A Novel Vedic Divider Architecture with Reduced Delay for VLSI Applications

Siba Kumar Panda
Assistant Professor, Dept. of ECE
CUTM, SOET
Bhubaneswar, Odisha, India

Arati Sahu
M.Tech Scholar, Dept. of ECE
CUTM, SOET
Bhubaneswar, Odisha, India

ABSTRACT

The ever increasing demand in VLSI architecture to handle complex systems has resulted for designing of high speed divider architecture. The divider is designed using ever known ancient methodology “Vedic mathematics”. There are several methods present in Vedic mathematics but here Parvartya sutra is used. It is a general division formula which can be applicable to all cases of division which is an efficient way for dividing large numbers with respect to delay and power consumption. Here thirty-two bit divider architecture is implemented using this sutra & synthesized and simulated using Xilinx ISE simulator and implemented on virtex4 FPGA device XC4VLX15.The output parameters such as propagation delay and device utilization are calculated from synthesis results. Our result shows speed improvement as compared to other architecture presented in this literature. This architecture can be implemented in many applications such as digital signal processing, cryptography, processor arithmetic unit design etc.

General Terms

Algorithms, Sutras

Keywords

Vedic Mathematics, Vedic Divider, Paravartya Sutra, VLSI design, Digital Signal Processing

1. INTRODUCTION

Now days digitization plays a vital role in the designing of processor, ALU unit [1] has greatest importance. In design of processor arithmetic units are always based on addition, substraction, multiplication [2] division operations. Out of all these operations there are a lot of scientific papers are present based on hardware implementation of addition & multiplication operation. But division also has the equal importance in designing arithmetic unit & has many applications in real world systems. But there are few scientific papers available on hardware implementation of division algorithms. As division is a sequential type of operation its more costlier in terms of complexity & complex compared with other mathematical operations like multiplication & addition. Division is mainly of two kinds a) slow division b) fast division. Slow division includes several methods like restoring, non restoring & SRT method. Where one digit of final quotient is produced per iteration. Fast division includes methods like Newton rapshon method & Goldschmidt method. It gives a close approximated result i.e. twice as many digit of the final quotient on each iteration.

Vedic mathematics [3] (Sri Bharati Krishna Tirthatji Maharaja 1992) is full of magic’s & mysteries. In ancient days Indians were able to understand these mysteries & able to develop simple keys to solve those mysteries .Thousand years ago Indians used these techniques in different fields like medicine, science etc. due to which INDIA emerged as the richest country in the world. They named these systems of calculation as “VEDIC MATHEMATICS”. Vedic mathematics is much simpler, easily understandable & tricky too.

Swami Bharati Krishna Tirthaji Maharaj sankaracharya of govardhan peath was introduced Vedic mathematics to this world in the form of 16 sutras (formulae) which are very significant for calculations. Bharati Krishna, who was himself a scholar of Sanskrit, Mathematics, History and Philosophy, was able to reconstruct the mathematics of the Vedas. He had explored the mathematical potentials from Vedas and proved that the mathematical operations can be carried out to produce fast answers using the Sutras .As most of the engineering system designs are based on mathematical approach so conventional mathematics become an integral part of education. The leading microprocessor manufacturers developed their architectures to be suitable for conventional binary arithmetic methods. The need for faster processing speed drives us for major improvement in processor technologies & also search for new methodologies. The Vedic mathematics approach is totally different &can be easily adaptable.

Vedic Mathematics introduces the wonderful applications to Arithmetical computations, theory of numbers, compound multiplications, algebraic operations, factorizations, simple quadratic and higher order equations, simultaneous quadratic equations, partial fractions, calculus, squaring, cubing, square root, cube root, coordinate geometry and wonderful Vedic Numerical code.

Ancient Vedic mathematics mainly deals with sixteen sutras. Paravartya sutra is one of them which is used for division. A shortcut method for dividing a polynomial [4] by a linear divisor is one of the forms of general division is considered as a special case of the paravartya sutra in Vedic mathematics.

The word “Vedic” is derived from the word “Veda” which means the store-house of all knowledge. Vedic mathematics is mainly based on sixteen sutras & from that paravartya sutra is for division. Sanskrit term PARAYARTYA means all Transpose and apply.

A divider is one of the important hardware block which is applicable in most of applications such as digital signal application, encryption, and decryption algorithms in cryptography [5] & also in various mathematical computations. With advances in technology, many researchers have tried to design dividers which offer either of the following- high speed, low power consumption, regularity of layout and hence less area or even combination of them in dividers. From which the Vedic divider is considered best here to satisfy our requirements.

In this paper, we present division operations based on paravartya sutra, and implemented in HDL language. As compared to other methods like restoring division method this divider provides reduced delay. The paper is organized as follows. In the starting section the background its related work are presented. In the second section Vedic division principle is described which is based on paravartya sutra. In the third section the proposed divider is described. Fourth section deal with the design of the architecture of the divider in BCD format. Fifth section summarizes the experimental result obtained & the final section presents the conclusion and future work of the literature.

2. BACKGRUND AND RELATED WRK

Dividers are important components in processor design. The normal division i.e. restoring division (R.Bhaskar.Ganapathi Hegde, P.R.Vaya 2011) is described for sixteen bit division & well explained its implementation in RSA [6] encryption system. Extensive research work has been published which implemented Vedic division method / algorithm for (Ratiranjan Senapati Bandan Bhoi & Manoranjan Pradhan 2012) for 8 bit division. But it is implemented for binary numbers & compared with our Vedic division & shows an improved delay result. In this literature we are using the Vedic mathematics sutra i.e. paravartya sutra for dividing a 32 bit number in BCD format.

In progress Floating-point divisor architectures[7] have low frequency, larger area and high latency in nature. With advent of more graphic, scientific and medical applications, floating point dividers have become crucial and increasingly important. However, most of these modern applications need higher frequency or low latency of operations with minimal area occupancy. In this work, highly optimized pipelined architecture of an IEEE-754 standard double precision floating point divider is designed in order to achieve high frequency on FPGAs. By using secondary clock to perform mantissa division the overall latency of the divisor is reduced to 30 clock cycles, i.e. 52% less compared to conventional divisors. This design is mapped onto a Virtex-6 FPGA and an operating frequency of 452.69 MHz is achieved. The proposed design also handles all the IEEE specified four rounding modes, overflow, underflow and various exception conditions**.**

R.panda,M.pradhan[8] presents the concepts behind the "Urdhva Tiryagbhyam Sutra" and "Nikhilam Sutra" multiplication techniques.It then shows the architecture for a 16×16 Vedic multiplier module using Urdhva Tiryagbhyam Sutra [9,10].This paper then extends multiplication to 16×16 Vedic multiplier using "Nikhilam Sutra" technique. The 16×16 Vedic multiplier module using Urdhva Tiryagbhyam Sutra uses four 8×8 Vedic multiplier modules; one 16 bit carry save adders, and two 17 bit full adder stages. The carry save adder in the multiplier architecture increases the speed of addition of partial products. The 16×16 Vedic multiplier is coded in VHDL, synthesized and simulated using Xilinx ISE 10.1 software. This multiplier is implemented on Spartan 2 FPGA device XC2S30-5pq208. The performance evaluation results in terms of speed and device utilization are compared with earlier multiplier architecture.

An improvement in the current restoring computer division algorithm [11] which minimizes the number of restorations performed by the computer. Currently, restoration is checked after each shifting of the register but deprived Restoring Division Algorithm checks restoration only when it is needed. The needless restorations are removed by the use of an extra register. This removal decreases the time for the division than taken in the current restoring division algorithm considerably.

A high speed squaring circuit for binary numbers [12] is proposed. High speed Vedic multiplier is used for design of the proposed squaring circuit. The key to our success is that only one Vedic multiplier is used instead of four multipliers reported in the literature. In addition, one squaring circuit is used twice. Our proposed Squaring Circuit seems to have better performance in terms of speed.

This paper bring out a 32X32 bit reversible Vedic multiplier [13] using "Urdhva Tiryakabhayam" sutra meaning Vertical and crosswise, is designed using reversible logic gates, which is the first of its kind. Also in this paper we propose a new reversible unsigned division circuit. This circuit is designed using reversible components like reversible parallel adder, reversible left-shift register, and reversible. multiplexer, reversible n-bit register with parallel load line. The reversible Vedic multiplier and reversible divider modules have been written in Verilog HDL and then synthesized and simulated using Xilinx ISE 9.2i. This reversible Vedic multiplier results shows less delay and less power consumption by comparing with array multiplier.

3. METHODOLOGY FOR VEDIC DIVISION

Researchers have anticipated many algorithms and Procedural architectures to carry out division in order to ease the computational time and thus enhancing the recital.

1. Restoring Division

Restoring division operates on fixed-point fractional numbers and depends on the following assumptions: (a) $D < N$ and (b) $0 < N, D < 1$.The quotient digits q are formed from the digit set {0, 1}. The basic algorithm for binary (radix 2) restoring division is:

To compute a/b, put a in register A, b in register B

and 0 in register P.

I.Shift the register pair (P, A) one bit left.

II.Subtract the content of Register B from register P.

iii. If the result of step 2 is negative, set the A0 to 0,

otherwise to 1.

iv.If the result of step 2 is negative, restore the old value of P by adding the contents of register B back into P.

After repeating the algorithm, n times, register A will

have quotient and register P with reminder.

2. Non-Restoring Division

Non-restoring division uses the digit set {−1, 1} for the quotient digits instead of {0, 1}. The basic algorithm for binary (radix 2) non-restoring division is:

If P is negative,

1a. Shift (P,A) one bit left.

2a. Add the content of register B to P.

else

1b. Shift (P,A) one bit left.

2b. Subtract the contents of register B from P.

3. If P is negative, set the low-order bit of A to 0,

Otherwise set it to 1.

Repeat the above procedure n times. After n cycles,

Register A will have the quotient and if P is positive, it is the remainder, otherwise it has to be restored (add

B to it) to get the remainder.

3. SRT Division

Named for its creators (Sweeney, Robertson, and Tocher), SRT division is a popular method for division in many microprocessor implementations. SRT division is similar to non-restoring division, but it uses a lookup table based on the dividend and the divisor to determine each quotient digit. The basic algorithm for binary (radix 2) non-restoring division is:

1. Load a and b into A and B registers (Figure A.2)

2. If B has k leading zero, shift B and (P,A) left k bits

3. For l=0, n-1,

a) If top 3 bits of P are equal, set qi=0 and

shift (P,A) one bit left.

b) If top 3 bits of P are not equal and P is negative, set qi=-1 (write as) shift (P,A) one bit left, add B.

c) Otherwise, set qi=1, shift (P,A) one bit left, sub B.

4. If final remainder is negative, correct the remainder by adding B; correct the quotient by subtracting 1 from q0.

5. Shift remainder k bits right.

4. Vedic Architecture: Nikhilam Sutra

Vedic Sutra, Nikhilam can be further extended to carry out Binary Division as an alternative to conventional algorithm. Assume that, A and B are dividend and divisor respectively. Dividend is n-bits wide. The flowchart diagram can be executed as follows:

a)Initialize the incrementer with '0'.

b) Determine the complement B with respect to 2n assume the complemented result is equal to B'.

c) Add B' with A. If the carry is '1', then feed the result to the adder.

d) Increment the content of the incrementer by one.

e) Repeat step-3 until the result is less than B.

f) The final result of the incrementer is the quotient and result from the adder is the remainder.

In this paper we presented the BCD (binary coded decimal) division of thirty two bit divider architecture based on paravartya sutra. It's an optimized method which led to reduction in several steps. So by using paravartya sutra we can reduce several steps as compared to other methods there by reducing the hardware complexity & latency of the circuit. Basically the division is mainly based on two factors

(a) Divisor greater than the base number (for digits like 8, 9, 7 used in the divisor)

(b) Divisor smaller than the base number (for digits like 1,2,3,4 used in the divisor)

For the base method we use the powers of ten as the base i.e. for a digit 898 the base number will be 1000 giving a positive difference of 102.Then the difference between the base number and divisor is calculated. The difference will further acts as the divisor which is a positive number. Then division by paravartya method. But when the divisor is greater than the base number we get the difference a negative number i.e. for a digit 1222 the base will be 1000 giving a difference of - (222). The number of digits of the difference is exactly will be the number of digits of the reminder. So before division we have to give a demarcation line for the quotient & reminder. But in this literature we specially discussed for the small digits which gives negative difference which are well explained in the later section.

3.1 Format for Paravartya Method

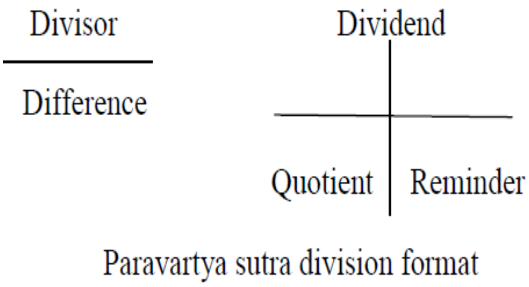


Fig 1:Format for Paravartya Method

3.2 Steps of Division by using Paravartya Sutra

1. We have the dividend & divisor given. Check whether the divisor is greater than the base number or smaller. If grater we will have a positive difference else a negative difference. which will act as the new divisor for further steps.

2. Bring down the very 1st digit of the dividend. Then multiply it with the difference. Now write down the result in the next line of the dividend shifting one bit right.

3. Perform addition to get the 2nd bit and put it down. Same operation should be performed till the last digit of the dividend.

4. In this manner finally the quotient & reminder can be obtained.

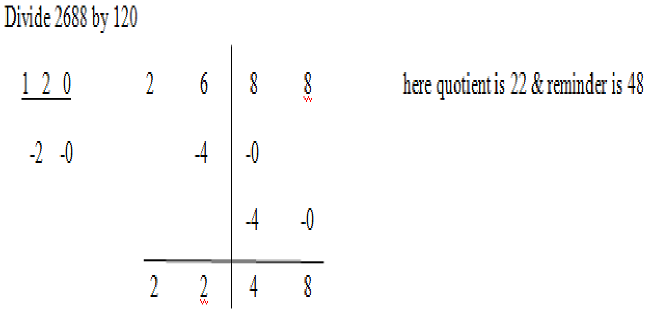
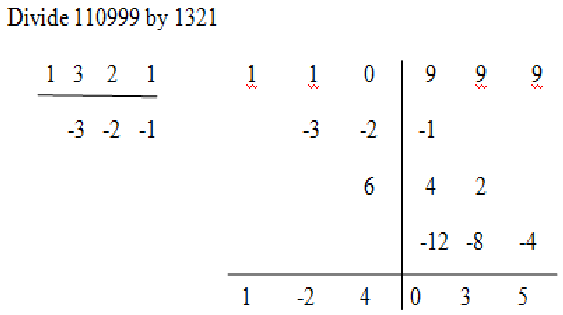


Fig 2: Division of two decimal numbers with positive quotient & reminder

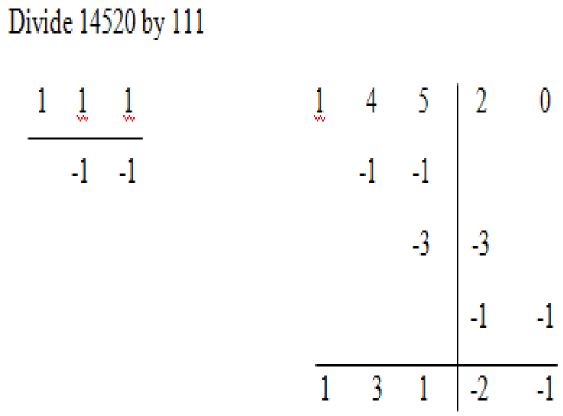


Divide 110999 by 1321

1	3	2	1	1	1	0	9	9	9
	-3	-2	-1		-3	-2	-1		
						6	4	2	
							-12	-8	-4
				1	-2	4	0	3	5

Fig 3:Division of two decimal number with positive quotient & reminder

If we are getting positive value of quotient & reminder then that will be the direct answer but we get negative quotient then we have to check the places of the digit & according to that we can be able to find the result. As in the above example by the places we can calculate it as (100 -20 +4) = 85, so the quotient can be found in this way& reminder is 035.



Divide 14520 by 111

1	1	1	1	4	5	2	0
	-1	-1		-1	-1		
					-3	-3	
						-1	-1
			1	3	1	-2	-1

Fig 4: Division of two decimal numbers with positive quotient & negative reminder

The reminder is found by subtracting it from the divisor as (111-20-1) = 90, So the quotient is 131 & reminder is 90.

3.3 Extension of Paravartya Sutra Using BCD Code

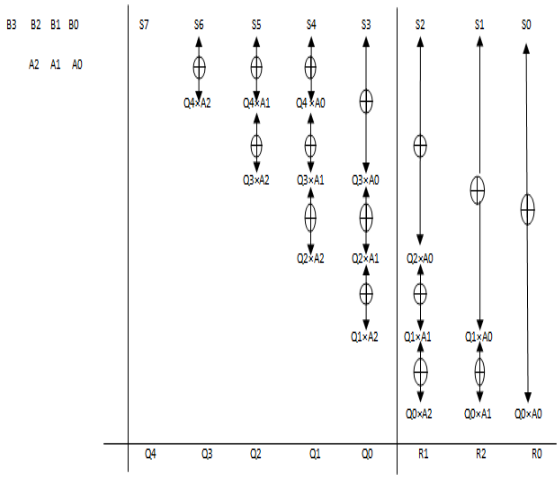


Fig 5: Paravartya Sutra for BCD Code

Here S0 – S7 represents the entire 32 bit no each having 4 digits & B0 – B3 denotes the 16 bit divisor and A2, A1, A0 denotes the 2s complement of B2, B1,B0. In the last row Q4 – Q0 represents the quotient & R1 – R0 the reminder.

4. FLOW CHART OF BCD DIVIDER

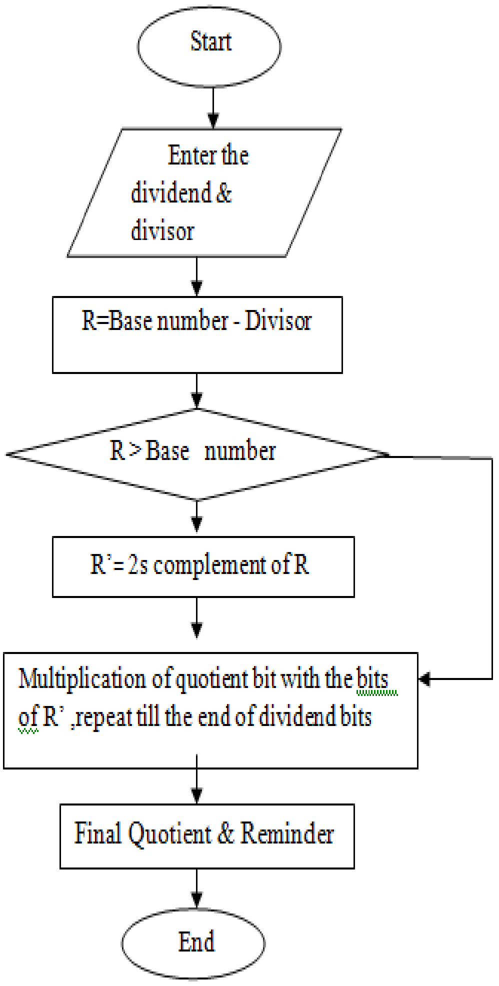

Start
Enter the dividend & divisor
R=Base number - Divisor
R > Base number
R'= 2s complement of R
Multiplication of quotient bit with the bits of R' ,repeat till the end of dividend bits
Final Quotient & Reminder
End

Fig.6. Flowchart for BCD divider

4.1 Algorithm for Implemented BCD Divider

1. The dividend & divisor are given in decimal number system. Write down the dividend & divisors BCD equivalents just below the digits. Give a demarcation line for quotient & reminder.

2. Take the difference between the base number & divisor resulting a negative number which will be acting as new divisor for further steps. Number of digits in new divisor is equal in number with the number of digits in reminder.

3. As the resulted divisor is a negative number & we are implementing in BCD format we have to take the 2s complement of the resulted divisor.

4. Now bring down the 1st digit of the dividend (1st four bits as in BCD format) & multiplication of the digit with the 2s complemented result of the divisor which is written just shifting one digit right of the dividend.

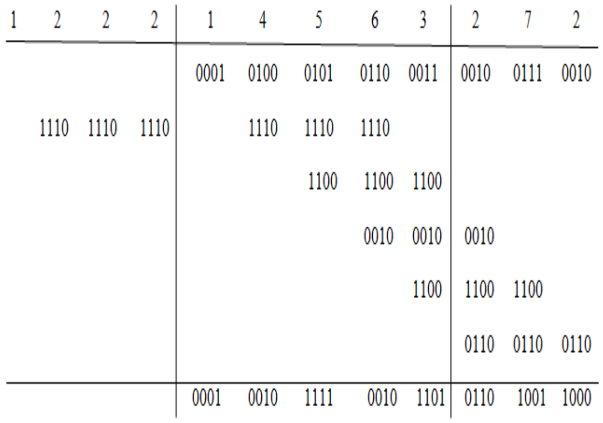
5. Then addition of the 2nd digit of the dividend with the multiplied value, if carry occurs only last four digits should be taken & carry should be discarded. (The addition should be bit by bit).

6. Repeat step 4 & 5 till the end of the digits of dividend bits.

7. The resulted bits after addition gives us the quotient & reminder but the values must not exceed the value 9 (1001) as it's in BCD format, so the values greater than 1001 should be complemented.

8. The obtained result which are in 2s complemented form i.e. either the quotient & reminder digits can be calculated as explained in the above examples.

4.2 32Bit Divided By 16Bit



1	2	2	2	1	4	5	6	3	2	7	2
				0001	0100	0101	0110	0011	0010	0111	0010
	1110	1110	1110		1110	1110	1110				
						1100	1100	1100			
							0010	0010	0010		
								1100	1100	1100	
									0110	0110	0110
				0001	0010	1111	0010	1101	0110	1001	1000

Fig.7. Division of 32 bit number with 16 bit number in BCD format

4.3 Description Of Paravartya Sutra For BCD Format

1. The dividend is 14563272, divisor 1222 is taken & whose BCD equivalent is written.

2. Here the difference we found is -2 -2 -2 so in BCD format we have to take its 2's complement. We will get it as 1100 1100 1100.

3. Then bring down the 1st place digits of the dividend.(0001)

4. Multiply of the 2's complemented result 1110 1110 1110 with 0001 & the result is 1110.

5. Multiplied result should be written below the 2nd place digits(0100) & addition should done (if carry occurs neglect & only last 4 digits should be taken).

6. Now the 2nd column result is 0010 and will be multiplied with 1110 1110 1110,result will be written in starting place of (2nd row) / under 3rd place digits.

7. In the very next step we have to perform addition operation, but that should be added bit by bit means 0101 + 1110 = 1,0011 here the 1 should be discarded & now 0011 is added with the next value i.e. 0011 + 1100 = 1111.

8. But as it's a BCD format 1111,1101 are not allowed .So in above example whatever values we are getting more than 0 - 9 all those values are again 2's complemented

9. Same process should be used until the last digit of the dividend; finally we have the quotient as0001 0010 0001 0010 0011 & reminder 0110 1001 1000.

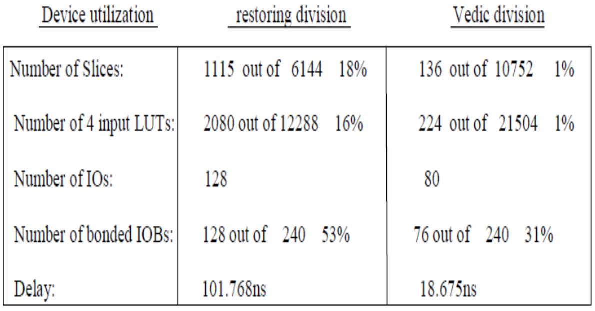
10. So the quotient is 1 2 -1 2 -3 & reminder is 698.

11. Now again the result should not be in 2s complement form so the final quotient can be calculated as 10000 + 2000 – 100 + 20 – 3 = 11917 & reminder 698.

5. RESULT ANALYSIS

The design of the divider using paravartya sutra of Vedic mathematics consists of several steps like design entry, synthesis, simulation & implementation. In the very first step i.e. design entry we have written the verilog HDL code for the proposed paravartya divider. In the next step the code is checked for errors & when no error found, and then the code is synthesized using XILINX9.2 software synthesis tool.We have simulated the proposed divider by creating a test bench file by taking two set of combination of dividend & divisor operands. XILINX 9.2 ISE simulation tool is used to simulate the created test bench file. By Xilinx implementation tool a user constraint file is created in virtex4 FPGA device XC4VLX15 with package SF363 of speed -12 for the implementation of the proposed divider. As there are no scientific papers available for hardware implementation of 32 bit divider architecture, so to compare our result we implemented restoring division method (R.Bhaskar.Ganapathi Hegde, P.R.Vaya 2011) in VHDL and synthesized using XILINX 9.2. and compared our result with this result. We implemented parvartya sutra in BCD format because 8 bit binary division discussed (Ratiranjan Bandan & Pradhan 2012) shows a delay of 19.9ns & our proposed divider which performs 32bit division shows a delay of 18.67ns.Oviously the hardware complexity is more in case of binary divider .

Table 1. Comparison of synthesis result of restoring division to paravartya division (32 bit)



Device utilization	restoring division	Vedic division
Number of Slices:	1115 out of 6144 18%	136 out of 10752 1%
Number of 4 input LUTs:	2080 out of 12288 16%	224 out of 21504 1%
Number of IOs:	128	80
Number of bonded IOBs:	128 out of 240 53%	76 out of 240 31%
Delay:	101.768ns	18.675ns

From the table we are very clear about the result of device utilization and delay as the restoring division method shows a larger delay as compared to the proposed divider & the percentage of device utilization in restoring division method is quite high from Vedic divider

6. CONCLUSION

This literature presents a novel design & an algorithm to construct a new high speed thirty two bit divider .It also discusses example to how the method is implemented & which is evaluated in terms of the path delay & device utilization. Our synthesis result shows better result i.e. improved delay & better device utilization. It is seen that the speed of the proposed divider is higher than that of the restoring division method i.e. the delay has been drastically reduced. This divider can be used in applications such as digital signal processing, cryptography & processor ALU design. Further the above thesis work modified for optimized delay, less power consumption, less area and also FPGA implementation will done as future work.

7. REFERENCES

[1] Abhishek Gupta “Arithmetic Unit Implementation Using Delay Optimized Vedic Multiplier with BIST Capability” International Journal of Engineering and Innovative Technology (IJEIT) Volume 1, Issue 5, May 2012.

[2] Manoranjan Pradhan and Rutuparna Panda “High speed multiplier using Nikhilam Sutra algorithm of Vedic mathematics’On: 12 April 2013, International Journal of Electronics.

[3] Jagadguru Swami Sri Bharati Krishna Tirthatji Maharaja, Vedic Mathematics, Motilal Banarsidass Publication, 1992.

[4] Ashish Joglekar , Shaunak Vaidya , Ajinkya Kale, “A Novel Binary Division Algorithm based on Vedic Mathematics and Applications to Polynomial Division” 2007.

[5] R.Bhaskar.Ganapathi Hegde, P.R.Vaya”An efficient hardware model for RSA Encryption system using Vedic mathematics. International Conference on Communication Technology and System Design 2011.

[6] Ming Der Shieh,jun Hong Chen, “A new modular exponentiation architecture for efficient design of RSA Cryptosystem”,IEEE transactions on VLSI systems,Sept.2008.

[7] Purna Ramesh Addanki, Venkata Nagaratna Tilak Alapati and Mallikarjuna Prasad Avana “Floating Point Double Precision Divisor Using Verilog” International Journal of Advanced Science and Technology Vol. 48, November, 2012.

[8] Rutuparna panda,M.pradhan,” Speed comparision of 16*16 vedic multipliers” IJCA,vol-21,may-2011

[9] Arati Sahu, Siba kumar Panda et al. “HDL Implementation and Performance Comparison of an Optimized High Speed Multiplier”, International Organization Of Scientific Research (IOSR), Issue 2, vol.5, April 2015.

[10] Siba Ku Panda, Ritisnigdha Das et al. “VLSI Implementation of Vedic Multiplier Using Urdhva-Tiryakbhyam Sutra in VHDL Environment: A Novelty”, IOSR, Vol.5, February 2015.

[11] Nitish Aggarwal, Kartik Asooja, Saurabh Shekhar Verma, Sapna Negi “An Improvement in the Restoring Division Algorithm” 2009 IEEE.

[12] K.sethi & R.panda “ An improved squaring circuit for binary numbers”, International Journal of Advanced Computer Science and Applications, page111–116 , 2012.

[13] Srikanth G, Nasam Sai Kumar” Design of High speed Low Power Reversible Vedic multiplier and Reversible Divider”IJERAISSN : 2248-9622, Vol. 4, Issue 9(Version 5), September 2014, pp.70-74

8. AUTHORS PROFILE

Mr.Siba Kumar Panda was born on November 09,1989.He received the B.Tech degree in Electronics & Communication Engineering from Biju Patnaik University of Technology,odisha in 2012 and M.Tech degree in *VLSI Signal Processing* Specialization from Veer Surendra Sai University Of Technology,odisha in 2014.

Currently he is working as an Assistant Professor at Centurion University of Technology and Management, Bhubaneswar; Odisha.He also awarded the University Silvermedal for best Electronics & Telecommunication Engineering Post Graduate for the academic year 2012-2014 at VSSUT, Odisha..

His research area of interest includes Ultra-wideband (UWB) device design, RF circuit design using CMOS Technique,VLSI implementation of Vedic mathematics,VLSI Signal Processing.

Mrs Arati Sahu was born in Rourkela , Odisha , India, in 1991. She has completed B.Tech in Electronics and Communication Engineering,from Sundargarh Engineering Collage,Sundargarh.Currently pursuing her M.tech degree in VLSI from ECE , Centurion University of Technology and Management, Bhubaneswar, Odisha 751020 ,India, in the academic period of 2013-2015.