

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2014

Designing Of Fast Multipliers with Ancient Vedic Techniques

Jeevitha.R¹, Jayanthi².P, Gowthami A.K³, Uvarajan K.P⁴

Assistant Professor, Dept. of ECE, Bannari Amman Institute of Technology, Sathyamangalam, India^{1,2,3}

Assistant Professor, Dept. of ECE, KSR College of Engineering, Erode, India⁴

ABSTRACT: Vedic mathematics is an ancient system of mathematics which performs unique technique of calculations based on 16 sutras. The performance of high speed multiplier is designed based on Urdhva Tiryabhyam, Nikhilam Navatashcaramam Dashatah, and Anurupye algorithms. These algorithms gives minimum delay and used for multiplication of all types of numbers. The performance of high speed multiplier is designed and compared using these sutras for various NxN bit multiplications and implemented on the FFT of the DSP processor. Anurupye Vedic multiplier on FFT is made efficient than Urdhva tiryabhyam and Nikhilam Navatashcaramam Dashatah sutras by more reduction in computation time. Logic verification of these design is verified by simulating the logic circuits in XILINX ISE 9.1 and MODELSIM SE 5.7g using VHDL.

KEYWORDS: Urdhva Tiryagbhyam, Nikhilam Navatashcaramam-Dashatah, Anurupye, Vedic multiplier.

I. INTRODUCTION

Multiplication is an important function in arithmetic operations. Multiply and Accumulate(MAC) is one of the frequently used Computation- Intensive Arithmetic Functions(CIAF) that are implemented in many Digital Signal Processing (DSP) applications such as convolution operation, filtering, Fast Fourier Transform(FFT) and in microprocessors in its arithmetic and logic unit(ALU) [2].

The demand for high speed processing has been increasing as a result of expanding computer and signal processing applications. High throughput arithmetic operation is important to achieve the desired performance in many image and real time processing applications. One of the arithmetic operations in such application is multiplication and the development of fast multiplier circuit. Reduction of both the time delay and power consumption is very essential requirements for many applications [3].

Multiplier design based on Vedic Mathematics is one of the fast and low power multiplier. Minimizing power consumption for digital systems is important and it involves optimization at all levels of the design. This optimization includes one technology that is used to implement the style, topology and the architecture of digital circuits.

II. VEDIC MATHEMATICS

Vedic mathematics is the part of four Vedas. Mainly it is a part of Sthapatya- Veda which is an upa-veda and a supplement of Atharva Veda. It gives various mathematical terms including arithmetic, geometry, trigonometry, factorization, calculus, etc. In the period of 1884-1960, Jagadguru Shankaracharya Bharati Krishna Teerthaji Maharaja comprised all this work together and gave its mathematical explanation while it is used for various applications. After extensive research in Atharva Veda, Swamiji constructed 16 sutras. But it is not found in present text of Atharva Veda because these formulae were constructed by Swamiji itself [6].

Vedic mathematics is not only a mathematical wonder but also it is used for logical operations. Vedic mathematics deals with various branches of mathematics like arithmetic, algebra, geometry etc. These methods and ideas are directly applied to geometry, trigonometry and applied mathematics. This is an interesting field and it gives some effective algorithms that can be applied to various branches of engineering such as computing and digital signal processing (DSP) [6].



(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2014

A large amount of work has been done by understanding various methodologies of sutras. The 16 sutras are as following:

- Chalana-Kalanabyham: Denotes "Differences and Similarities operation".
- Ekanyunena Purvena: Denotes "By one less than the previous one operation".
- Gunitasamuchyah: Denotes "The product of the sum is equal to the sum of the product operation".
- Paraavartya Yojayet: Denotes "Transpose and adjusted operation".
- Sankalana- vyavakalanabhyam: Denotes "By addition and subtraction operation".
- Shunyam Saamyasamuccaye: Denotes "If the sum is same then it will be zero operation".
- Urdhva-tiryagbhyam: Denotes "Vertical and crosswise operation".
- Yaavadunam: Denotes "An extent of its deficiency operation".
- (Anurupye) Shunyamanyat: Denotes "If one is in ratio, the other is zero operation".
- Ekadhikina Purvena: Denotes "By one more than the previous one operation".
- Gunakasamuchyah: Denotes "Factor of the sum is equal to sum of the factor operations".
- Nikhilam Navatashcaramam Dashatah: Denotes "All from 9 and last from 10 operations".
- Puranapuranabyham: Denotes "A complete or non complete operation".
- Shesanyankena Charamena: Denotes "The remainders by the last digit operation".
- Sopaantyadvayamantyam: Denotes "An ultimate and penultimate operation".
- Vyashtisamanstih: Denotes "The Part and Whole operation".

III. DESIGN OF HIGH SPEED VEDIC MULTIPLIER IN FFT

The Vedic multiplier which is designed based on "Urdhva Tiryagbhyam" sutra (algorithm) is a general multiplication formula applicable to all cases of multiplication and it means that "Vertically and Crosswise" and it is used for the multiplication of two decimal numbers. The advantage of this sutra is that it is based on novel concept through which the generation of all partial products can be done with the concurrent addition of partial products. This algorithm can be generalised for the design of n x n bit numbers. The multiplier is independent of clock frequency of the processor while calculating the partial products and their sums in parallel. Nikhilam Navatashcaramam Dashatah Vedic mathematical algorithm is done by "All from 9 and last from 10" and it is more efficient for large numbers. Anurupye sutra is done by "if one is in ratio, then the other is in zero operation".

Digital signal processing (DSP) is the fastest growing technology that is important in almost all engineering discipline. Therefore, it possesses tremendous challenges to the engineering community. Fast operation of additions and multiplications are of extreme importance in DSP for digital filters, discrete Fast Fourier transforms etc..,

A.NIKHILAM NAVATASHCARAMAM DASHATAH SUTRA

Nikhilam Sutra means "all from 9 and last from 10" multiplication operation. It is applicable for all cases of multiplication but it is more efficient for large numbers. Small numbers are also possible but complexity was increased. The multiplication operation is performed by found out the compliment of the large number from its nearest base. Hence the complexity is lesser for larger original number multiplication.

Nikhilam Sutra performs subtraction of a two number from its nearest power base i.e. 10, 100, 1000, etc. The difference is calculated from the power of 10 is called as Base. It is seen that the difference between the base and the number is Positive and hence it is called as NIKHILAM. The multiplication of two 8-bit numbers is reduced to the multiplication of their compliments and addition is shown in Fig.1



(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2014

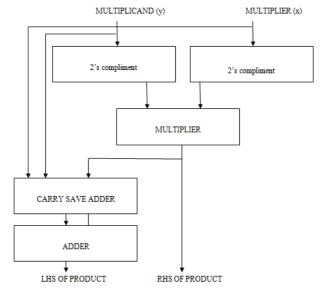
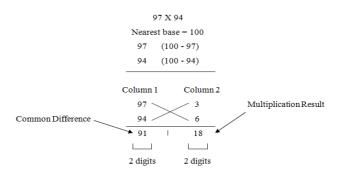
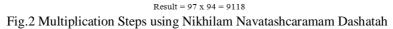


Fig.1 Structure of Nikhilam Navatashcaramam Dashatah Vedic Multiplier

The Nikhilam sutra is applicable to both binary and decimal number system. The subtraction result of multiplicand and multiplier is represented by taking 2[°]s compliment of those two numbers. The right hand side (RHS) part of the product is implemented using 8x8 bit multiplier. The left hand side (LHS) part of the product is implemented using 8-bit carry save adder. If there is carry in RHS of product then it is added to LHS of product. The multiplication of 97 and 94 using Nikhilam sutra is shown in Fig.2.





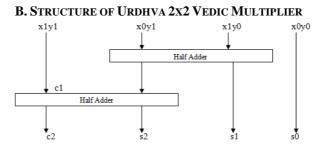


FIG.3 STRUCTURE OF URDHVA 2x2 VEDIC MULTIPLIER



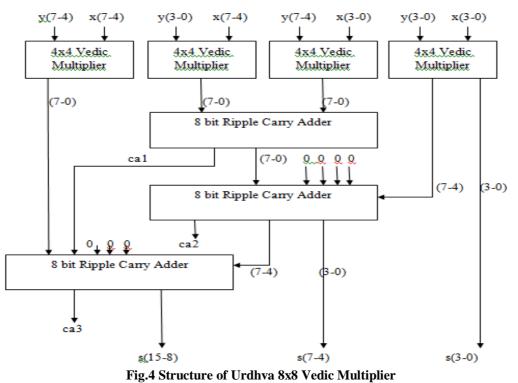
(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2014

The method is described for both the 2 bit numbers X and Y where X = x1x0 and B = y1y0 as shown in Fig 3. In the first step the least significant bit of the final product (vertical) is obtained by multiplication of least significant bits of two numbers. In the second step the LSB of the multiplicand is multiplied with next higher bit of the multiplier. Then it is added with the multiplier product of LSB and next higher bit of the multiplicand (crosswise). Likewise this process is to be done concurrently.

The sum is obtained in second bit of the final product and the carry is added with partial product. By multiplying the most significant bits gives sum and carry. Digital multiplier architecture is developed by applying Urdhva tiryakbhyam Sutra to the binary number system. This is similar to the popular array multiplier architecture.

Another important multiplication algorithm is Booth multiplication. For high speed application large booth array is required and it provides the fast process of multiplying two numbers. Delay is the only time for the signals to propagate through the gates and the multiplication array is formed. An array multiplier is less economical because it requires a large no gates.



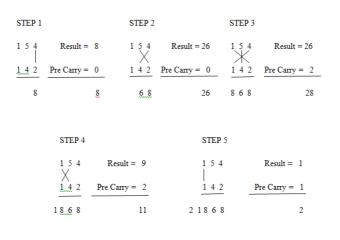
C. Multiplication of Two Decimal Numbers

The Urdhva Sutra is illustrated by consider the multiplication of two decimal numbers 154×142 by Urdhva-Tiryakbhyam method as shown in Fig 5. The result is obtained by multiplying the digits on both sides of the line and added with the carry from the previous step. This generates result for one of the bits and carry. This process goes on by adding carry in the next step. If more than one line is present in one step then all the results are added to the previous carry. In every step, least significant bit (LSB) acts as a result bit and all other bits act as carry for the next step. Initially the value of carry is zero.



(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2014



154 × 142 = 21868 Fig.5 Multiplication Steps using Urdhva

D. ANURUPYE (SHUNYAMANYAT) SUTRA

The upa-Sutra Anurupye means 'proportionality'. This Sutra is used to find products of two numbers. The Common bases are 50, 60, 200 etc (multiples of powers of10).

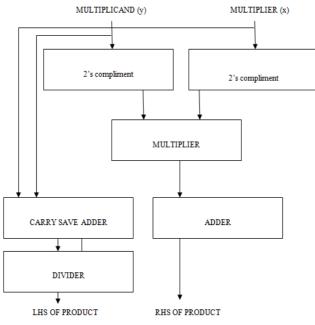
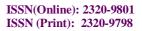


Fig.6 Structure of Anurupye Vedic Multiplier

The Anurupye sutra is illustrated by consider the multiplication of two decimal numbers 58 x 48 by Anurupye method as shown in Fig 7.





(An ISO 3297: 2007 Certified Organization)

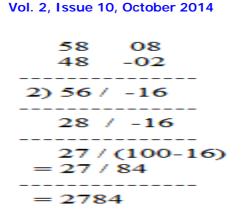


Fig.7 Multiplication Steps using Anurupye

V. RESULTS AND DISCUSSIONS

The Output window for 8x8 Anurupye Vedic Multiplier which was implemented in FFT is shown in Fig.8. It perform multiplication of two decimal numbers 58 x 48 by Anurupye method.

• •	n • 📽 🖬 🛸 🖴	1 X B B 20	2 0 . 4 8: 3	⊊•¥•∰]••		*E1_mm[] \$				
Norm Norm <th< th=""><th>9 * * + E</th><th>100 ps 🛃 🖺</th><th>E B B C I</th><th>10 C</th><th></th><th></th><th></th><th></th><th></th><th></th></th<>	9 * * + E	100 ps 🛃 🖺	E B B C I	10 C						
Norm Norm <th< th=""><th>191110</th><th>Layout Size</th><th>ulate 💌</th><th>ColumnLayout De Eau</th><th>15.</th><th>3</th><th>- B-1</th><th>8·12-18-4</th><th>F</th><th></th></th<>	191110	Layout Size	ulate 💌	ColumnLayout De Eau	15.	3	- B-1	8·12-18-4	F	
Norm Norm Norm Other State Other	1000	u 🖉 💽	종 표시 🗗 🕹 🕹	1.25425.	f 3+ + +€ +	3- Search:		→		
Multice Mups 2 /ffl_design/st 0 0 /ffl_design/st 0 0 /ffl_design/st 0 0 /ffl_design/st 0 0 /ffl_design/st 1011001100110011001100110111 0 /ffl_design/st 1011001100110011001100110111 0 /ffl_design/st 101100110011001100110011011 0 /ffl_design/st 101100110011001100110011011 0 /ffl_design/st 101100110011001100110111 0 /ffl_design/st 101100110011001100110111 0 /ffl_design/st 101100110011001100110111 0 /ffl_design/st 101100110011001100110111 0 /ffl_design/st 1011001100110011001100110111 0 /ffl_design/st 10110011001100110011001101101 0 /ffl_design/st 1011001100110011001100110011001100100 0 /ffl_design/st 1011001000001001110010010000000000000		I I I								
Multice Mups 2 /ffl_design/st 0 0 /ffl_design/st 0 0 /ffl_design/st 0 0 /ffl_design/st 0 0 /ffl_design/st 1011001100110011001100110111 0 /ffl_design/st 1011001100110011001100110111 0 /ffl_design/st 101100110011001100110011011 0 /ffl_design/st 101100110011001100110011011 0 /ffl_design/st 101100110011001100110111 0 /ffl_design/st 101100110011001100110111 0 /ffl_design/st 101100110011001100110111 0 /ffl_design/st 101100110011001100110111 0 /ffl_design/st 1011001100110011001100110111 0 /ffl_design/st 10110011001100110011001101101 0 /ffl_design/st 1011001100110011001100110011001100100 0 /ffl_design/st 1011001000001001110010010000000000000	Wave - Default									_
Imil.descyt.dk 1 0 0 mil.descyt.dk 0 0 0 0 0 0 mil.descyt.dk 0 0 0 mil.descyt.dk 0 0 0 0 0 </th <th>S1-</th> <th>1 1</th> <th>Auros I</th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th> <th></th>	S1-	1 1	Auros I							
Control Display Mit Display Mit <thdisplay mit<="" th=""> <thdisplay mit<="" th=""> <th< td=""><td>A Itt designable</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></th<></thdisplay></thdisplay>	A Itt designable									
Image: Seguride 011001101010 0110011011010 0110011010100 Image: Seguride 011001101010 011001101010 0110011010100 Image: Seguride 011001101010 011001101010 011001101010 0110011010100 Image: Seguride 01100110100 01100110100 01100110100 01100110100 01100110100 Image: Seguride 01100110100 01100110100 01100110100 01100110100 01100110100 Image: Seguride 01100110100 01100110100 01100110100 01100110100 01100110100 Image: Seguride 01100110100 01100110100 01100110100 0110011000 0110011000 Image: Seguride 01100110100 0110011000 0110011000 0110011000 Image: Seguride 0110011000 0110011000 0110011000 0110011000 Image: Seguride 0110011000 0110011000 0110011000 0110010000 Image: Seguride 0110011000 0110011000 010000000000000 010000000000000000000000000		0								
Bits Mill Bits Dist Dist <thdist< th=""> Dist Dist <thd< td=""><td></td><td>101100130110</td><td>1</td><td>1101101100110111</td><td></td><td></td><td></td><td></td><td></td><td></td></thd<></thdist<>		101100130110	1	1101101100110111						
Bit Mill Session Dittopic 110010110010 Dittopic 110010110010 Dittopic 11001010010 Mill Session Dittopic 110010110010 Dittopic 1100101000 Dittopic 11001000000 Dittopic 1100100000000000000000000000000000000	1 /fft_design/x1	301100110110	1-1011001101101101	1101101100110111						
b M1_seep/vid 011001 0000.00001 00110100 011001 0000.00001 00110100 011001 0000.00001 00110100 C M1_seep/vid 011001 00100.00001 001101001 001001 0000001 00110100 0000001 00110100 0000001 00110100 0000001 00110100 0000001 00110100 0000001 0011000 0000001 001100 0000001 001100 0000001 001100 0000001 001100 0000001 001100 0000001 001100 0000001 001100 0000001 001100 0000001 001100 0000001 001100 0000001 001100 000000000000000000000000000000000000	11 design/k2	101111110110	Internation	1101101100110111						
No. Mil. desegn/di Mil. 1000 101 101 101 1001 101 1101 1000 1000000	/fft_design/k3	011100110110	0111001101101100	1101101100110111						
b M1_encycle 000000111110101. 00000011110101. 00000011110101. 00000011110101. 00000011110101. 0000001111001. 0110110110101. 0000001111001. 0110110110101. 0000001111000001. 011011011001. 011011011001. 011011011001. 011011011001. 011011011001. 011011011001. 01101011001. 01101011001. 01101011001. 01101011001. 01101011001. 01101011001. 01101011001. 01101011001. 01101011001. 01101011001. 01101011001. 01101011001. 01101011001. 0110101010000. 0110101010000. 0110101010000. 01101010100000. 01101010100000. 01101010100000. 011010100000. 01101010100000. 0100101100000. 000000000000000000000000000000000000				1101101100110111						
D M1_design/h0 I013001101111. I013001101111001001101101 D M1_design/h0 I0130011000. I0110111001001010011001 D M1_design/h0 I0130011000. I01101110010001000100 D M1_design/h0 I0130011000. I011011000000100 D M1_design/h0 I0130011000. I01100110000000110 D M1_design/h0 I00300010000100001000000010000000000000		101100130110	1 1011001101101100	1401101100000001						
b Mtl. design/vid 0110111000001.010110010010000000 0 Mtl. design/vid 001000010111.0001000000010011001001000000	D 4 /fft_design/k6	000000110110	0000001101101100	1101101100110111						
Discrete Mill desprivid 001810001111000 001810001111000000110000011000001100000110 Discrete Mill desprivid 010000010111 0100000101110000001100000110 Discrete Mill desprivid 01000001011 01000001011 Discrete Mill desprivid 01000001011 01000001011 Discrete Mill desprivid 0000010101000000000000000000000000000		101100110111	1011001101111100	1101101100110111						
n. fml_design/v2 0100000010111 010000001011 010000001011 010000011000000011001001000000011000000		301301130003	1011011100010101	1 10 1 100 1 100000 10						
Control 1100100000000000000000000000000000000			and the second second	1 100 11 100 10 10 110						
No. Nov 1400 ps 000 ps 1000 ps 000 ps 1000 ps 100 ps </td <td></td> <td></td> <td></td> <td></td> <td></td> <td>-</td> <td>_</td> <td></td> <td></td> <td></td>						-	_			
B - Grading 1000001100000 1000001100000 1000001100000 B - Grading Millione 1000001100000 1000001100000 B - Millione 1000001100000 1000001100000 1000001100000 B - Millione 10001001100 10001011001001001100100 10001011000000000000 B - Millione 1000100110010011001001100100000000000			ALL PLACE AVERAGE AV							
D MT_design/vf 0000011100000 0000011100011110011111001 D MT_design/vf 011010001111 01101000011110000101010101010101010101										
D MIL Security/2 O11030000111.00000 [0 101101 000000] O110100000 [0 101101 00000] D MIL Security/2 O11001000111.00000 [0 10101 00100] O11001000111.00000 [0 10100] O110010001100000 D MIL Security/2 O11001001101.0010 [0 10100] O110010001100000 O11001000000 O11001000000 O1100100000000000000000000000000000000										
D										
Do MT, design ML 01.000 101 1001 1.00 1.00 100 100 00 10.000 11.0000 Do MT, design ML 01.000 100 100 100 100 100 100 100 100 1										
Discourting 130000 20130313 110000 10110110001001101100 Now 1400 ps 600 ps 900 ps 9000 ps 1000 ps 1400 ps Currier 1 0 ps 600 ps 900 ps 900 ps 1000 ps 1400 ps										
Nov 1400 ps 600 pil 600 pil 600 pil 1000 ps 1 200 ps 1 200 ps										
Nonv 1400 ps 600 pi 800 pi \$000 pi 1200 pi 1200 pi 1400 pi	D /m design/s2	3 20000 10 1 10 1	1100001011011000	1011011001101110						
A Zeror 1 Opi	COMPANIES.	Nov 1400	os Internet	dimension and the second	THE REAL PROPERTY AND	Charles and the state		A CONTRACTOR OF THE OWNER		
		Contraction and Contraction		co bi	900 pe	1000 [s		1300 fe		1400 b
	<u>.</u>	new 1 C								

Fig.8 Simulation Output window for 8x8 Anurupye Vedic Multiplier which implemented in FFT

The Output window for 8x8 Urdhva Vedic Multiplier which was implemented in FFT is shown in Fig.9. It perform multiplication of two decimal numbers 154 x 142 by Urdhva-Tiryakbhyam method



(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2014

File Edit View Compile	e Simulate Add	Wave Tools Layout Bookmark	s Window Help			
B	K BB 001	0.AE 0 # 28	B + +++	100 ps - EL	BBSSIDE	
				(and) and an (E) and	1	
191 191	Layout Simula	te 💌 ColumnLayout	Default	¥	8.8.94.4	
I 0 10 0 M	1 1 1 4	NALD LLETT	151 3.	+€ - 3+ Search:	↓ 10.02 00-	
				Sector Sector Sector	100 C 17 K	11
	F.F.					
Wave - Default						
S	Hara		00071			
Second Second Second Second	POPPOPULATION POPULATION					
D-> /ff_design/u1 D-> /ff_design/u2	1101110001100	1 10 11 1000 1000 111 111 1000 10 111 10 00000 1 10 100 10				
D /ff_design/u2 D /fff_design/u3	000000101000	000000101001010101010100000				
G-S /M_design/u4	101000010101010	10 10000 10 10 10 10 10 10 10 10 10 10 1				
C-> //ft_design/u5	0101000100101	0 10 1000 100 10 10 10 10 10 10 10 1000 110				
D-5 /m design/u6	0100011010011	0100011010011000001100111001001				
	0011001100010	001100110001001010101000000011				
n //ft_deagn/inter	0000101101001111	9990101101001111				
- //ft_design/inter		11111010010110000				
//ft_design(Inter0	000000000000000000000000000000000000000	000000000000000				
🖬 🌍 //ft_design/inten@	1111000000000000	1111000000000000				
//ft_design/vi)	3011110011000	1011110011000100111000010011010	Q			
🖬 🧼 //fft_ifesign/v1	1101110001100	110111000130011111111000101011110			sim:/fft_design/	
Int_design/v2	0000011010015	0000011010010110101010100000	Contraction in contraction of the local division of the local divi		1111000000000000	,
Int_design/v3	0010001101011	00100011010110110001001000000101				
C → /fft_design/v4	1010000101010	10 10000 10 10 10 1 10 10 10 1 1000 1 10 1 10				
/fft_design/v5 /fft_design/v5	0101000100101	0 10 1000 100 10 1 1 100 10 1 10 1 10000 1 10				_
Int_stellingtyvo Int_design/v7	0100011010011	0100011010011000110011100110011				
0-3 /ftt_design/vr3	1010001000110	10 1000 1000 10 10 10 110 1111 11111111				
Incleage/vrs (fft_design/)vr3	0000000 100 100	0000000 100 100000 10 1 1000 00000000				
G //ft_deugn/ut7	0110001001111	011000100111101001011111111111				
D-> /fft_design/bvr7	0010111111010	0010111111010000111000000000				
No. of Concession, Name	_					
Now		600 ps	800 ps	1000 ps	1200 ps	1400 ps
	0 pe					

Fig.9 Simulation Output window for 8x8 Urdhva Vedic Multiplier which implemented in FFT

The Output window for 8x8 Nikhilam Vedic Multiplier which was implemented in FFT is shown in Fig.10. It perform multiplication of two decimal numbers 97 x 94 by Nikhilam method.

8-00-00 91:00 1:00 1:00 1:00 1:00 1:00 1:00 1	100 pe 🛊 🖹 🖹			ColumnLayou		≫ 3+ 9€		•=====================================	-	- 41	
101010	× 1 9 1	nd I 🗛	1 1	チモチ	255	3+ - +€ -	🔂 Search	-	* #	100.00	
	J.J										
Wave - Default						-					-
-	Huge										
/fft_design/clk	1		1		<u>ا</u>						-
Ch-4 /IIIL_blesign/w0	1010101101010			1101010101010					_		
 /fft_design/v1 	1110110101010			0101010101010							
D-2 /fft_design/v2 D-2 /fft_design/v3	1011100110101			10 10 10 10 10 10 10 10 10 10 10 10							
D-2 /m_design/k4	1010101111111			00000010101010							
m-4 /fft_design/v5	0001111010100			0101010101010		_					
CI-4 /Tit_design/vti	10 10 10 10 10 100			111111101010							
C 4 /m design/x7	10 10 10 10 10 10 L	1010101010	10101010	00111101010	90						
🖬 📥 /m_design/y0	0001110011100	0001110011	1000 1000	01101010011		_	_			_	
El-🖕 /Ill_stesign/y1	1001100000018			011000001100							
EI- 🗶 ///Lutenign/yZ	000011010111			01110000110							
D 4 ///Likelign/y3	1111000101111			00100111000							
m. /m.design/y4	6351201001818			1111100000000							
	1011001001000			111010011100							
Image: Although and Although	130000013001			110011111010							
D- /fft_design/s0	0101011101001			11011010101010							
m_hesign/v1	000010111110			1010101010101							
m/mtesign/s2	0 1 100 1000 100 1			10 10 100 10 100							
THE PARTY PARTY	rev 1600 ps	2000 Distance	800	and other point	100	G ges		O pe	1400 :		00 B
Cursor	A Des	-	1000		100	1.1.1					

Fig.10 Simulation Output window for Nikhilam Vedic Multiplier which implemented in FFT



(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 10, October 2014

The performance analysis of vedic multiplier with high speed and low delay are explained as in Table No.1.

Vedic Multipliers Design in FFT	Time Constraint(delay)
8x8 Multiplier using Nikihilam sutra	25.196 ns
8x8 Multiplier using Urdhva sutra	20.844 ns
8x8 multiplier using Anurupye sutra	17 ns

Table No.1 Performance of Vedic Multiplier in FFT

V. CONCLUSION

Using Urdhva tiryabhyam and Nikhilam Navatashcaramam Dashatah sutras, multipliers were designed. Anurupye Vedic multiplier on FFT is made efficient than Urdhva tiryabhyam and Nikhilam Navatashcaramam Dashatah sutras by more reduction in computation time. Anurupye is used for multiplication of larger numbers.

REFERENCES

- 1. Manoranjan Pradhan, Rutuparna Panda, Sushanta International Journal of Computer Applications. Kumar Sahu (2011),"Speed Comparison of 16x16 Vedic Multipliers" published in
- 2. Umesh Akare, T. V. More, R. S. Lonkar (2012),"Performance Evaluation and Synthesis of Vedic Multiplier" published in International Journal of Computer Applications.
- 3. Pushpalata Verma, K. K. Mehta (2012), "Implementation of an Efficient Multiplier based on Vedic Mathematics Using EDA Tool" published in International Journal of Engineering and Advanced Technology.
- 4. P. D. Chidgupkar and M. T. Karad (2004), "The Implementation of Vedic Algorithms in Digital Signal Processing", Global J. of Engg. Edu., vol.8, no. 2, pp. 153–158.
- 5. Kumaravel.Š, Ramalatha Marimuthu(2007), "VLSI Implementation of High ICCIMA, vol. 4, pp.126-128, International Conference on Computational Intelligence and Multimedia Applications (ICCIMA).
- 6. Harpreet Singh Dhilon (2008), "A Reduced-Bit Multiplication Algorithm for Digital Arithmetic", International Journal of Computational and Mathematical Sciences.
- 7. Anvesh kumar, Ashish raman (2009),"Low Power, High Speed ALU Design by Vedic Mathematics "Publish in National conference organized by NIT, hamirpur.
- 8. Kerur.S, Prakash Narchi, Jayashree C N, Harish M Kittur and Girish V.A (2011), "Implementation of Vedic Multiplier for Digital Signal Processing" International conference on VLSI communication & instrumentation (ICVCI).
- Ramalatha, M.Dayalan, K D Dharani, P Priya, and S Deoborah (2009), "High Speed Energy Efficient ALU Design using Vedic Multiplication Techniques", International Conference on Advances in Computational Tools for Engineering Applications (ACTEA) IEEE, pp. 600-603, July 15-17.
- H. Thapliyal and M. B. Srinivas (2004), "High Speed Efficient N £ N Bit Parallel Hierarchical Overlay Multiplier Architecture Based on Ancient Indian Vedic Mathematics", Enformatika Trans., vol. 2, pp. 225-228.
- H. Thapliyal, R. V. Kamala and M. B. Srinivas (2005), "RSA Encryption/Decryption in Wireless Networks Using an Efficient High Speed Multiplier", in Proc. IEEE Int. Conf. Personal Wireless Comm. (ICPWC-2005), New Delhi, pp. 417–420.

BIOGRAPHY

Ms.R.Jeevitha has received BE degree in Electronics and Communication from Anna University, Coimbatore, Tamil Nadu (2011). She received her ME Degree in Embedded Systems from Anna University, Chennai, Tamil Nadu (2013). She is working as an Assistant Professor in Department of Electronics and Communication, Bannari Amman Institute of Technology, Sathyamangalam, Tamil Nadu for the past 1.6 Years.