time taken by the carry to propagate through the adders which form the multiplication array. So, this is not an efficient algorithm for the multiplication of large numbers as a lot of propagation delay will be involved in such cases. To overcome this problem, Nikhilam Sutra will present an efficient method of multiplying two large numbers.

A. Nikhilam Sutra

Nikhilam Sutra means "all from 9 and last from 10". It is also applicable to all cases of multiplication; it is more efficient when the numbers involved are large. Since it find out the compliment of the large number from its nearest base to perform the multiplication operation on it. Larger the original number, lesser the complexity of the multiplication. We will illustrate this Sutra by considering the multiplication of two decimal numbers (96 × 93) where the chosen base is 100 which is nearest to and greater than both these two numbers.

As shown in Fig. 4, we write the multiplier and the multiplicand in two rows followed by the differences of each of them from the chosen base, i.e., their compliments. We can write two columns of numbers, one consisting of the numbers to be multiplied (Column 1) and the other consisting of their compliments (Column 2). The product also consists of two parts which are distributed by a vertical line. The right hand side of the product will be obtained by simply multiplying the numbers of the Column 2 (7×4 = 28). The left hand side of the product will be found by cross subtracting the second number of Column 2 from the first number of Column 1 or vice versa, i.e., 96 - 7 = 89 or 93 - 4 = 89. The final result will be obtained by combining RHS and LHS (Answer = 8928).

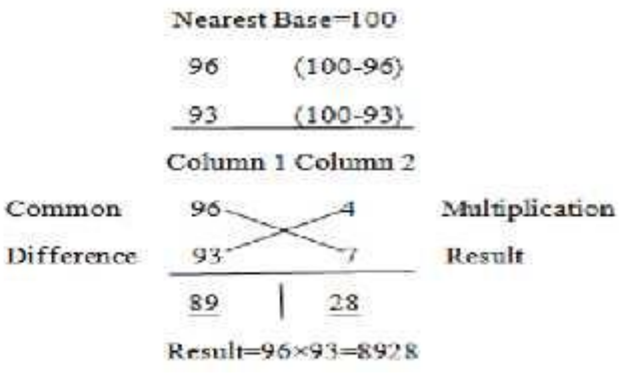


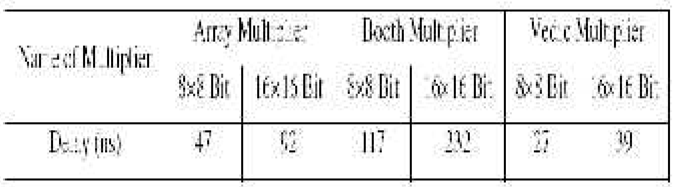
Fig. 4. Multiplication by Nikhilam Method

VI COMPARISION AND DISCUSSION

FPGA implementation results shows that multiplier Nikhilam Sutra based on of Vedic mathematics for multiplication of binary numbers is faster than multipliers based on Array and Booth multiplier. It also proves that as the number of bits increases to N, where N can be any number, the delay time is greatly reduced in Vedic Multiplier as compared to other multipliers(Table I).

Vedic Multiplier has the advantages as over other multipliers also for power and regularity of structures.

Table I. Comparison of Multiplier w.r.t. delay (ns)

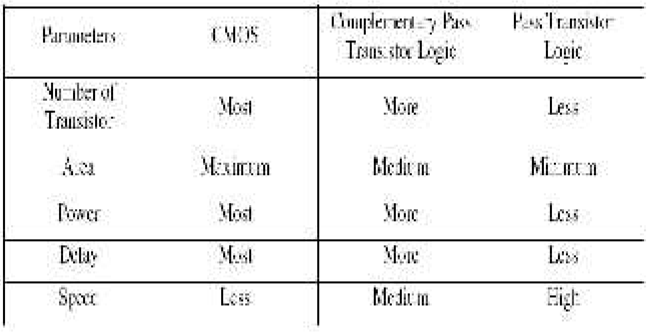


Name of Multiplier	Array Multiplier		Booth Multiplier		Vedic Multiplier	
	8x8 Bit	16x16 Bit	8x8 Bit	16x16 Bit	8x8 Bit	16x16 Bit
Delay (ns)	47	92	117	232	27	39

There are number of techniques for logic implementation at circuit level that improves the power dissipation, area and delay parameters in VLSI design. Implementation of parallel Multiplier in CPL logic shows significant improvement in power dissipation. CPL requires more number of transistors to implement as compared to the CMOS and provides only a little improvement in speed.

Pass Transistor Logic which offers better performance over both the CMOS and CPL in terms of delay, power, speed and transistor count. The PTL outperforms the CMOS implementation in speed and great in power dissipation, with approximately same transistor count. When compared to CPL, PTL is faster and Table II shows improvement in power and transistor count.

Table II. Comparison between multiplier designs in three different Logics

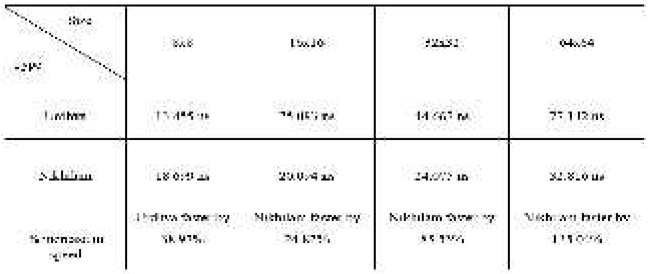


Parameters	CMOS	Complementary Pass Transistor Logic	Pass Transistor Logic
Number of Transistor	Most	More	Less
Area	Maximum	Medium	Minimum
Power	Most	More	Less
Delay	Most	More	Less
Speed	Less	Medium	High

VII. SPEED ANALYSIS REPORT

The designs of 8x8 bits, 16x16 bits, 32x32 bits and 64x64 bits Vedic multiplier have been implemented on Xilinx ISE 11 series for a series of ten multiplicands each, which fall in the Nikhilam range. It is therefore seen from Table III that, on an average, in case of 8 bit multipliers, Urdhva performs better than Nikhilam because of the small size of the multiplicands. However, as the size of the multiplicands increase, Nikhilam performs much faster than Urdhva & achieves an increase of 135.04% (more than twice as fast) in speed for 64 bit multiplicands.

Table III. Speed Anaysis synthesis report



Size / SPE	8x8	16x16	32x32	64x64
Urdhva	17.455 ns	25.053 ns	14.667 ns	23.112 ns
Nikhilam	18.039 ns	20.074 ns	24.072 ns	32.810 ns
Synthesis in speed	Urdhva faster by 56.93%	Nikhilam faster by 24.87%	Nikhilam faster by 55.19%	Nikhilam faster by 115.02%

VIII PROPOSED DIGITAL ROOTS METHOD FOR SIGNAL VERIFICATION

After convert the signals into numbers we can handle mathematical calculation for Convolution process.

A. Summing digits

You may have noticed that with the nine times table all the answers have digits adding up to nine.

1X9 =9
2X9=18, 1 + 8 = 9
3 x 9 =27, 2 + 7 = 9
4 X 9 = 36, 3 + 6 = 9, etc

This brings to the important idea of digital root. The digital roots of any number is the sum of all the digits, continued until there is only one digit left. Here are some examples (Table IV).

Table IV. Examples of Digital roots

Number	Summing Digits		Digital Root
71	7+1 =8		8
231	2+3+1=6		6
85	8+5=13	1+3	4
7562	7+5+6+2=20	2+0	2

It is also a fact that the digital root of a number is the same as remainder when that number is divided by 9,For example, 71 ÷ 9 = 7 remainder 8 and 231 ÷ 9 =25 remainder 6.

The digital root of a number in fact tells us something of the quality of that number and can also help us check answers to many calculations.

B. Casting out nines

An easy method for finding the digital root of any number is to cast out nines and groups of digits which add up to 9. This is done by crossing out any nines in the number or any digits adding up to nine. The numbers which are left at the end are added up for the digital root. The sutra used is By Elimination and Retention.

The only number which is left is 6 and nine and this is the digital root. If there is nothing left after having cast out nines then the digital root is 9. This is because in the process of casting out nought and nine are interchangeable. This also follows from the fact that when dividing any number by 9 and the remainder is 0 it may also be said to be 9, for example, 18 + 9 = 2 remainder 0, or 1 remainder 9.

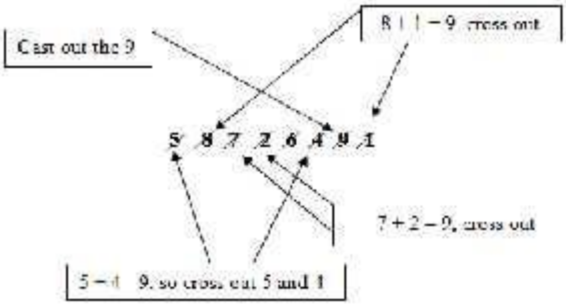


Fig.5. Digital Root of 58726491

C. Using digital roots to check final answer

By casting out nines from all the numbers in any sum the same sum also holds true for the digital roots. To take a simple example, consider the sum, 256 + 174. On adding the two numbers the answer is found to be 430.The digital roots are 4(for 256), 3(for 174) and 7 (for 430), 4 + 3 = 7 provides us with a check to the correctness of out sum. This may be set out as shown on the right:

256 digital root = 4
+174 digital root = 3
____ ____
430 = digital root = 7 addition of (4 + 3) is = 7

When the digital root check is very is very useful for multiplication and division. This is an application of the rule – The product of the sum of the digits in the factors is equal to the sum of the digits in the product. The two numbers to be multiplied are the factors and the answer is called the product.

322 digital root = 7
X 263 digital root = X 2
______ ____
84686 =5 14 digital root = 5
______ ____

IX. CONCLUSION

The time taken for multiplication operation is reduced by employing the Vedic algorithms. Here integrated Vedic multiplier architecture is proposed for further reduction in time. Depending on the inputs, the better sutra is selected by the architecture itself.

The main point of this paper was to introduce a multiplier algorithm applied for input and output signals with Digital Roots Method for verification. The DR method is used to check the digital signals after the convolution process is performed for a particular domain oriented signals.

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 solve by Nikhilam Sutra which reduces the multiplication of two large numbers to the multiplication of two small numbers. Signal verification is very hard to handle the correct signals while send and receive signals verify this method will implement many signal to data conversions then we can use this kind of verification for Digital image processing in future.

REFERENCES

[1] Mr. Abhishek Gupta1, Mr. Utsav Malviya, Prof. Vinod Kapse," A Novel Approach to Design High Speed Arithmetic Logic Unit Based On Ancient Vedic Multiplication Technique",International Journal of Modern Engineering Research (IJMER), www.ijmer.com,Vol.2, Issue.4, pp-2695-2698,ISSN: 2249-6645,July-Aug 2012

[2] V.Vamshi Krishna and S. Naveen Kumar. " High Speed, Power and Area efficient Algorithms for ALU using Vedic Mathematics", International Journal of Scientific and Research Publications,ISSN 2250-3153,Volume 2, Issue 7, July 2012

[3] Pushpalata Verma,:Design of 4x4 bit Vedic Multiplier using EDA Tool",International Journal of Computer Applications (0975 – 888) Volume 48– No.20, PP 32-35,June 2012.

[4] Abhishek Gupta,"Arithmetic Unit Implementation Using Delay Optimized Vedic Multiplier with BIST Capability ",International Journal of Engineering and Innovative Technology (IJEIT),ISSN: 2277-3754,Volume 1, Issue 5, May 2012

[5] Kabiraj Sethi and Rutuparna Panda,"An Improved Squaring Circuit for Binary Numbers",(IJACSA) International Journal of Advanced Computer Science and Applications,Vol. 3, No.2, PP 111-116,2012

[6] Pooya Asadi,"A New Optimized Tree Structure in High-Speed Modified Booth Multiplier Architecture", American Journal of Scientific Research,ISSN 1450- 223X Issue 52 (2012), pp. 48-56

[7] Ramachandran.S and Kirti.S.Pande,"Design, Implementation and Performance Analysis of an Integrated Vedic Multiplier Architecture", International Journal Of Computational Engineering Research, ISSN: 2250–3005,Vol. 2 ,Issue No.3 , PP 697-703, May-June 2012

[8] Asmita Haveliya, "FPGA IMPLEMENTATION OF A VEDIC CONVOLUTION ALGORITHM",International Journal of Engineering Research and Applications (IJERA)ISSN: 2248-9622,Vol. 2, Issue 1, pp.678-684,Jan-Feb 2012

[9] Nidhi Mittal Abhijeet Kumar,"Hardware Implementation of FFT using vertically and Crosswise Algorithm",International Journal of Computer Applications (0975 – 8887),Volume 35– No.1, PP 17-20,December 2011

[10] P. Saha1, A. Banerjee2, A. Dandapat3, P. Bhattacharyya1,"Vedic Mathematics Based 32-Bit Multiplier Design for High Speed Low Power Processors",INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS, VOL.4,NO.2, June 2011

[11] Manoranjan Pradhan,Rutuparna Panda and Sushanta Kumar Sahu,"Speed Comparison of 16x16 Vedic Multipliers",International Journal of Computer Applications (0975 – 8887),Volume 21– No.6, May 2011

[12] Rashmi K. Lomte (Mrs.Rashmi R. Kulkarni), Prof.Bhaskar P.C “Speedy Deconvolution using Vedic Mathematics”, International Journal of Scientific & Engineering Research Volume 2, Issue 5,ISSN 2229 -5518,May 2011

[13] Sree Nivas A and Kayalvizhi N,"Implementation of Power Efficient Vedic Multiplie",International Journal of Computer Applications (0975 – 8887),Volume 43– No.16, April 2012

[14] Asmita Haveliya,"A Novel Design for High Speed Multiplier for Digital Signal Processing Applications (Ancient Indian Vedic mathematics approach)",International Journal of Technology And Engineering System(IJTES):Vol 2,No.1,Jan – March 2011

[15] S. S. Kerur, Prakash Narchi,Jayashree C N, Harish M Kittur and Girish V A "Implementation of Vedic Multiplier for Digital Signal Processing",Proceedings published by International Journal of Computer Applications (IJCA), International Conference on VLSI, Communication & Instrumentation (ICVCI), 2011

[16] Rana Mukherji,Amit Kumar Chatterjee and Manishita Das,"Implementation of an efficient multiplier architecture based on ancient indian Vedic mathematics using System C",KIST Journal of Science and Technology,(KJST) PP 47-57,Volume 1 Number 1,2011

[17] Sumit Vaidya and Deepak Dandekar,"DELAY – POWER PERFORMANCE COMPARISON OF MULTIPLIERS IN VLSICIRCUIT DESIGN",International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.4, July 2010

[18] N. Sreenivas, Prof. K Ashok Babu,"Design and Implementation of VLSI Architecture for Different Memory Reduction Methodologies in DWT/IDWT ",Journal of Current Computer Science and Technology, Vol. 1,Issue 5,PP 184-202,2011

[19] Ashish Raman, Anvesh Kumar and R.K.Sarin,"High Speed Reconfigurable FFT Design by Vedic Mathematics",JOURNAL OF COMPUTER SCIENCE AND ENGINEERING, VOLUME 1, Issue 1, May 2010

[20] Harpreet Singh Dhillon and Abhijit Mitra,"A Reduced-Bit Multiplication Algorithm for Digital Arithmetic",International Journal of Mathematics Sciences,Volume 2,Number 2,Mar 01, 2008