# Implementation of High Speed FFT using Reversible Logic Gates for Wireless DSP Applications

V.Jeya Ramya, Associate Professor Panimalar Institute of Technology Chennai A. SELVARANI Associate Professor Panimalar Institute of Technology Chennai, L.Ashok Kumar Associate Professor Panimalar Institute of Technology Chennai J.Navarajan Associate Professor Panimalar Institute of Technology Chennai

#### Abstract

The proposed FFT is constructed using Modified Reversible Logic Gates logic using Ripple carry Adder and its performance is compared with the existing Multiplier Circuits. This type of adders can be applied in the field of digital image processing and signal processing where importance is given to accuracy. The design will be implemented and simulated using Xilinx-ISE-Simulator and performance will be tested. The proposed Reversible Logic Gates logic implemented in Ripple carry adder system using xor logic has an accuracy of 99.98% and power consumption has reduced by more than 75%. Implementation of Modified Reversible Logic Gates logic using xor logic is done using the Xilinx-ISE-simulator tool. Using this adder we have designed an Multiplier based on the Vedic mathematics computation using reversible logic gates. The Vedic multiplier is implemented in FFT, which uses Urdhva Tiryabhyam, Nikhilam Navatashcaramam Dashatah, and Anurupye Vedic mathematical algorithms. The N point DFT is computed by using efficient Fast Fourier transform (FFT) algorithm. It's necessary for a multiplier to be fast and power efficient in order to make this process rapid and simple. It is used to solve partial differential equations and also to perform convolution operations.

© 2020 Elsevier Ltd. All rights reserved.

Selection and/or Peer-review under responsibility of International Conference on Mechanical, Electronics and Computer Engineering.

Keywords: Type your keywords here, separated by semicolons ;

#### 1. Introduction

The core intention of this proposed modified Ripple carry adder is to propose a low power high speed VLSI "adder subsystem" to increase the CPU performance and the entire Design of the Ripple carry adder is implemented with Backend tools. The performance improvement of a CPU is directly dependent on the performance of the subsystems. The adder subsystem is the heart of a processor, with its manipulation, other than addition, operations like subtraction, multiplication and address fetching can be achieved. Hence, this proposed Modified Reversible Logic Gates logic based RCA, Adder made a novel technique which reduces the power, delay, and no of transistors when compared to the previous logic. In previous paper [1] work we have already implemented a 4 \* 4 Vedic multiplier using reversible logic. In that multiplier we have implemented by using ripple carry adder using reversible logic gates. In this paper we have modified our xor logic by using Modified Reversible Logic Gate design. By using proposed logic we have designed a ripple carry adder which executes high speed carry addition operation simulated by using the Backend tool called Cadence. The output of the Modified Reversible Logic Gates logic based Ripple carry adder is compared with existing adder like Carry select adder, carry shift logic, and carry look ahead adder. The proposed Modified Reversible Logic Gates logic XOR logic based Ripple carry adder has reduced the delay, power, size and no of transistor count when compared to the previous logic.

DFT can be applied to numerous tasks, including FFT, and likewise, also to image enhancement. In order to improve things, we must find the solution. Fast execution of the FFT and DFT

operations should be expected. The Vedic multipliers use less power and work compared to traditional ones such multipliers such as Wallace, but are faster to complete. Speed and performance also depend on the number of additions and multiplications that take place in the system at any given time. Because of the use of parallel adders, the long multiplication will increase. (To make a mixture). Partial Products Generation (PPG), PPG and Partial Products Addition (PPA) both expand in the first two steps and end users to increase the total product in the third Final Convectional Addition (FCA). When it comes to the main concerns, the biggest issue is that the multipliers move more slowly.

## 2. Proposed Method :-

Our proposed modified reversible logic gates based RCA Adder will overcome all the drawbacks faced in the existing Our proposed modified reversible logic gates based RCA Adder has finds a solution to address all of the issues & drawbacks faced in the existing Ripple Carry adder(RCA) systems. The output comparison can be done by using the simulator software called Xilinx-ISE-10.1 and The graph above shows that the proposed modifiable reversible logic gates based RCA Adder has reduced the delay, power, area, and not of transistor count compared to the existing adder techniques. Our previous work, which used modified reversible logic functions, is depicted in the diagram above. Adder shows how to calculate 4x high speed in a simple way. Internet access with high-speed A Modified RCA multiplier is created by swapping out two classic RCA multipliers for two improved versions. The 4x4 High Speed Multiplier's top-level diagram is depicted in this diagram. Each block has four times two bits of input and four bits of output, as illustrated in the block diagram

Our proposed modified reversible logic gates based RCA Adder will overcome all the drawbacks faced in the existing Our proposed modified reversible logic gates based RCA Adder has finds a solution to address all of the issues & drawbacks faced in the existing Ripple Carry adder(RCA) systems. The output comparison can be done by using the simulator software called Xilinx-ISE-10.1 and The graph above shows that the proposed modifiable reversible logic gates based RCA Adder has reduced the delay, power, area, and not of transistor count compared to the existing adder techniques. Our previous work, which used modified reversible logic functions, is depicted in the diagram above. Adder shows how to calculate 4x high speed in a simple way. Internet access with high-speed A Modified RCA multiplier is created by swapping out two classic RCA multipliers for two improved versions. The 4x4 High Speed Multiplier's top-level diagram is depicted in this diagram. Each block has four times two bits of input and four bits of output, as illustrated in the block diagram

#### 3. Vedic Mathematics :-

The sixteen sutra of Vedic maths are listed below. The thirteen upa sutras are also listed below. The main issue is to increase the speed of the multiplier partial products should be reduced. The product of the sum is the sum of the products of the Vedic mathematician's previous foundations

S.n.o	Name of The Sutras	Description
1.	Anurupyena	Proportionality
2.	Sisyate Sesasmjnah	The remainder remains constant
3.	Adyamadyenantyamantyena	The first by the first and the last by the last
4.	Kevalaih Saptakam Gunyat	For 7 the multiplicand is 143
5.	Vestanam	By osculation
6.	Yavadunam Tavadunam	Lesser by the deficiency
7.	Yavadunam Tavadunikrtya Vargnca Yojayet	Whatever the deficiency lesser by the amount and
		set up the square of the deficiency
8.	Antyayordasake'pi	Last totalling 10
9.	Antyayoreva	Only the last terms
10.	Samuccayagunitah	The sum of the product
11.	Lopanasthapanabhyam	By alternate elimination and retention
12.	Vilokanam	My mere observation
13.	Gunitasmuccayah Samuccayagunitah	The product of the sum is the sum of the products
15.	Vyashtisamanstih	Part and whole
16.	Yaavadunam	Whatever the extent of its deficiency

#### The thirteen upa- sutras of Vedic mathematics are listed below.

## 4. Reversible logic:

Reversible logic has its basics from thermodynamics of information processing. According to this, traditional irreversible circuits generate heat due to the loss of information during computation.. A circuit that is built from reversible gates is known as a reversible logic circuit. Some of the reversible gates are shown below that are used in the proposed design. The representation of a 3\*3 reversible gate is shown below. The Quantum cost of this gate is 4.4 joules. The circuit should contain at least the minimum number of reversible gates as possible to achieve reversibility. It is calculated knowing the number of primitive reversible gates (1\*1 or 2\*2) required to realize the circuit. The design parameters are below: Garbage outputs (GO), Gate Counts (GC), Constant inputs (CI) and Quantum cost (QC) The circuit is said to be reversible when it contains at least four reversible gates.

# 5.1 Toffile Gate

Toffile gate is a 3\*3 reversible gate. Peres gate is one of the popular gate and used in many applications. BME gate has four inputs and four outputs. The quantum cost of this gate is 6.2 \* 6. Toffile and BME gates are reversible. While designing the circuit using reversible logic gates the designer should concentrate on the following parameters

S.N.O	PARAMETERS OF REVERSIBLE LOGIC GATES	DESCRIPTION
1.	. Garbage Outputs(GO)	Indicates the unused outputs in the circuit which cannot be avoided and it is necessary to achieve reversibility. The circuit should contain minimum number of garbage outputs
2.	Constant inputs (CI)	This refers to the number of inputs that are to be maintained constant at either 0 or 1 in order to synthesize the given logical function
3.	Gate Counts (GC)	Indicates the number of reversible gates that are required to design the circuit, so that the circuit should contain reversible gates as minimum as possible
4.	Quantum cost (QC)	This refers to the cost of the circuit in terms of the cost of a primitive gate. It is calculated knowing the number of primitive reversible logic gates (1*1 or 2*2) required to realize the circuit

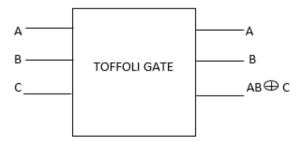
Some of the reversible logic gates are shown below that are used in the proposed design: 5.2 PERES GATE: It is a 3\*3 reversible gate i.e., it has three inputs and three outputs. The representation of Peres gate is shown below. Quantum cost of this gate is 4.



Fig.1: Peres gate.

Peres gate is one of the popular gate and used in many applications.

5.3 TOFFOLI GATE: It is a 3\*3 reversible gate i.e., it has three inputs and three outputs. The representation of Toffile gate is shown below. Quantum cost of this gate is 5.



# Fig.2: Toffoli gate.

5.4 BME GATE:

It is a 4\*4 reversible gate i.e., it has four inputs and four outputs. The representation of BME gate is shown below. The quantum cost of this gate is 6.

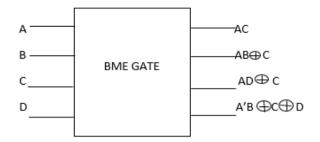


Fig.3: BME gate.

#### 5.5 Structure of Urdhva 2x2 Vedic Multiplier

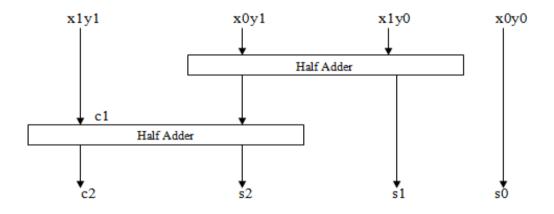


Figure 4.Structure of Urdhva 2x2 Vedic Multiplier

The Urdhva tiryakbhyam Sutra is applied to the binary number system to create a digital multiplier architecture. This is similar to the widely used array multiplier design. As shown in Fig 4, the method is described for two 2-bit numbers X and Y, where X = x1x0 and B = y1y0. Booth multiplication is another important multiplication algorithm. Large booth arrays are required for high-speed applications, and it allows for a quick multiplication of two numbers. Because it requires a large number of no gates, an array multiplier is more expensive.

The algorithm is depicted in the figure below. 101 times 110 is a multiple of 101.

1. First, take the right-hand digits from both the multiplicand and the multiplier. Then add them all together. Then we'll get the answer's LSB digit.

2. Multiply the top number's second bit by the bottom number's LSB. Then multiply the top number's LSB by the bottom number's second bit. Then add them up to get the second part of the answer.

3. Multiply the third bit of the bottom number by the LSB of the top number, the second bit of the bottom number by the second bit of the top number, and the third bit of the top number by the LSB of the bottom number. Then add them to get the third part of the answer.

4. Repeat the second step, but this time moves one space to the left. We'll multiply one number's second digit by the other number's MSB.

5. Finally, to get the final product, multiply the LSB of the top and bottom numbers.

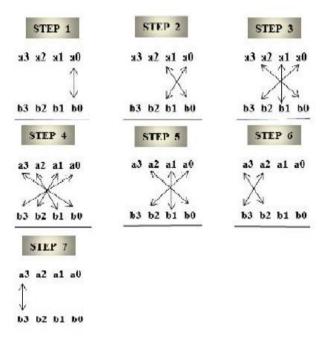


Fig.5: MULTIPLICATION OF TWO 4 BIT NUMBERS USING URDHVA TIRYAKBHYAM.

324	r 20	324 R 14
715	PC 00	715 PC 02
0 CARRY 2-	20	60 -16
324	R 45	3,24 R 17
715	PC 01	715 PC 04
660	.46	1660 21
CARRY 4		CANNY 2
	324	R Z1
	715	PC 02
	231660	23
	R in the Recall Personal	

Fig.6: URDHVA TIRYAKBHYAM ALGORITHM FOR DECIMAL MULTIPLICATION.

#### 5. Proposed System:-

The existing adder system designs are the conventional adder designs that usually consist of a long carry propagation chain for serial addition. Such adders so far exists are the Ripple Carry Adder, Carry Look-Ahead adder and few high speed adders using Pass-transistor logic and Transmission gate logic will increases the power, Delay ,size and no of transistor counts. Our proposed ripple carry adder using Modified reversible logic will overcome all the drawbacks faced in the existing system. The output comparison can be done by using the Simulator Software tool called Xilinx-ISE-simulator. and it shows that the proposed Modified reversible logic Ripple carry adder has reduced the delay,power,area,and not of transistor count compared to the existing adder techniques. The figure 7 shows our previous research work based on 4 \* 4 Vedic multiplier by using reversible logic gates. Now in future we are going to modify the Existing ripple carry adder with our proposed logic. Which in further reduces the size, area, and power.

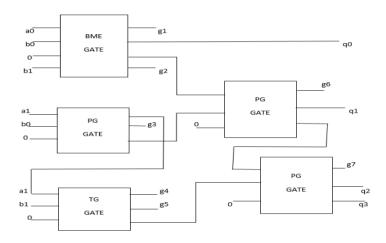


Fig 7a proposed 2\*2 Vedic multiplier using Reversible logic gates

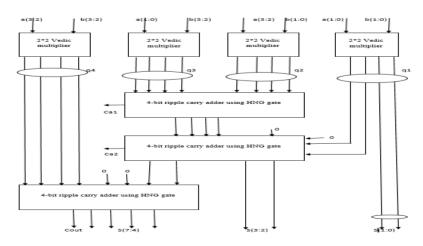


Fig. 7b . Block diagram of Reversible 4\*4 Vedic multiplier

Designing a low-power VLSI sub system has turn into a significant performance objective because of the high-speed budding technology in mobile computation and communication. Most of the works concentrated basically on reducing the size and number of transistor used in the design. We have designed Modified reversible loic gate and hence managed to bring down the transistor count which is nothing but the drop in area, power utilization and delay in the circuit. The importance given to the design of reversible logic gate is because of the fact that most of the systems constitute HNO gate circuits associated with other circuits. As in case of a HNO based Ripple carry adder, random number generator etc., reversible logic gates plays a major part. Hence the improvement in performance of an adder subsystem and so on. the schematic diagram of ripple carry adder is shown in fig6. For better observation of the output results, the gates can be connected to an adder and the performance of each can be compared with existing technique.

## 5.1. Drawbacks In The Existing System

Power is one of the peak factor resources of the designer tries to save when designing a VLSI based sub system. The goal is to increase the battery life time of portable electronic devices, by reducing the no of transistors, power consumption and to decrease the power spent per arithmetic operation, however lower power utilization does not essentially bring about lower energy dissipation and higher performance. To execute mathematics operations, a device can use very low power by functioning at very low frequency but it may use extremely a long time to finish the operation. For that reason we calculate the energy indulgence and evaluate the performance of the system by calculating the Power Delay Product (PDP), which is the product of normal power consumption and worst case delay. All these are the important factors of performance evaluation for any VLSI subsystem. When these are taken into consideration, then the conventional adders fails to achieve certain requirements. Here is always a exchange between power consumption and speed of the processing unit. Also there is a cost spent for the sizes of the system with newer logics bringing in more transistors. These were the major drawbacks of the existing system.

#### 6. Proposed High speed FFT Transform using modified reversible logic gates

The performance of high speed multiplier is to be improved which is designed based on different Vedic sutras can be implemented by using modified high reversible logic based carry select adder Urdhva tiryabhyam, Nikhilam Navatashcaramam Dashatah, Anurupye Vedic mathematical algorithm. By using this multiplier, computation time for NxN bits is to be reduced. Fast Fourier transforms (FFT) transforms of a DSP processor is to be implemented by these NxN multiplier and compare the following factors in the above three sutras: The calculation time for the multiplication of bits is to be minimized. The worst case propagation delay in the optimized Vedic multiplier case is 32.16ns. The Vedic multiplier is implemented in FFT, which uses Urdhva Tiryabhyam, Nikhilam Navatashcaramam Dashatah, and Anurupye Vedic mathematical algorithms. The N point DFT is computed by using efficient Fast Fourier transform (FFT) algorithm. It's essential for the multiplier to be high-speed and power efficient in order to make this process rapid and simple. It is used to solve partial differential equations and also to perform convolution operations

#### 7. Implementation of Vedic Sutras in Fast Fourier transforms using Modified reversible logic

Vedic algorithm is proposed for the implementation of multipliers that are used significantly in Fast Fourier Transform (FFT) and it provides fast and reliable approach to compute the N point DFT. Eq. 1. Shows the DFT function of X (k) that is an *N*-point sequence of x (n).

$$X(k) = \sum_{n=0}^{N=1} x(n) \exp^{\left\{-j\left(\frac{2\Pi}{N}\right)kn\right\}}, 0 \le k \le N-1$$
(1)

The simplified notation of DFT is mentioned in Eq. 2,

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, 0 \le k \le N-1$$
(2)

#### 8. Results & Comparison of various Vedic FFT Mathematic computation sutras

The Output window for 8x8 Anurupye Vedic Multiplier which was implemented in FFT is shown in Fig.8. It performs multiplication of two decimal numbers 58 x 48 by "If one is in ratio, the other is zero operation" (Anurupye method). Using the Output window for 8x8 Urdhva Vedic Multiplier which was implemented in FFT is shown in Fig.9 It performs multiplication of two decimal numbers 154 x 142 by "Vertical and crosswise multiplications" (Urdhva- Tiryakbhyam method). The Output window for 8x8 Nikhilam Vedic Multiplier which was implemented in FFT is shown in Fig.10. It performs multiplication of two decimal numbers 97 x 94 by "all from 9 and last from 10 operations" (Nikhilam method). The performance analysis of Vedic multiplier using modified reversible logic with high speed and low delay performance characteristics are represented in the Table

#### 9. RESULTS AND DISCUSSIONS

The Output window for 8x8 Anurupye Vedic Multiplier which was implemented in FFT is shown in Fig.8.

Image:		● • M 📰 🔤 • × • 🐵 🗍 J	BB221	
Image: Second			100 ps 🛊 🔝 🕅	· · · · · · · · · · · · · · · · · · ·
Muse         Muse           01100110101010101010101010101010101010	Coleman Co	te 👻 Colanniayout 🛛	Layout Simulat	1911221
Numeric Conflict         Huge           2         APL Second 201           3         APL Second 201           4         APL Second 201           4         APL Second 201           5         APL Second 201           6         APL Second 201	5	ndin later test test	R 94	1 0 10 II ALL
Migging         Migging           # //fl:_design/st         0			.r	
Mage         Mage           # //fl_design/still         0           # //fl_des	-1010-			and Make - Default
Attl. design/dit         1           Mitt. design/dit         1         1           Mitt. design/dit         1         1         1         1         1           Mitt. design/dit         1         1         1         1         1         1         1           Mitt. design/dit         1         1         1         1         1         1         1           Mitt. design/dit         1         1         1         1         1         1         1         1         1           Mitt. design/dit         1         1         1 <th1< th=""> <th< th=""><th></th><th></th><th>Maria</th><th></th></th<></th1<>			Maria	
Imm_description         0           Imm_description         0			1	
Diss         Interpret/set         Distribution         Distribution           Diss         Interpret/set         Distribution         Distribution         Distribution           Diss         Interpret/set         Diss         Distribution <thdis< th=""> <thdis< th=""></thdis<></thdis<>			0	
D = melempt/st         01100110101.         01100110101.0101101001001           D = melempt/st         01100110101.0101101001001001         01100110101.00101001001           D = melempt/st         01100110101.0010101001001001         0110010100100100000000000000000000000	1001001101100110111	1011001101101001001001001100110	1011001301103	
D = //m_sheap/v2         01100110101010101010101010101010101           D = //m_sheap/v2         0110011010100100101000000010000000000				
D         ml_sheep/v1         101.001.0000.0001.01.01.01.001.0001.00				I //fft_design/k2
D         mt         piespy/st         bit 100 its 101.         bit 100 its 101 is 101 is 0000001           D         mt         piespy/st         bit 200 its 101.         bit 100 its 101 is 0000001           D         mt         piespy/st         bit 200 its 100	100110110110111	0111001101101100101101100110111		
D = //T_strengt/st         0000001 101 01 00 01 01 01 01 00 100 10	0001801101100110111	10110011000100001101101100110111		
D         Im. Jesupt/V         101:00110111.00110110110010010101101           D         Im. Jesupt/V         001:0011010010001000100000000000000000				
D         3/11.045499/v0         1011011000001.         101101100000010011001.           D         3/11.045499/v1         001001011000.         001001111000.           D         3/11.045499/v1         001000010111.000000100110010000100000001000000				
D         ML_desupt/v1         000010100111000.0001100110010001100100001100100				
D         mt_despr/v2         01000000111         010000001011         010000001100000011           D         mt_despr/v1         0101011000001         00100110010001         0010011000000000000000000000000000000				
D ≤ //L_desupt/v1         1131510000011         1101100100001         110110010001           D ≤ //L_desupt/v1         1001101100010         110110010001         10011010001           D ≤ //L_desupt/v1         1000110100010         100110100010         100110100010           D ≤ //L_desupt/v1         10000110100010         1000011000010         1000011000010           D ≤ //L_desupt/v1         10000110100010         10000110000010         100001100000000				
D=2         Mdesgr/y/         10010011100100100000000000000000000				
C → (Th. design/v5 1000001 00000 10 1000 1 00000 10 10000 10000 1000000				Contraction of the second seco
Control (19) (19) (19) (19) (19) (19) (19) (19)				
D-2 /M_design/y7 0110100001111 0110100001110000 0101111101000				
The design/s2 130000 to 130 11 100000 P011011000101100110011001100				
			Protection of the local data	Statement of the local division of the local
New 1400 ps 000 ps 500 ps 1200 ps		600 (14		and the second se
Career 1 D pm				Cursor 1

Figure 8.Simulation Output window for 8x8 Anurupye Vedic Multiplier using Modified reversible logic which implemented in FFT

The Output window for 8x8 Urdhva Vedic Multiplier which was implemented in FFT is shown in Fig.9. It performs multiplication of two decimal numbers  $154 \times 142$  by "Vertical and crosswise multiplications" (Urdhva-Tiryakbhyam method).

	K BB 221	0.A #	• × • 🕸 📘	er er, ve Sje	34 北 25	· [[]] \$				
9 1 + + II	100 ps 🛊 🚺 🕅	日気の一副								
1911281	Layout Simula	te 💌	Columniayout [p	efault			-1 -8-1	8.94		
1 0 to 11 m	1	11 ID 1 1 1	チモチェ	5.5 3		Search: [		-	(8 /8 / B /	
	ГГ									
Wave - Default										
	Mags	0			-					
	Maga				_					
/fft_design/dk:										
//ft_design/rst ()	0	10110011011011001								
B-4 //ft_design/x1	1011001101101	10110011011011011								
D-1/fft_design/x2	1011111101101	10111111011011001								
Int_design/k3	0111001301101	01110011011011001								
D-4 //TL design/v4	3011001300010	10110011000100001								
D 3 /fft_deegn/x5	1011001301101	10110011011011001								
D 4 /m_design/k6	0000001101101	000000011011011001								
m-1 /m_stesign/x7	1011001101111	10110011011111001								
D 🔥 /fft_design/y0	1011011100010	101101110001010111								
D / /TTL design/y1	0010100111100	00101001111000001								
B to /fft_design/y2	0 1000000 10 111	010000001011100000								
	1110110010001	11101100100010101			_					
- /fft_design/y4	1001101110010	10011011100100110	0000000110110							
- /fft_design/y5	100000 10 1000 1	100000 10 1000 11 1000	01010111101100							
Int_design/v6	0000011010010	00000110100100001	11110011111001							
D-de /fft_design/y7	0110100001111	01101000011110000	01011111101000		_					
III.design/s0	0110011001111	011001100111110010	11011001101110							
- ///t_destgn/s1	0110011011011	011001101101100110	11011000111000							
III /III	1100001011011	110000101101100010	1101100110110							
and the second se			THE REPORT OF THE	THE OWNER OF TAXABLE PARTY.	CONTRACTOR OF	THE REAL PROPERTY OF	STATES OF STATES		COLUMN TWO IS	
Now				800 ps						1400 p
Cursor 1	D pm									
x	F 6	•								

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

	<b>R 6 22</b>	0-48-	5-2-31	J== = := 33-	3* 36 46 1	mi 🗢 🖽 🕯	72 (195		
96 * ** **   3.7	100 pe 🛊 陆 🗈	113615	1 (Ta) (*)						
1011221	Leyout Simul	ace 🗣	Column Ayout	Default		-	18.18.10	41-43	
1 0 10 10 mil	1 1 1 1	adia a	L'EST. 3	553	- +6 - 3+ 1	Search:		1,0000 00 11	
				11-1				11	
Wave Defait	IS-NO.								
and the second at	1	-		1010					
and the second s									
/fft_design/rst	0								
D- /fft_deskpr/x0	3011001301105-	10110011011011	001901101100110131						
CHAR /Ht_design/s1		10110011011011	0111011001100110111						
m-do /fft_deaps/s2	#F1111101801	10111111011011	001001001001001001						
Co-do (mt_desept/c2		01110011011011	00100100100110111						
CI-2 /TTL_terAps/v4		10110011000100	00101101100110011011						
El-20 /fft_design/v1	8011001101101	10110011011011	100100110110100000001						
Int_deskgn/xtil	0000001101101	00000011011011	001101101100110111						
Chille /H1_stenipr/x7		ROTIOGICO TOTAL	00100100100110010						
CD & /TTL, deskpr/y0	8711011100018		011001000110000010						
El- Anti-design/v1	00101007111000		001 900 11 100 10 10 110						
Children (TTL_densign/y/2	0 2000000 20111		00000001100000111						
E- fft_design/y7			101 20000 111 10 10 100						
D-la (fft_design/y4			11000000000110110						
CD- (TTL_design/v/l-			10000 10 10 11 10 1100						
E ////.destpv/yti-	00000110010010		00111110011111001						
El Carlotta design/y7	01101000011111		000 80 10 11 11 110 1000						
Int_stestph/s0	0110011001111		001011011001101100						
Int_design/s1			0110110011000111000						
Int_design/h2		110000010110110	001011011001101100						
No.	1400 pe	ALL DATE AND A DESCRIPTION OF THE OWNER OWNER OF THE OWNER OWNER OF THE OWNER OWNER OF THE OWNER OWNE OWNER OWNE		ALL PLATER BARN	AREA COLORADOR	THE R. P. LEWIS CO., LANSING MICH.		ALL DELLAS DE LA DELLA D	A DURAL DU
Curser 1				300 ptr		10003 gee			1400 (4

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

Figure 11. delay calculatation for the proposed 8 \* 8 FFT by using **Anurupye Vedic Multiplier** using **Modified reversible logic** 

N				Virtuoso (R) Visualizatio
Eite Tools Vie	w Options	Constants Help	p	
In Contest R	esuits DE: /re	ot/simulation/C	TL_test/spectre/schem	atic/paf
off C Family	💌 Wave 🖌 👱	cup   BA	Append	
	Contraction of the second second	CHEMOLOGICAL SECOND SECOND	and herein a second sec	
1671E-12				
and the second se	COLUMN STREET,			
Po Po	O BO HO	How Most	M+ ME E+ EE	1 9 C
A LUM	a ling. Item			
And Line	5 1 0 D. III 10			
ALCONG.	s 1 mgr Il te			
ALCONG.	A LOD. No.	•		
All				- 100 - 22 - 20 - 20 - 20 - 20 - 20 - 20
All	delay	gpc_gain	peak.ToPeak.	sin
All	delay der/v		peakToPeak period itter	sinh
All	delay	gpc_gain		
All 10 10 72 80	delay der/v	gpc_gain groupDetay gt	period_jitter phase	sinh slev/Rate
All 101-5 Rn a2d	delay deriv dit ditob	gpc_gain groupDetay gt gumx	period_jitter phase phaseDeg	sinh slev/Rate spectralPower
All 10° x Rn a2d abs	delay deriv dft dftbb dfd	gpc_gain groupDetay gt gums harmonic	period_jitter phase phaseDeg phaseDegUnwrapped	sinh slevRate spectralPower spectrum
All 10° >> Fin a2d abs accs	delay deriv df dfbb dnl dutyCycle	gpc_gain groupDetay gt gumx harmonic harmonicFreg	period_jitter phase phaseDeg phaseDegUnwrapped phaseMargin	sinh slewRate spectralPower spectrum opectrumMeas
All 10°% Fin a2d abs acoth acoth	delay der:v df dfbb dnt dutyCycle evin GAM	gpc_gain group Delay gt gums harmonic harmonic Freq histo	period_jitter phase phaseDeg phaseDegUnwrapped phaseNatgIn phaseNoise	sinh slevRate spectralPower spectrum opectrumMeas sqrt
All 10° % Pin a2d abs acost acost acost acost	delay deriv dft dfbb dnl dutyCycle evmGAM evmGAM evmGpik	gpc_gain group Delay gt harmonic harmonicFreq histo integ	period_jitter phase phaseDeg phaseDegUnwrapped phaseNargin phaseNarse phaseRad	sinh slewfate spectralPower spectrumMeas sert sisb
All 1/2 Rn a2d abs acost acost asim osimb	delay deriv dft dftbb dnl evtr GAM evtr GAM evtr GAM evtr Gpik evp	gpc_gain group Delay gt harmonic harmonicFreq histo iinteg iimag	period_jitter phaseDeg phaseDegUnwrapped phaseMargin phaseNoise phaseRad phaseRad	sinh slev/Rate spectralPower spectrum spectrum sat sat sat
All 10° % Rn a2d abs acosh acosh asim asim asim	delay deriv dft dftbb dft datyCycle ovmCAM ovmCAM ovmCAM exp ovmCAM	gpc_gain group Delay gt harmonic harmonic Freq histo integ integ int	period_jitter phaseDeg phaseDegUnwrapped phaseNargin phaseNargin phaseRad phaseRadUnwrapped pow	simb slowAsto spectralPower spectrum spectrum spectrum sbato stato tan
All 1/2 Rn a2d abs acost acost asim osimb	delay deriv dft dftbb dnl evtr GAM evtr GAM evtr GAM evtr Gpik evp	gpc_gain group Delay gt harmonic harmonicFreq histo iinteg iimag	period_jitter phaseDeg phaseDegUnwrapped phaseMargin phaseNoise phaseRad phaseRad	sinh slev/Rate spectralPower spectrum spectrum sat sat sat
All 10° % En a2d abs acosh acosh asinh asinh alan alan	delay deriv dft dftbb dft datyCycle ovmCAM ovmCAM ovmCAM exp ovmCAM	gpc_gain group Delay gt harmonic harmonic Fre q histo iinteg int integ	period_jitter phaseDeg phaseDeg phaseDegUnwrapped phaseMargin phaseRad phaseRadUnwrapped pow psd	sinh slowRate spectrum spectrum spectrum soft soft sub tan tan tan tan tan
All 10° % Fin a2d abs acost acost asin asin asin asin asin asin asin	delay deriv dft dftbb ddt ddyCycte overGpsk overGpsk overGpsk fair fair fair fair fair fair fair fair	gpc_gain group Delay gt gumx harmonicFreq histo integ integ intersect	period_jitter phaseDeg phaseDeg phaseDegUnwrapped phaseNargih phaseRad phas	sinh stev/fate spectrum opectrum opectrum spectrum stat sat sat tab tan tan tan tan tan tan
All 1/x En abo abo abo asin asin asin average bit	detay deriv df dfbb dfbb dutyCycte ovm GAM ovm GAM ovm GAM ovm Gam eyeDiagram fail Time fail fine fourEval	gpc_gain groupDelay gt harmonic harmonic freq histo integ integ integ integ intersect ipn	period_jitter phase phaseDegUnwrapped phaseDegUnwrapped phaseMargin phaseRad phaseRad part of the the the the the phaseRad pow part of the the the the the part of the the the the the part of the the the the the the part of the the the the the the the part of the the the the the the the the part of the the the the the the the the the part of the	sinh slev/fat spectrum spectrum spectrum set set set slet slet tan tan tan tan tan tan tan tan tan ta
All 10° % Pin a2d abs acosh asimh atam awerage b11 average b11 ban dvidth	delay deriv dft dftb dftb dftb dftb dftb dftb dftb	gpc_gain groupDelay gt harmonicFreq histo linto int intersect ipn ipn k	period_jitter phaseDeg phaseDegUnwrapped phaseDegUnwrapped phaseRadunwrapped phaseRadunwrapped psd psd psd psd psd psdb psbde pstliter	sinh skowfatk spectral worr spectral more sent sent tan tan tan tan tan tan tan tan tan t
All 1% 10"% En a2d absin acts	delay deriv dfb dibb didy Cycle didy Cycle didy Cycle didy Cycle owe Disk owe owe Disk owe owe Disk owe fing fing fing fing fing fing fing fing	gpc_gain groupDelay gt gums harmonic harmonic harmonic harmonic integ int gint gint gint gint gint gint gint	period_lifter phaseDeg phaseDeg Unwrapped phaseDegUnwrapped phaseNargin phaseNargin phaseRadUnwrapped pow phaseRadUnwrapped poid psdbb psdbb psbbcde pather real	Sinh Skovitsis Skovitsis Spectrum Mess Sent Sent Skotev Tan and Tan and Tan Tan Tan Tan Tan Tan Tan Tan Tan Tan
All 10°% Rn a2d abs acosh acosh asinh atannn atann atann atann atann atannn atann atann ata	delay deriv dr. dr. dr. dr. dr. dr. dr. dr. dr. dr.	gpc_gain groupDelay gt harmonicFreq histo into inte inte inte interect ipn Xf kf	period_jitter phaseDeg phaseDeg Unwrapped phaseDeg Unwrapped phaseDagUnwrapped phaseDadUnwrapped pow phaseDadUnwrapped pow psdibb pzbode pzther real real	Sinh Sectal Sectors Sectors The Ware Sectors S
All 10° X Rn a2d abs acosh acos	delay deriv dito ditob ditob ditob ditob even Opak even Opak even Opak even Opak even Opak even Opak even Opak even Opak fig freq_lifter freq_lifter freq_lifter freq_uency ga	gpc_gain groupDelay gt gums harmonicFreg histo integ int gint grov int grov int grov grov Ri to grov Ri to grov Ri to grov histo int grov ki to grov histo int grov histo int grov histo int grov histo int grov histo int grov histo int grov histo int grov histo int grov histo int grov histo int grov histo int grov histo int grov histo int grov histo int grov histo int grov histo int grov histo int grov histo histo grov histo grov histo grov histo grov histo grov histo grov histo grov histo	period_litter phaseDeg_phaseDeg_phaseDeg_unvrapped phaseDeg_unvrapped phaseAsd_ phaseAsd_ phaseAsd_ phaseAsd_ phaseAsd_ protection p	sinh skovfists skovfiste spectrum spectrum set set set stangen tangen tangen unty dainpreq value value value value
All 10°% Rn a2d abs acosh acosh asinh atannn atann atann atann atann atannn atann atann ata	delay deriv dito ditob ditob ditob ditob even Opak even Opak even Opak even Opak even Opak even Opak even Opak even Opak fig freq_lifter freq_lifter freq_lifter freq_uency ga	gpc_gain groupDelay gt harmonicFreq histo into inte inte inte interect ipn Xf kf	period_jitter phaseDeg phaseDeg Unwrapped phaseDeg Unwrapped phaseDagUnwrapped phaseDadUnwrapped pow phaseDadUnwrapped pow psdibb pzbode pzther real real	Sinh Sectal Sectors Sectors The Ware Sectors S

Fig12. Delay calculation for the Existing Simulation Output window for 8x8 Urdhva Vedic Multiplier

N				Virtuoso (R)
Eile Tools Vie	w Options	Constants Help	,	
In Context B	esults DB /rc	ol/simulation/C	ontroixor_test/spectre/s	chematic/psf
-	5 A.M. 1		- In	
			Ovn Osp Ovsw	
OII OII O	ide  Q is	opt omp	🔾 vn2 🔾 zp 🔾 yp	🔾 gd 🔾 data
	a mare the		The second	
Off O Family	• wave I 🗹	Cub 2 4	Append	
49.0E-12				
43.0E*16				
1 -	1 Diana and	I Hann Manne I	many states I am and	
- 🔁 🔁 Po	ap	E EDMC ([SMC	M+ ME E+ EE	9 6
OAL WATER ALL DOGS	Contraction (Martin	A BAR AND		
4.1994				
All				
(All	dBm		nfmin	s21
(AII			nfmin overshoot	s21 s22
(All 1/x	dBm	gp gpc_freq		s21 s22 sample
All 1/x 10"x	dBm delay	gp gpc_freq gpc_cain	overshoot	s22 sample
All 1/x 10"x Rn a2d	dBm delay deriv df	gpc_freq gpc_gain groupDelay	overshoot peak peakToPeak	s22 sample settingTime
All 10**× Rn a2d abs	dBm delay deriv dri dribb	gp gpc_freq gpc_gain groupDelay gt	overshoot peak peakToPeak period_jitter	s22 sample settingTime sin
All 10"× Rn a2d abs acos	dBm delay deriv dri dribb dribb	gp gpc_freq gpc_gain groupDetay gt gumx	overshoot peak peakToPeak period_itter phase	s22 sample settingTime sin sinh
All 1/x 10"x Rn a2d abs acosh acosh	dBm delay deriv dfi dfi dni dutyCycle	gp gpc_freq gpc_gain groupDetay gt gumx namonic	overshoot peak peakToPeak period_itter phase phaseDeg	s22 sample settingTime sin sinh slewRate
All 1/x 10"x Rn a2d abs acos acosh asin	dBm delay deriv dfi dfi dfi dfi b dfi dfi b dfi b dfi dfi b dfi dfi b dfi dfi b dfi dfi b dfi dfi b dfi dfi b dfi dfi b dfi dfi b dfi dfi dfi dfi dfi dfi dfi dfi dfi dfi	gp gpc_freq gpc_gain groupDelay gl gumx harmonic harmonicFreq	overshoot peak peakToPeak period_jitter phase phaseDeg phaseDegUnwrapped	s22 sample settingTime sin sinh slewRate spectralPower
All 1/x 10"× Rn a2d abs acosh acosh asinh asinh	dBm delay deliv dfi dfibb dni dutyCycle evmQAM evmQAM	gp gpc_freq gpc_gin groupDetay gt gumx harmonicFreq histo	overshoot peak peakToPeak period_itter phase phaseDeg phaseDegUnwrapped phaseMargin	s22 sample settingTime sinn slewRate spectralPower spectrum
All 10"× Rn a2d abs acosh acosh asinh asinh aian	dBm delay delay dfb dfb dfb dfb dfb dfb dfb dfb dfb dfb	gp gpc_freq gpc_gain groupDetay gl gumx namonic hamonicFreq histo iinteg	overshoot peak peakToPeak period_jitter phaseDeg phaseDeg phaseDegUnwrapped phaseMargin phaseMoise	s22 sample settingTime sinh sinh stev.Rate spectralPower spectrumMeas
All 1/x R0"x R0 abs acosh acosh asinh asinh alan alanh	dBm delay deriv dfi dfi dibb dni dutyCycle evmQAM evmQpsk exp eveplagram	gp gpc_freq gpc_gain groupDelay gt gumx harmonic harmonicFreq histo iinteg iimag	overshoot peak.ToPeak period_litter phase phaseDeg phaseDegUnwrapped phaseMargin phaseNoise phaseNargin	s22 sample settingTime sinh slewFlate spectralPower spectrum spectrumMeas sert
All 10% En a2d abs acost acost asinh asinh atan atan atan	dBm delay deriv dfi dfibb dfil dutyCycle evmGAM evmGAM evmGpsk exp eyeDlagram failTime	gp_freq gpc_gain groupDetay gl gumx hamonic hamonicFreq histo iinteg imag int	evershoot peak peakToPeak period_litter phase phaseDegUnwrapped phaseMargin phaseMargin phaseRadUnwrapped phaseRadUnwrapped	s22 sample settingTime sinh slewRate spectralPower spectrum spectrumMeas sqt
All 1/X Rn a2d acosh acosh acosh asinh alan alanh alanh average bir	dBm detay detay driv dribb driv driv driv driv cycle evmGAM evmGAM evmGptk evyeDiagram faiTime faiTime	gp gpc_freq gpc_gain groupDelay gumx harmonic harmonic harmonic harmonic integ integ int integ	overshoot peak peaktoPeak penod_lifter phase phaseDeg phaseMose phaseMose phaseMose phaseRadUnwtaped phaseRadUnwtaped pow	s22 sample settingTime sinh sinh slewRate spectrum spectrumMeas sqrt ssb stdde v
All 10% En a2d abs acost acost acost acost acost acost acost acost acost acost acost acost acost acost bath atan	dBm delay derity drito drito drito drito drito drito evmQptk exp eyeOlagram failTime flip fourEval	BP gpc_freq gpc_gain groupDetay gl mamonic namonicFreq nisto iinteg imter integ inter ect	overshoot peak peaktoPeak period_litter phaseDeg phaseDegUnwrapped phaseMargin phaseMargin phaseRad phaseRad phaseRad pow	s22 sample settingTime sinh slewRate spectralPower spectrum spectrumMeas sqt ssb stddev tan
All 1% 10"% Fin abost actors accorb assin	dBm delay delay deriv dfi dfibb dfibb dityCycle evmCpsk evmCpsk evpClagram failTime filp fourEyai fourEyai	gp gpc_freq gpc_gain groupDelay gumx namonic namonicFreq histo imag int integ integ integ integ integ integ	overshoot peak peaktoPeak penod_lifter phase phaseDeg phaseDeg phaseMorise phaseNorise phaseRadUnwrapped phaseRadUnwrapped pow pidb	s22 sample settingTime sinh sinh slewRate spectralPower spectrumMeas sqct ssb sqt ssb stddev tangent
All 16× 10°× 8n a2d abs acost acost asinh atanh atanh atanh atanh atanh bandwidth clip compare	dBm delay derity drito drito drito drito drito drito evmQptk exp eyeOlagram failTime flip fourEval	BP gpc_freq gpc_gain groupDetay gl mamonic namonicFreq nisto iinteg imter integ inter ect	overshoot peak peaktoPeak period_litter phaseDeg phaseDegUnwrapped phaseMargin phaseMargin phaseRad phaseRad phaseRad pow	s22 sample settingTime sinh slewRate spectralPower spectrum spectrumMeas sqt ssb stddev tan
All 16× 10°× 8n a2d abs acost acost asinh atanh atanh atanh atanh atanh bandwidth clip compare	dBm delay delay deriv dfi dfibb dfibb dityCycle evmCpsk evmCpsk evpClagram failTime filp fourEyai fourEyai	gp gpc_freq gpc_gain groupDelay gumx namonic namonicFreq histo imag int integ integ integ integ integ integ	overshoot peak peaktoPeak penod_lifter phase phaseDeg phaseDeg phaseMorise phaseNorise phaseRadUnwrapped phaseRadUnwrapped pow pidb	s22 sample settingTime sinh sinh slewRate spectralPower spectrumMeas sqct ssb sqt ssb stddev tangent
All 1% 10"% Fin abost actors accorb assin	dBm delay delay deliv dfi dtip dtip cycle evmGAM evmGAM evmGPsk evp evpChagram failTime failTime fourEval freq_iter	P ppc_freq ppc_gain proupDelay gl pumx namonicFreq nisto mag integ integ intersect pnVRI	overshoot peak peak peak phase phasebeg	s22 sample settingTime sinh sinh sinh spectralPower spectrum spectrumMeas spectrum spectrumMeas store store store tan tangent tanh

Table 4. Comparison of Various Adders with our Proposed reversible logic gate based RCA.

Type of Adder	Power	Delay	PDP	Transistorcount
	(mW)	(ns)	(pJ)	
Existing Ripple Carry Adder (E-RCA)	0.22	4.49	0.89	902
Proposed Ripple Carry Adder (P-RCA)	0.13	1.87	0.28	856
Carry Look ahead Adder (CLA)	5.1	2.37	1.21	2208
Carry Shift Adder(CSL)	6	3.06	1.81	2176

## Table 5. Performance characteristics.

Modified reversible logic based FFT	Time constraint(delay)
8x8 multiplier using Nikhilam sutra	26.196ns
8x8 multiplier using Urdhva sutra	18.844ns
8x8 multiplier using Anurupye sutra	14 ns

The above is the screen shots of the Delay calculation for the 8 bit reversible logic gate based RCA. The above calculation is done after analysis after Modified reversible logic gate based RCA output. The screenshots show the Output response for the 8 bit reversible logic based RCA adder.

## **10.** Conclusions

The concept of Modifed reversible logic gate is introduced in VLSI design. The Modified reversible logic based Ripple carry adder" which trades certain amount of accuracy for significant power saving and high speed operation. The Proposed reversible logic based RCA adder is simulated using Xilinx –ISE-Simulator tool. The proposed Adder which Extensive comparisons with conventional digital adders showed that the proposed reversible logic based RCA output performance shows that it reduces the overall power consumption, power delay product ,no of transistor count and increases the speed which in turn performs high speed arithmetic operations. By using this vedic sutras technique we have implemented a 8\*8 vedic multiplier using ANURUPYE sutras and by using this multiplier we have implemented in Fast fourier transform and it has performs a high speed computation process when compared to other sutras using reversible logic gates.

## References

- [1] Rakshith, T. R., & Saligram, R. (2013, March). Design of high speed low power multiplier using reversible logic: A vedic mathematical approach. In 2013 International Conference on Circuits, Power and Computing Technologies (ICCPCT) (pp. 775-781). IEEE.
- [2] Charishma, G. G. K. V. (2012). Design of high speed vedic multiplier using vedic mathematics techniques. Itn'l J. of Scientific and Research Publications, 2(3).
- [3] Islam, M. S., Rahman, M. M., Begum, Z., & Hafiz, M. Z. (2009). Low cost quantum realization of reversible multiplier circuit. Information technology journal, 8(2), 208-213.
- [4] Dhillon, H. S., & Mitra, A. (2008). A Digital Multiplier Architecture using UrdhvaTiryakbhyam Sutra of Vedic Mathematics. Department of Electronics and Communication Engineering, Indian Institute of Technology, Guwahati, 781, 039.
- [5] Maslov, D., Dueck, G. W., & Miller, D. M. (2005). Synthesis of Fredkin-Toffoli reversible networks. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 13(6), 765-769.
- [6] Chidgupkar, P. D., & Karad, M. T. (2004). The implementation of vedic algorithms in digital signal processing. Global J. of Engng. Educ, 8(2), 153-158.
- [7] Rangari, M., Saraswat, R., & Jain, R. (2015). Design of reversible logic alu using reversible logic gates with low delay profile. International Journal of Advanced Research in Computer and Communication Engineering, 4, 344-348.
- [8] David, A. (2020). Corporate and individual environmental responsibility towards automobile. Book Rivers.